Citrix ADC 13.0

Contents

Citrix ADC Release Notes

November 15, 2021

Release notes describe how the software has changed in a particular build, and the issues known to exist in the build.

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

- **What's New**: The enhancements and other changes released in the build.
- **Fixed Issues**: The issues that are fixed in the build.
- **Known Issues**: The issues that exist in the build.
- **Points to Note**: The important aspects to keep in mind while using the build.
- **Limitations**: The limitations that exist in the build.

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

To view the release notes document for a specific build, click the corresponding link in the following table. When the release notes are updated for a build, the version number of the release notes and the publish date are also updated. The release notes publish date might not be the same as the build GA date.

Release notes for Citrix ADC release 13.0	Release notes publish date	Release notes version
Build 83.29	November 15, 2021	1.0
Build 82.45	July 19, 2021	2.0
Build 79.64	April 28, 2021	3.0
Build 76.31	April 06, 2021	3.0
Build 71.44	February 19, 2021	4.0
Build 67.43	November 26, 2020	2.0
Build 64.35	January 21, 2021	4.0
Build 61.48	September 04, 2020	3.0
Build 58.32	July 07, 2020	1.0
Build 52.24	July 20, 2020	4.0
Build 47.24	April 20, 2020	2.0

Release notes for Citrix ADC release 13.0	Release notes publish date	Release notes version
Build 41.28	April 20, 2020	3.0

Getting started with Citrix ADC

September 14, 2021

This topic describes the basic features and configuration details of a Citrix ADC appliance. System and network administrators who install and configure network equipment can refer to the content.

Understanding Citrix ADC

The Citrix ADC appliance is an application switch which performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4–L7) network traffic for web applications. For example, a Citrix ADC appliance load balances decisions on individual HTTP requests instead of long-lived TCP connections. The load balancing feature helps slowing down the failure of a server with less disruption to clients. The ADC features can be broadly classified as:

1. Data switching
2. Firewall security
3. Optimization
4. Policy infrastructure
5. Packet flow
6. System limitation

Data switching

When deployed in front of application servers, a Citrix ADC ensures optimal distribution of traffic by how it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP or TCP request, and based on L4–L7 header information such as URL, application data type, or cookie. Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

Firewall security

The Citrix ADC security and protection protect web applications from Application Layer attacks. An ADC appliance allows legitimate client requests and can block malicious requests. It provides built-in

defenses against denial-of-service (DoS) attacks and supports features that protect against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from Application Layer attacks, including buffer overflow exploits, SQL injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

Optimization

Optimization offloads resource-intensive operations, such as Secure Sockets Layer (SSL) processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. This improves the performance of the servers in the server farm and therefore speeds up applications. An ADC appliance supports several transparent TCP optimizations which mitigate problems caused by high latency and congested network links. Thereby accelerating the delivery of applications while requiring no configuration changes to clients or servers.

Policy infrastructure

A policy defines specific details of traffic filtering and management on a Citrix ADC. It consists of two parts: the expression and the action. The expression defines the types of requests that the policy matches. The action tells the ADC appliance what to do when a request matches the expression. For example, the expression might be to match a specific URL pattern for a security attack with the configured to drop or reset the connection. Each policy has a priority, and the priorities determine the order in which the policies are evaluated.

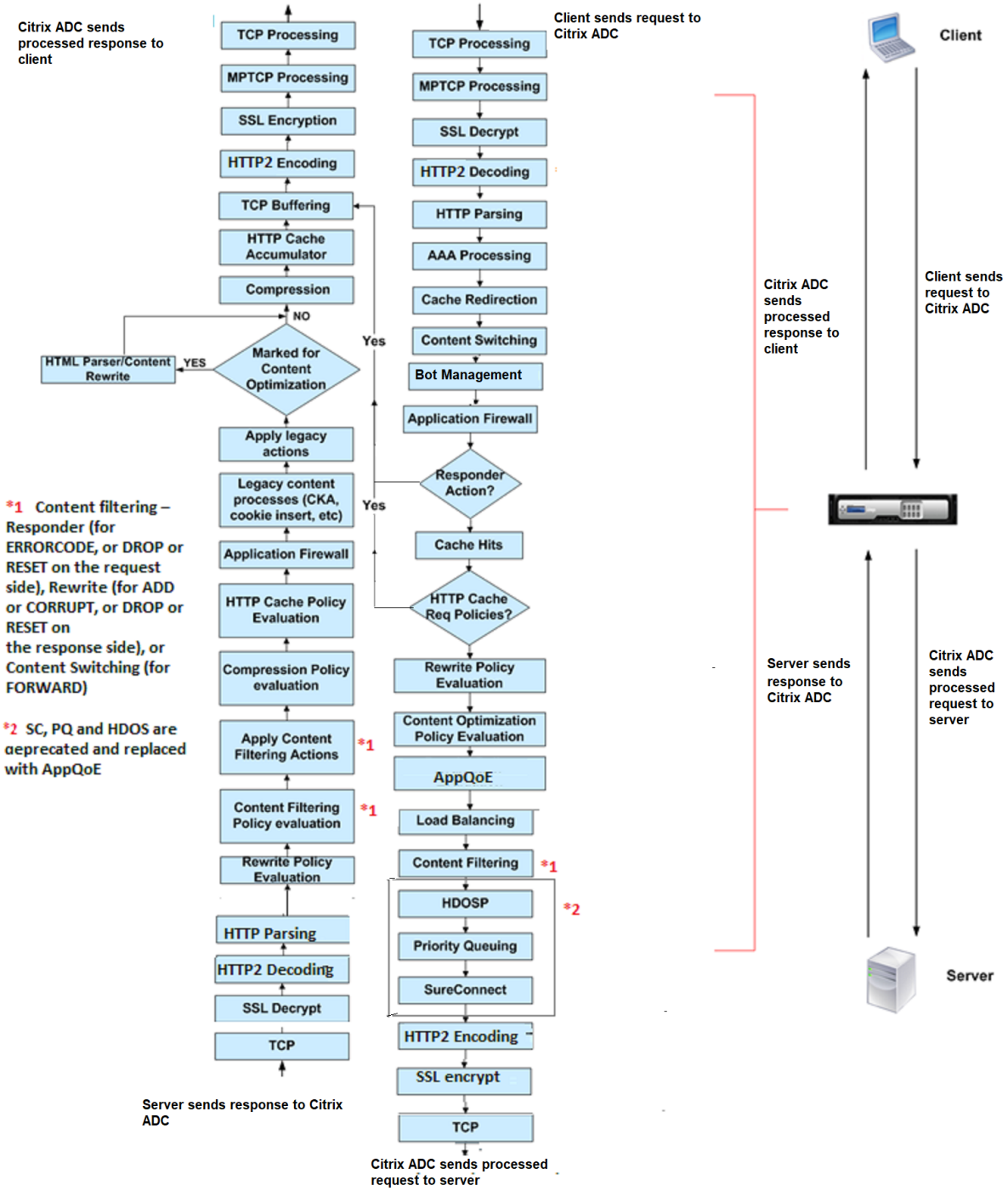
When an ADC appliance receives traffic, the appropriate policy list determines how to process the traffic. Each policy on the list contains one or more expressions, which together define the criteria that a connection must meet to match the policy.

For all policy types except rewrite, the appliance implements only the first policy that has a request match. For Rewrite policies, the ADC appliance evaluates the policies in order and performs the associated actions in the same order. Policy priority is important for getting the results you want.

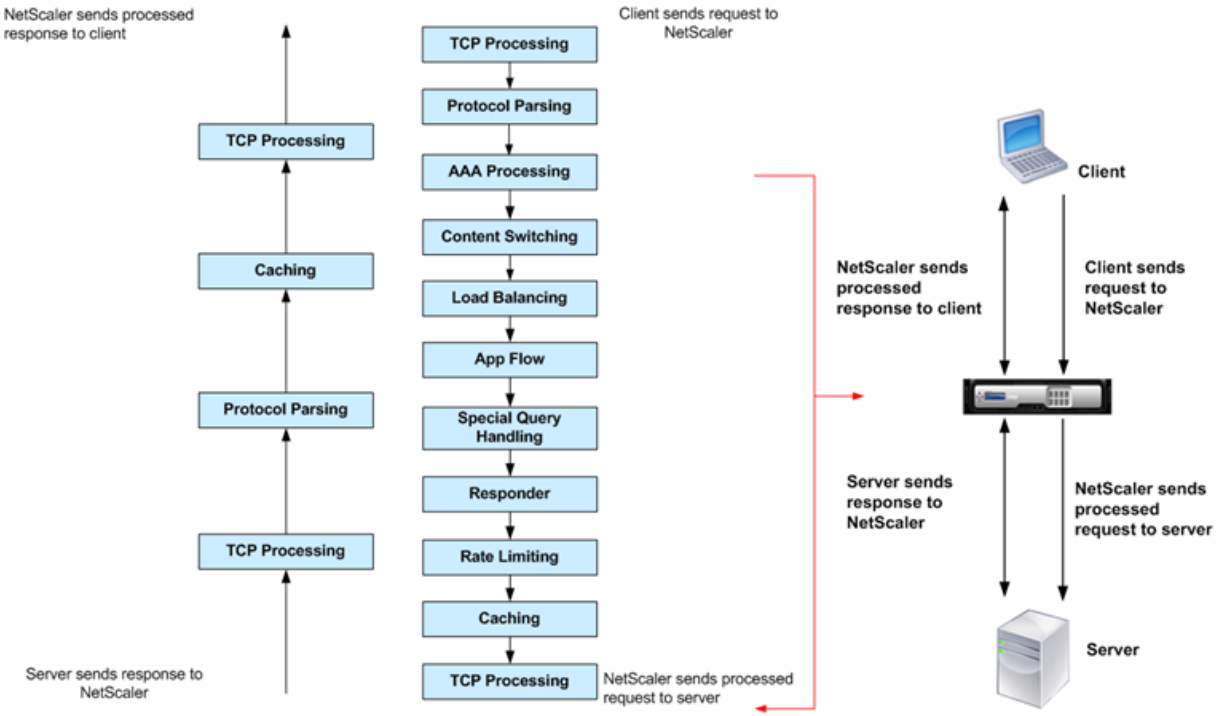
Packet flow

Depending on requirements, you can choose to configure multiple features. For example, you might choose to configure both compression and SSL offload. As a result, an outgoing packet might be compressed and then encrypted before being sent to the client.

The following figure shows the DataStream packet flow in the Citrix ADC appliance. DataStream is supported for MySQL and MS SQL databases.



The following figure shows the DataStream packet flow in the Citrix ADC appliance. DataStream is supported for MySQL and MS SQL databases. For information about the DataStream feature, see DataStream.



> **Note:** If the traffic is for a content switching virtual server, the appliance evaluates policies in the following order:
>
> 1. bound to global override.
> 2. bound to load balancing virtual server.
> 3. bound to content switching virtual server.
> 4. bound to global default.
>
> This way, if a policy rule is true and gotopriorityexpression is END, we stop further policy evaluation.
> In content switching, if no load balancing virtual server is selected or bound to the content switching virtual server, then we evaluate responder policies bound only to the content switching virtual server.

System limitation

There are system limitations for each Citrix ADC feature when you install Citrix ADC software 9.2 or later. For more information, see Citrix article, CTX118716.

Where does a Citrix ADC appliance fit in the network?

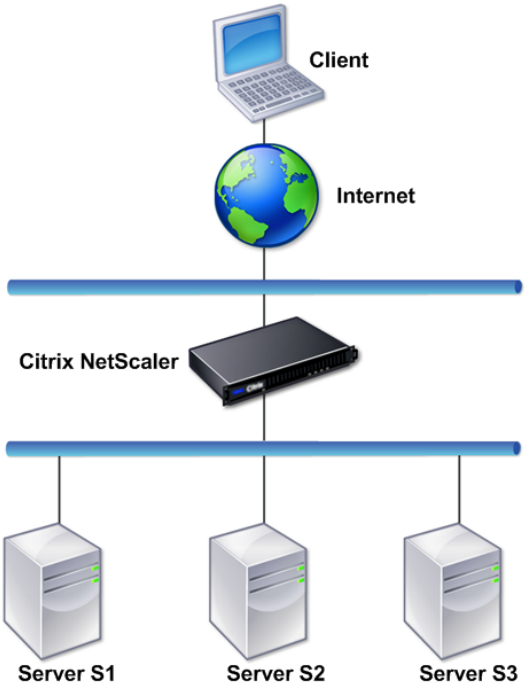
September 14, 2021

A Citrix ADC appliance resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. In this case, the appliance owns public IP addresses that are associated with its virtual servers, while the real servers are isolated in a private network. It is also possible to operate the appliance in a transparent mode as an L2 bridge or L3 router, or even to combine aspects of these and other modes.

Physical deployment modes

A Citrix ADC appliance logically residing between clients and servers can be deployed in either of two physical modes: inline and one-arm. In inline mode, multiple network interfaces are connected to different Ethernet segments, and the appliance is placed between the clients and the servers. The appliance has a separate network interface to each client network and a separate network interface to each server network. The appliance and the servers can exist on different subnets in this configuration. It is possible for the servers to be in a public network and the clients to directly access the servers through the appliance, with the appliance transparently applying the L4-L7 features. Usually, virtual servers (described later) are configured to provide an abstraction of the real servers. The following figure shows a typical inline deployment.

Figure 1. Inline Deployment



In one-arm mode, only one network interface of the appliance is connected to an Ethernet segment. The appliance in this case does not isolate the client and server sides of the network, but provides access to applications through configured virtual servers. One-arm mode can simplify network changes needed for Citrix ADC installation in some environments.

For examples of inline (two-arm) and one-arm deployment, see Understanding Common Network Topologies.

Citrix ADC as an L2 device

A Citrix ADC appliance functioning as an L2 device is said to operate in L2 mode. In L2 mode, the ADC appliance forwards packets between network interfaces when all of the following conditions are met:

- The packets are destined to another device's media access control (MAC) address.
- The destination MAC address is on a different network interface.
- The network interface is a member of the same virtual LAN (VLAN).

By default, all network interfaces are members of a pre-defined VLAN, VLAN 1. Address Resolution Protocol (ARP) requests and responses are forwarded to all network interfaces that are members of the same VLAN. To avoid bridging loops, L2 mode must be disabled if another L2 device is working in parallel with the Citrix ADC appliance.

For information about how the L2 and L3 modes interact, see Packet forwarding modes.

For information about configuring L2 mode, see the “Enable and disable layer 2 mode” section in Packet forwarding modes.

Citrix ADC as a packet forwarding device

A Citrix ADC appliance can function as a packet forwarding device, and this mode of operation is called L3 mode. With L3 mode enabled, the appliance forwards any received unicast packets that are destined for an IP address that does not belong to the appliance, if there is a route to the destination. The appliance can also route packets between VLANs.

In both modes of operation, L2 and L3, the appliance generally drops packets that are in:

- Multicast frames
- Unknown protocol frames destined for an appliance’s MAC address (non-IP and non-ARP)
- Spanning Tree protocol (unless BridgeBPDUs is ON)

For information about how the L2 and L3 modes interact, see Packet forwarding modes.

For information about configuring the L3 mode, see Packet forwarding modes.

How a Citrix ADC appliance communicates with clients and servers

September 14, 2021

A Citrix ADC appliance is usually deployed in front of a server farm and functions as a transparent TCP proxy between clients and servers, without requiring any client-side configuration. This basic mode of operation is called Request Switching technology and is the core of Citrix ADC functionality. Request Switching enables an appliance to multiplex and offload the TCP connections, maintain persistent connections, and manage traffic at the request (application layer) level. This is possible because the appliance can separate the HTTP request from the TCP connection on which the request is delivered.

Depending on the configuration, an appliance might process the traffic before forwarding the request to a server. For example, if the client attempts to access a secure application on the server, the appliance might perform the necessary SSL processing before sending traffic to the server.

To facilitate efficient and secure access to server resources, an appliance uses a set of IP addresses collectively known as Citrix ADC-owned IP addresses. To manage your network traffic, you assign Citrix ADC-owned IP addresses to virtual entities that become the building blocks of your configuration. For example, to configure load balancing, you create virtual servers to receive client requests and distribute them to services, which are entities representing the applications on your servers.

Understanding Citrix ADC-owned IP addresses

To function as a proxy, a Citrix ADC appliance uses a variety of IP addresses. The key Citrix ADC-owned IP addresses are:

- Citrix ADC IP (NSIP) address

  The NSIP address is the IP address for management and general system access to the appliance itself, and for communication between appliances in a high availability configuration.

- Virtual server IP (VIP) address

  A VIP address is the IP address associated with a virtual server. It is the public IP address to which clients connect. An appliance managing a wide range of traffic may have many VIPs configured.

- Subnet IP (SNIP) address

  A SNIP address is used in connection management and server monitoring. You can specify multiple SNIP addresses for each subnet. SNIP addresses can be bound to a VLAN.

- IP Set

  An IP set is a set of IP addresses, which are configured on the appliance as SNIP . An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it.

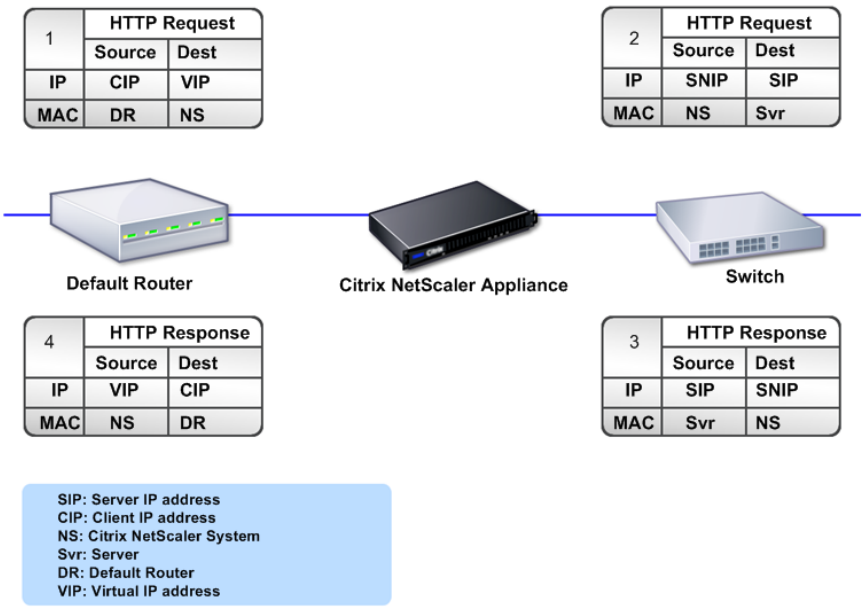
- Net Profile

  A net profile (or network profile) contains an IP address or an IP set. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. During communication with physical servers or peers, the appliance uses the addresses specified in the profile as source IP addresses.

How Traffic flows are managed

Because a Citrix ADC appliance functions as a TCP proxy, it translates IP addresses before sending packets to a server. When you configure a virtual server, clients connect to a VIP address on the Citrix ADC appliance instead of directly connecting to a server. As determined by the settings on the virtual server, the appliance selects an appropriate server and sends the client's request to that server. By default, the appliance uses a SNIP address to establish connections with the server, as shown in the following figure.

Figure 1. Virtual Server Based Connections



In the absence of a virtual server, when an appliance receives a request, it transparently forwards the request to the server. This is called the transparent mode of operation. When operating in transparent mode, an appliance translates the source IP addresses of incoming client requests to the SNIP address but does not change the destination IP address. For this mode to work, L2 or L3 mode has to be configured appropriately.

For cases in which the servers need the actual client IP address, the appliance can be configured to modify the HTTP header by inserting the client IP address as an additional field, or configured to use the client IP address instead of a SNIP address for connections to the servers.

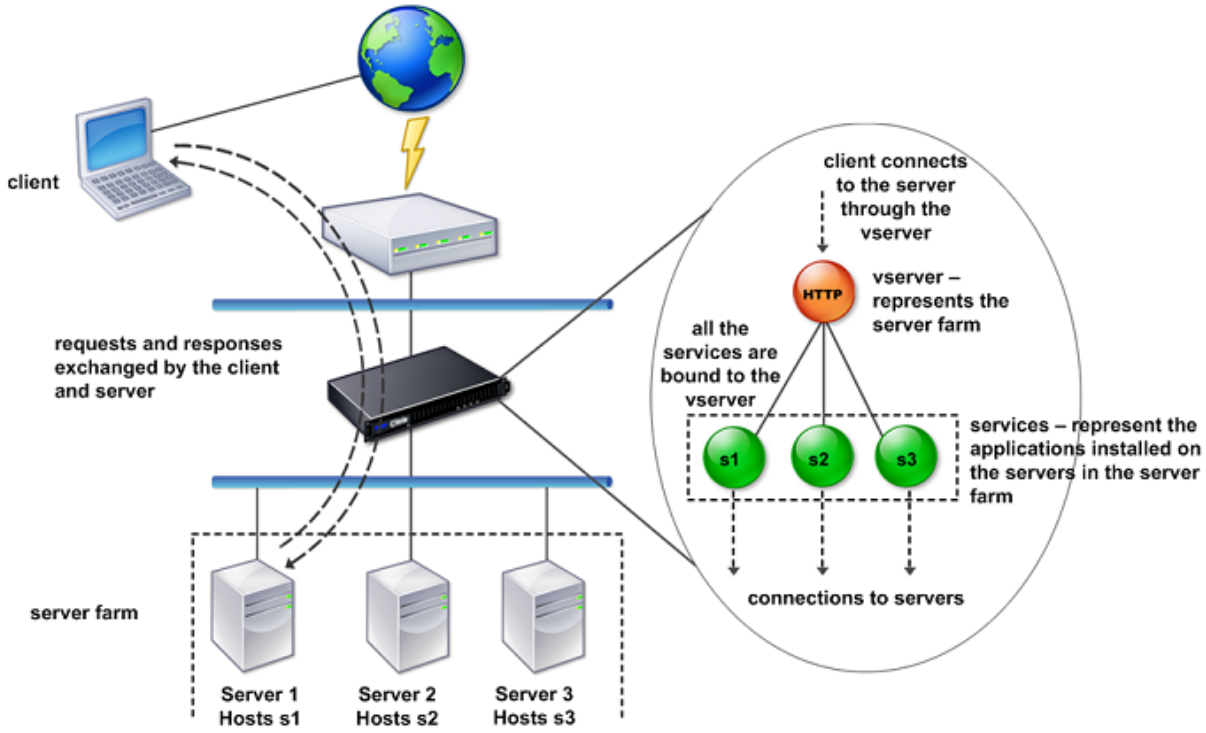
Traffic management building blocks

The configuration of a Citrix ADC appliance is typically built up with a series of virtual entities that serve as building blocks for traffic management. The building block approach helps separate traffic flows. Virtual entities are abstractions, typically representing IP addresses, ports, and protocol handlers for processing traffic. Clients access applications and resources through these virtual entities. The most commonly used entities are virtual servers and services. Virtual servers represent groups of servers in a server farm or remote network, and services represent specific applications on each server.

Most features and traffic settings are enabled through virtual entities. For example, you can configure

an appliance to compress all server responses to a client that is connected to the server farm through a particular virtual server. To configure the appliance for a particular environment, you need to identify the appropriate features and then choose the right mix of virtual entities to deliver them. Most features are delivered through a cascade of virtual entities that are bound to each other. In this case, the virtual entities are like blocks being assembled into the final structure of a delivered application. You can add, remove, modify, bind, enable, and disable the virtual entities to configure the features. The following figure shows the concepts covered in this section.

Figure 2. How traffic management building blocks work



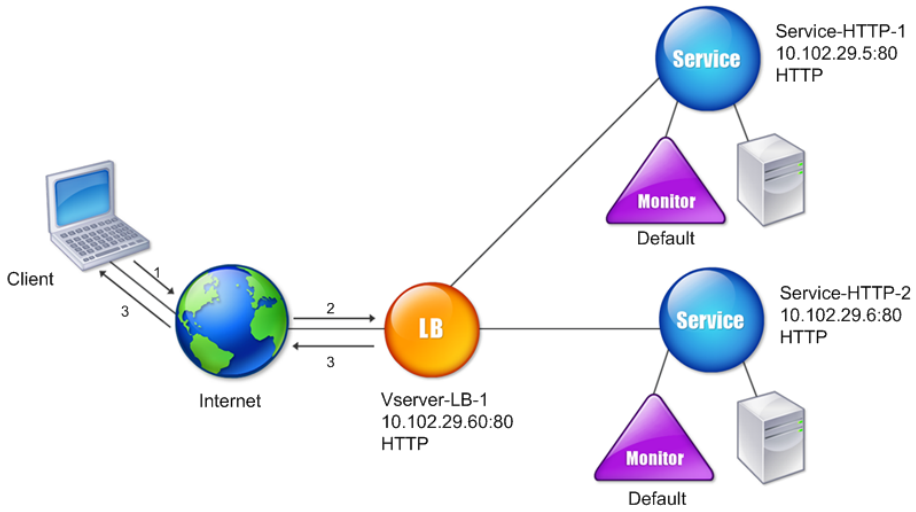
A simple load balancing configuration

In the example shown in the following figure, the Citrix ADC appliance is configured to function as a load balancer. For this configuration, you need to configure virtual entities specific to load balancing and bind them in a specific order. As a load balancer, an appliance distributes client requests across several servers and thus optimizes the utilization of resources.

The basic building blocks of a typical load balancing configuration are services and load balancing virtual servers. The services represent the applications on the servers. The virtual servers abstract the servers by providing a single IP address to which the clients connect. To ensure that client requests are sent to a server, you need to bind each service to a virtual server. That is, you must create services for every server and bind the services to a virtual server. Clients use the VIP address to connect to a Citrix ADC appliance. When the appliance receives client requests sent to the VIP address, it sends

them to a server determined by the load balancing algorithm. Load balancing uses a virtual entity called a monitor to track whether a specific configured service (server plus application) is available to receive requests.

Figure 3. Load balancing virtual server, services, and monitors



In addition to configuring the load balancing algorithm, you can configure several parameters that affect the behavior and performance of the load balancing configuration. For example, you can configure the virtual server to maintain persistence based on source IP address. The appliance then directs all requests from any specific IP address to the same server.

Understanding virtual servers

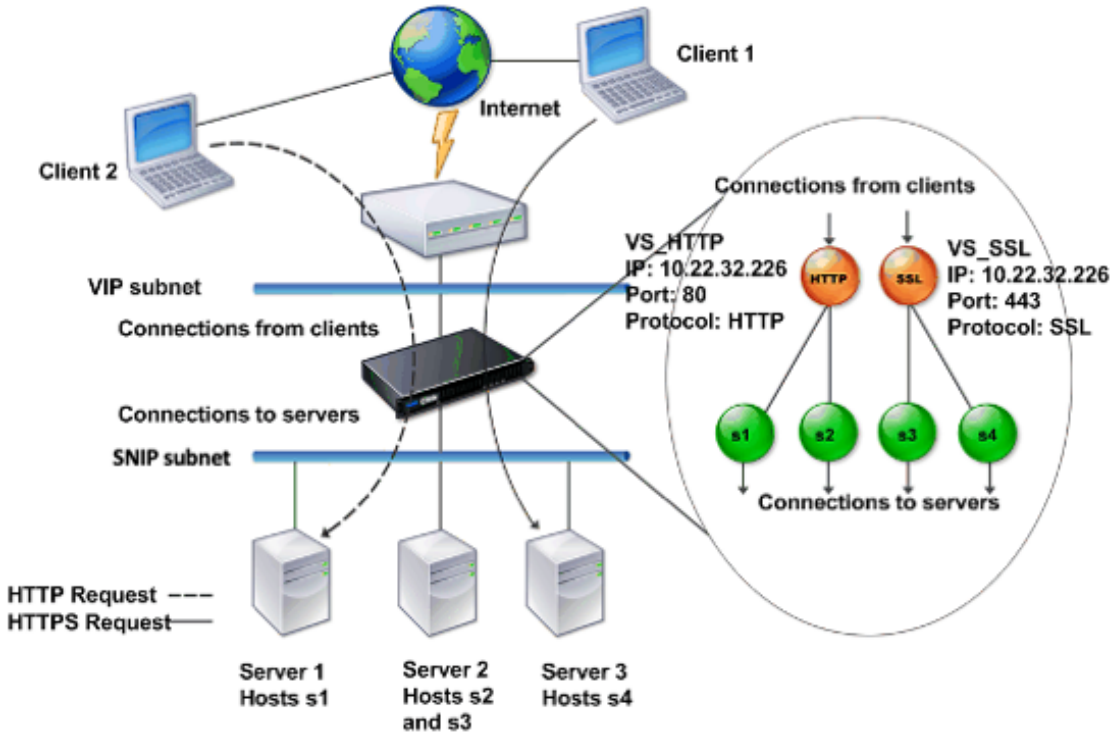
A virtual server is a named Citrix ADC entity that external clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP (VIP) address, port, and protocol. The name of the virtual server is of only local significance and is designed to make the virtual server easier to identify. When a client attempts to access applications on a server, it sends a request to the VIP instead of the IP address of the physical server. When the appliance receives a request at the VIP address, it terminates the connection at the virtual server and uses its own connection with the server on behalf of the client. The port and protocol settings of the virtual server determine the applications

that the virtual server represents. For example, a web server can be represented by a virtual server and a service whose port and protocol are set to 80 and HTTP, respectively. Multiple virtual servers can use the same VIP address but different protocols and ports.

Virtual servers are points for delivering features. Most features, like compression, caching, and SSL offload, are normally enabled on a virtual server. When the appliance receives a request at a VIP address, it chooses the appropriate virtual server by the port on which the request was received and its protocol. The appliance then processes the request as appropriate for the features configured on the virtual server.

In most cases, virtual servers work in tandem with services. You can bind multiple services to a virtual server. These services represent the applications running on physical servers in a server farm. After the appliance processes requests received at a VIP address, it forwards them to the servers as determined by the load balancing algorithm configured on the virtual server. The following figure illustrates these concepts.

Figure 4. Multiple Virtual Servers with a Single VIP Address



The preceding figure shows a configuration consisting of two virtual servers with a common VIP address but different ports and protocols. Each of the virtual servers has two services bound to it. The services s1 and s2 are bound to VS_HTTP and represent the HTTP applications on Server 1 and Server 2. The services s3 and s4 are bound to VS_SSL and represent the SSL applications on Server 2 and Server 3 (Server 2 provides both HTTP and SSL applications). When the appliance receives an HTTP

request at the VIP address, it processes the request as specified by the settings of VS_HTTP and sends it to either Server 1 or Server 2. Similarly, when the appliance receives an HTTPS request at the VIP address, it processes it as specified by the settings of VS_SSL and it sends it to either Server 2 or Server 3.

Virtual servers are not always represented by specific IP addresses, port numbers, or protocols. They can be represented by wildcards, in which case they are known as wildcard virtual servers. For example, when you configure a virtual server with a wildcard instead of a VIP, but with a specific port number, the appliance intercepts and processes all traffic conforming to that protocol and destined for the predefined port. For virtual servers with wildcards instead of VIPs and port numbers, the appliance intercepts and processes all traffic conforming to the protocol.

Virtual servers can be grouped into the following categories:

- Load balancing virtual server

  Receives and redirects requests to an appropriate server. Choice of the appropriate server is based on which of the various load balancing methods the user configures.

- Cache redirection virtual server

  Redirects client requests for dynamic content to origin servers, and requests for static content to cache servers. Cache redirection virtual servers often work in conjunction with load balancing virtual servers.

- Content switching virtual server

  Directs traffic to a server on the basis of the content that the client has requested. For example, you can create a content switching virtual server that directs all client requests for images to a server that serves images only. Content switching virtual servers often work in conjunction with load balancing virtual servers.

- Virtual private network (VPN) virtual server

  Decrypts tunneled traffic and sends it to intranet applications.

- SSL virtual server

  Receives and decrypts SSL traffic, and then redirects to an appropriate server. Choosing the appropriate server is similar to choosing a load balancing virtual server.

Understanding services

Services represent applications on a server. While services are normally combined with virtual servers, in the absence of a virtual server, a service can still manage application-specific traffic. For example, you can create an HTTP service on a Citrix ADC appliance to represent a web server application. When the client attempts to access a web site hosted on the web server, the appliance intercepts the HTTP requests and creates a transparent connection with the web server.

In service-only mode, an appliance functions as a proxy. It terminates client connections, uses a SNIP address to establish a connection to the server, and translates the source IP addresses of incoming client requests to a SNIP address. Although the clients send requests directly to the IP address of the server, the server sees them as coming from the SNIP address. The appliance translates the IP addresses, port numbers, and sequence numbers.

A service is also a point for applying features. Consider the example of SSL acceleration. To use this feature, you must create an SSL service and bind an SSL certificate to the service. When the appliance receives an HTTPS request, it decrypts the traffic and sends it, in clear text, to the server. Only a limited set of features can be configured in the service-only case.

Services use entities called monitors to track the health of applications. Every service has a default monitor, which is based on the service type, bound to it. As specified by the settings configured on the monitor, the appliance sends probes to the application at regular intervals to determine its state. If the probes fail, the appliance marks the service as down. In such cases, the appliance responds to client requests with an appropriate error message or re-routes the request as determined by the configured load balancing policies.

Introduction to the Citrix ADC product line

September 14, 2021

The Citrix ADC product line optimizes delivery of applications over the internet and private networks, combining application-level security, optimization, and traffic management into a single, integrated appliance. You can install a Citrix ADC appliance in your server room and route all connections to your managed servers through it. The Citrix ADC features that you enable and the policies you set are then applied to incoming and outgoing traffic.

A Citrix ADC appliance can be integrated into any network as a complement to existing load balancers, servers, caches, and firewalls. It requires no additional client or server side software, and can be configured using the Citrix ADC web-based GUI and CLI configuration utilities.

This topic includes the following sections:

- Citrix ADC Hardware Platforms
- Citrix ADC Editions
- Supported Releases on ADC Hardware
- Supported Browsers

Citrix ADC Hardware Platforms

Citrix ADC hardware is available in a variety of platforms that have a range of hardware specifications:

Citrix ADC MPX hardware platform

Citrix ADC SDX hardware platform

Citrix ADC Editions

The Citrix ADC operating system is available in three editions:

- Standard
- Advanced
- Premium

The Standard and Advanced editions have limited features available. Feature licenses are required for all editions.

For more information about Citrix ADC software editions, see the Citrix ADC Editions datasheet.

For information about how to obtain and install licenses, see Licensing.

Supported releases on Citrix ADC hardware

See the following compatibility matrix tables for all Citrix ADC hardware platforms and the software releases supported on these platforms:

Citrix ADC MPX Hardware-Software Compatibility Matrix

Citrix ADC SDX Hardware-Software Compatibility Matrix

Supported browsers

To access the Citrix ADC GUI, your workstation must have a supported web browser.

The following table lists the compatible browsers for NetScaler GUI version 12.0, 12.1, and 13.0:

Operating System	Browser	Versions
Windows 7 & later	Internet Explorer	11, Edge, & later
Windows 7 & later	Mozilla Firefox	45 & later
Windows 7 & later	Chrome	60 & later
MAC	Mozilla Firefox	45 & later
MAC	Safari	10.1.1 & later

The compatible browser versions for Citrix ADC 11.1 are as follows:

Operating System	Browser	Versions
Windows 7 & later	Internet Explorer	8,9,10, 11, Edge
Windows 7 & later	Mozilla Firefox	45 & later
Windows 7 & later	Chrome	60 & later
MAC	Mozilla Firefox	45 & later
MAC	Safari	10.1.1 & later

Install the hardware

September 14, 2021

Before installing a Citrix ADC appliance, review the pre-installation checklist.

To use the SDX appliance, you must complete the following tasks by following the instructions given in the resources provided in table. Complete the tasks in the sequence given.

Task

Description

1. Read safety, cautions, warnings, and other information

   Read the caution and danger information you need to know, before installing the product.

2. Prepare for installation

   Unpack your appliance and ensure all parts were delivered, prepare the site and rack, and follow basic electrical safety precautions before you install your new appliance.

3. Install the hardware

   Rack mount the appliance, install transceivers (if available), and connect the appliance to the network and a power source.

4. Configure the appliance.

   Configure the initial settings of the Citrix ADC appliance by using the GUI or the serial console.

   Follow the steps given in the following documentations to complete these tasks:

   - Citrix ADC MPX hardware documentation
   - Citrix ADC SDX hardware documentation

Access a Citrix ADC appliance

September 14, 2021

A Citrix ADC appliance has both a command line interface (CLI) and a GUI. The GUI includes a configuration utility for configuring the appliance and a statistical utility, called Dashboard. For initial access, all appliances ship with the default Citrix ADC IP address (NSIP) of 192.168.100.1 and the default subnet mask of 255.255.0.0. You can assign a new NSIP and an associated subnet mask during initial configuration.

If you encounter an IP address conflict when deploying multiple Citrix ADC units, check for the following possible causes:

- Did you select an NSIP that is an IP address already assigned to another device on your network?
- Did you assign the same NSIP to multiple Citrix ADC appliances?
- The NSIP is reachable on all physical ports. The ports on a Citrix ADC are host ports, not switch ports.

The following table summarizes the available access methods.

Access Method	Port	Default IP Address Required? (Y/N)
CLI	Console	N
CLI and GUI	Ethernet	Y

Command line interface

Access the CLI locally by connecting a workstation to the console port, or remotely by connecting through the secure shell (SSH) from any workstation on the same network.

Log on to the Command Line Interface through the Console Port

The appliance has a console port for connecting to a computer workstation. To log on to the appliance, you need a serial crossover cable and a workstation with a terminal emulation program.

To log on to the CLI through the console port, follow these steps:

1. Connect the console port to a serial port on the workstation. For more information, see Connect the console cable.
2. On the workstation, start HyperTerminal or any other terminal emulation program. If the logon prompt does not appear, you might have to press ENTER one or more times to display it.

3. In User Name, type nsroot. In Password, type nsroot and if that password does not work, try typing the serial number of the appliance. The serial number bar code is available at the back of the appliance.

Log on to the Command Line Interface by using SSH

The SSH protocol is the preferred remote access method for accessing an appliance remotely from any workstation on the same network. You can use either SSH version 1 (SSH1) or SSH version 2 (SSH2.)

If you do not have a working SSH client, you can download and install any of the following SSH client programs:

- PuTTY

  Open Source software supported on multiple platforms. Available at:

  http://www.chiark.greenend.org.uk/~sgtatham/putty/

- Vandyke Software SecureCRT

  Commercial software supported on the Windows platform. Available at:

  http://www.vandyke.com/products/securecrt/

These programs are tested by the Citrix ADC team, who have verified that they work correctly with a Citrix ADC appliance. Other programs might also work correctly, but have not been tested.

To verify that the SSH client is installed properly, use it to connect to any device on your network that accepts SSH connections.

To log on to a Citrix ADC appliance by using an SSH client, follow these steps:

1. On your workstation, start the SSH client.
2. For initial configuration, use the default IP address (NSIP), which is 192.168.100.1. For subsequent access, use the NSIP that was assigned during initial configuration. Select either SSH1 or SSH2 as the protocol.
3. In User Name, type nsroot. In Password, type nsroot and if that password does not work, try typing the serial number of the appliance. The serial number bar code is available at the back of the appliance. For example.

```
login as: nsroot


Using keyboard-interactive authentication.


Password:

```



```
9
10 Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9
11
12
13
14
15
16 Done
17
18
19 >
20
21 <!--NeedCopy-->
```

Citrix ADC GUI

> **Important:**
>
> A certificate-key pair is required for HTTPS access to the Citric ADC GUI. On the ADC, a certificate-key pair is automatically bound to the internal services. On an MPX or SDX appliance, the default key size is 1024 bytes, and on a VPX instance, the default key size is 512 bytes. However, most browsers today do not accept a key that is less than 1024 bytes. As a result, HTTPS access to the VPX configuration utility is blocked.

Also, if a license is not present on an MPX appliance when it starts, and you add a license later and restart the appliance, you might lose the certificate binding.

Citrix recommends that you install a certificate-key pair of at least 1024 bytes on the appliance for HTTPS access to the GUI. Also, install an appropriate license before starting the appliance.

The GUI includes a configuration utility and a statistical utility, called Dashboard, either of which you access through a workstation connected to an Ethernet port on the appliance.

The system requirements for the workstation running the GUI are as follows:

- For Windows-based workstations, a Pentium 166 MHz or faster processor.
- For Linux-based workstations, a Pentium platform running Linux kernel v2.2.12 or above, and `glibc` version 2.12–11 or later. A minimum of 32 MB RAM is required, and 48 MB RAM is recommended. The workstation must support 16-bit color mode, KDE, and KWM window managers used in conjunction, with displays set to local hosts.
- For Solaris-based workstations, a Sun running either Solaris 2.6, Solaris 7, or Solaris 8.

Your workstation must have a supported web browser to access the configuration utility and Dashboard.

The following browsers are supported.

Operating System: Windows 7

Browser: Internet Explorer (version 9, 10, and 11), Mozilla Firefox (version 3.6.25 and above), Google Chrome (latest).

Operating System: Windows 64 bit

Browser: Internet Explorer (version 8, 9, 10, and 11), Google Chrome (version latest)

Operating System: MAC
Browser: Mozilla Firefox (version 3.6.25 and above), Safari (version 5.1.3 and above), Google Chrome (version latest)

Use the Citrix ADC GUI

Once you log on to the configuration utility, you can configure the appliance through a graphical interface that includes context-sensitive help.

To log on to the GUI, follow these steps:

1. Open your web browser and enter the Citrix ADC IP (NSIP) as an HTTP address. If you have not yet set up the initial configuration, enter the default NSIP (http://192.168.100.1). The Citrix Logon page appears.

   Note: If you have two Citrix ADC appliances in a high availability setup, do not access the GUI by entering the IP address of the secondary Citrix ADC appliance. If you do so and use the GUI to configure the secondary appliance, your configuration changes are not applied to the primary Citrix ADC appliance.

2. In the User Name text box, type `nsroot`.
3. In the Password text box, type the administrative password you assigned to the `nsroot` account during initial configuration and click **Login**. If that password does not work, try typing the serial number of the appliance. The serial number bar code is available at the back of the appliance.

   To access the online help, select Help from the Help menu at the top right corner.

Use the Statistical Utility

Dashboard, the statistical utility, is a browser-based application that displays charts and tables on which you can monitor the performance of a Citrix ADC appliance.

To log on to Dashboard, follow these steps:

1. Open your web browser and enter the NSIP as an HTTP address. The Citrix Logon page appears.
2. In the User Name text box, type `nsroot`.

3. In the Password text box, type the administrative password you assigned to the `nsroot` account during initial configuration. If that password does not work, try typing the serial number of the appliance. The serial number bar code is available at the back of the appliance.

Configure the ADC for the first time

September 14, 2021

For initial configuration of a Citrix ADC MPX appliance, see Initial Configuration of a Citrix MPX appliance.

For initial configuration of a Citrix SDX appliance, see Initial Configuration of a Citrix SDX appliance.

NITRO API

You can use the NITRO API to configure the Citrix ADC appliance. NITRO exposes its functionality through Representational State Transfer (REST) interfaces. Therefore, NITRO applications can be developed in any programming language. Additionally, for applications that must be developed in Java or .NET or Python, NITRO APIs are exposed through relevant libraries that are packaged as separate Software Development Kits (SDKs). For more information, see NITRO API.

Secure your Citrix ADC deployment

September 14, 2021

To maintain security through the deployment life cycle of Citrix ADC appliance, Citrix recommends you to consider the following security aspects:

- Physical Security
- Appliance Security
- Network Security
- Administration and Management

Different deployments might require different security considerations. The Citrix ADC secure deployment guidelines provide general security guidance to help you decide on an appropriate secure deployment based on your specific security requirements.

For more information on guidelines for securely deploying the Citrix ADC appliance, see Citrix ADC secure deployment guidelines.

Configure high availability

September 14, 2021

You can deploy two Citrix ADC appliances in a high availability configuration, where one unit actively accepts connections and manages servers while the secondary unit monitors the first. The Citrix ADC appliance that is actively accepting connections and managing the servers is called a primary unit and the other one is called a secondary unit in a high availability configuration. If there is a failure in the primary unit, the secondary unit becomes the primary and begins actively accepting connections.

Each Citrix ADC appliance in a high availability pair monitors the other by sending periodic messages, called heartbeat messages or health checks, to determine the health or state of the peer node. If a health check for a primary unit fails, the secondary unit retries the connection for a specific time period. For more information about high availability, see High Availability. If a retry does not succeed by the end of the specified time period, the secondary unit takes over for the primary unit in a process called failover. The following figure shows two high availability configurations, one in one-arm mode and the other in two-arm mode.

Figure 1. High availability in one-arm mode

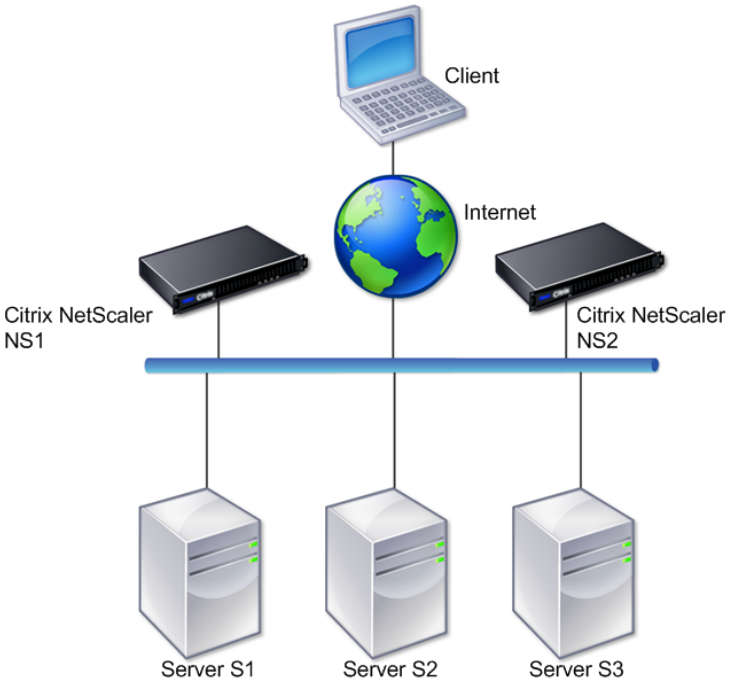
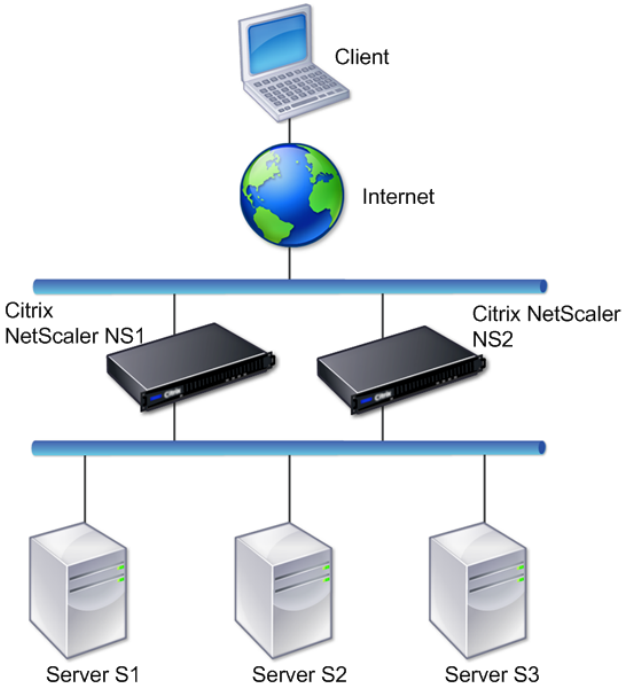


Figure 2. High availability in two-arm mode



In one-arm configuration, both NS1 and NS2 and servers S1, S2, and S3 are connected to the switch.

In two-arm configuration, both NS1 and NS2 are connected to two switches. The servers S1, S2, and S3 are connected to the second switch. The traffic between client and the servers passes through either NS1 or NS2.

To set up a high availability environment, configure one ADC appliance as primary and another as secondary. Perform the following tasks on each of the ADC appliances:

- Add a node.
- Disable high availability monitoring for unused interfaces.

Add a Node

A node is a logical representation of a peer Citrix ADC appliance. It identifies the peer unit by ID and NSIP. An appliance uses these parameters to communicate with the peer and track its state. When you add a node, the primary and secondary units exchange heartbeat messages asynchronously. The node ID is an integer that must not be greater than 64.

Through CLI

To add a node by using the command line interface, follow these steps:

At the command prompt, type the following commands to add a node and verify that the node has been added:

- add HA node <id> <IPAddress>
- show HA node <id>

Example

```
add HA node 0 10.102.29.170
 Done
> show HA node 0
1)      Node ID:      0
        IP:   10.102.29.200 (NS200)
        Node State: UP
        Master State: Primary
        SSL Card Status: UP
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 1:0:41:50 (days:hrs:min:
           sec)
<!--NeedCopy-->
```

Through GUI

To add a node by using the GUI, follow these steps:

1. Navigate to **System** > **High Availability**.
2. Click **Add** on the **Nodes** tab.
3. On the **Create HA Node** page, in the **Remote Node IP Address** text box, type the NSIP Address (for example, 10.102.29.170) of the remote node.
4. Ensure that the **Configure remote system to participate in High Availability setup** check box is selected. Provide the login credentials of the remote node in the text boxes under **Remote System Login Credentials**.
5. Select the **Turn off HA monitor on interfaces/channels that are down** check box to disable the HA monitor on interfaces that are down.

Verify that the node you added appears in the list of nodes in the Nodes tab.

Disable high availability monitoring for unused interfaces

The high availability monitor is a virtual entity that monitors an interface. You must disable the monitor for interfaces that are not connected or being used for traffic. When the monitor is enabled on an interface whose status is DOWN, the state of the node becomes NOT UP. In a high availability configuration, a primary node entering a NOT UP state might cause a high availability failover. An interface is marked DOWN under the following conditions:

- The interface is not connected
- The interface is not working properly
- The cable connecting the interface is not working properly

Through CLI

To disable the high availability monitor for an unused interface by using the command line interface, follow these steps:

At the command prompt, type the following commands to disable the high availability monitor for an unused interface and verify that it is disabled:

- set interface <id> -haMonitor OFF
- show interface <id>

  Example

```
1  > set interface 1/8 -haMonitor OFF
2   Done
3  > show interface 1/8
4          Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
5          flags=0x4000 <ENABLED, DOWN, down, autoneg, 802.1q>
6          MTU=1514, native vlan=1, MAC=00:d0:68:15:fd:3d, downtime
               238h55m44s
7          Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
8                    throughput 0
9
10         RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
11         TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
12         NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
               Muted(0)
13         Bandwidth thresholds are not set.
14 <!--NeedCopy-->
```

  When the high availability monitor is disabled for an unused interface, the output of the show interface command for that interface does not include “HAMON.”

Through GUI

To disable the high availability monitor for unused interfaces by using the GUI, follow these steps:

1. Navigate to System > Network > Interfaces.
2. Select the interface for which the monitor must be disabled.
3. Click Open. The Modify Interface dialog box appears.
4. In HA Monitoring, select the OFF option.
5. Click OK.
6. Verify that, when the interface is selected, “HA Monitoring: OFF” appears in the details at the bottom of the page.

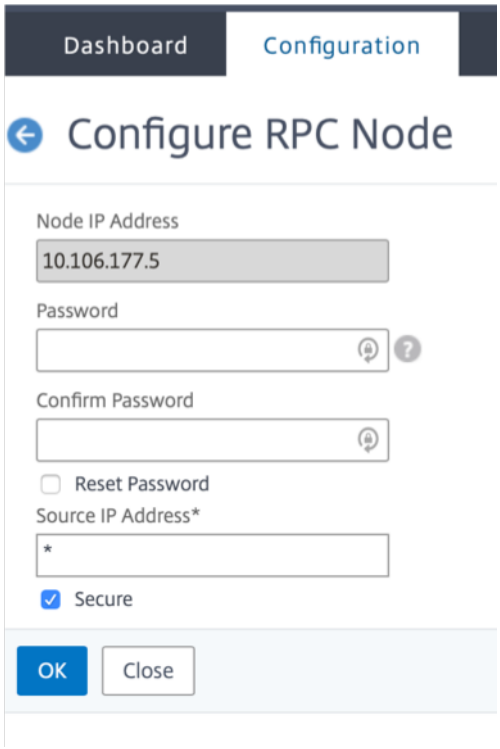
Change an RPC node password

September 14, 2021

To communicate with other Citrix ADC appliances, each appliance requires knowledge of the other appliances, including how to authenticate on Citrix ADC appliance. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. One RPC node exists on each Citrix ADC appliance and stores information, such as the IP addresses of the other Citrix ADC appliance and the passwords used for authentication. The Citrix ADC appliance that contacts the other Citrix ADC appliance checks the password within the RPC node.

To change an RPC node password by using the GUI

1. Navigate to **System** > **Network** > **RPC**.
2. In the **RPC** pane, select the node and then click **Edit**.
3. In **Configure RPC Node**, type the new password.
4. In **Source IP Address**, type the existing node’s IP address to be used to communicate with the peer system node.



5. Select **Secure** and then click **OK**.

> **Note**
>
> For enhanced security, Citrix recommends you to enable the **Secure** option on RPC nodes. When you enable the **Secure** option, the appliance encrypts all the RPC communication sent from one ADC node to other ADC nodes thus securing the RPC communication. This secure communication uses the port number 3008. If the firewall between the ADC nodes blocks the port number 3008, unblock it and proceed. Otherwise, configuration synchronization and configuration propagation might fail.

To change an RPC node password by using the CLI

At the command line, type the following commands:

```
set ns rpcNode <IPAddress> {
 -password }
  [-secure ( YES | NO )]
show ns rpcNode
<!--NeedCopy-->
```

Example:

```
> set ns rpcNode 192.0.2.4 -password mypassword -secure YES
 Done
```



```
> show rpcNode
.
.
.
 IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
     SrcIP:  *           Secure:  ON
Done
>

<!--NeedCopy-->
```

Configure a FIPS appliance for the first time

October 14, 2021

> **Note**
>
> - FIPS FAQ can be found here: FIPS FAQ.

A certificate-key pair is required for HTTPS access to the configuration utility and for secure remote procedure calls. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. One RPC node exists on each appliance. This node stores the password, which is checked against the one provided by the contacting appliance. To communicate with other Citrix ADC appliances, each appliance requires knowledge of the other appliances, including how to authenticate on the other appliance. RPC nodes maintain this information, which includes the IP addresses of the other Citrix ADC appliances and the passwords used to authenticate on each.

On a Citrix ADC MPX appliance virtual appliance, a certificate-key pair is automatically bound to the internal services. On a FIPS appliance, a certificate-key pair must be imported into the hardware security module (HSM) of a FIPS card. To do so, you must configure the FIPS card, create a certificate-key pair, and bind it to the internal services.

Configure secure HTTPS by using the CLI

To configure secure HTTPS by using the CLI, follow these steps

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see Configure the HSM.
2. If the appliance is part of a high availability setup, enable the SIM. For information about enabling the SIM on the primary and secondary appliances, see Configure FIPS appliances in a high availability setup.

3. Import the FIPS key into the HSM of the FIPS card of the appliance. At the command prompt, type:

   ```
   import ssl fipskey serverkey -key ns-server.key -inform PEM
   ```

4. Add a certificate-key pair. At the command prompt, type:

   ```
   add certkey server -cert ns-server.cert -fipskey serverkey
   ```

5. Bind the certificate-key created in the previous step to the following internal services. At the command prompt, type:

   ```
   bind ssl service nshttps-127.0.0.1-443 -certkeyname server
   ```

   ```
   bind ssl service nshttps-::11-443 -certkeyname server
   ```

Configure secure HTTPS by using the GUI

To configure secure HTTPS by using the GUI, follow these steps:

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see Configure the HSM.
2. If the appliance is part of a high availability setup, enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, see Configure FIPS appliances in a high availability setup.
3. Import the FIPS key into the HSM of the FIPS card of the appliance. For more information about importing a FIPS key, see the Import an existing FIPS key section.
4. Navigate to **Traffic Management > SSL > Certificates**.
5. In the details pane, click Install.
6. In the Install Certificate dialog box, type the certificate details.
7. Click Create, and then click Close.
8. Navigate to **Traffic Management > Load Balancing > Services**.
9. In the details pane, on the Action tab, click Internal Services.
10. Select `nshttps-127.0.0.1-443` from the list, and then click Open.
11. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
12. Select `nshttps-::11-443` from the list, and then click Open.
13. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
14. Click OK.

Configure secure RPC by using the CLI

To configure secure RPC by using the CLI, follow these steps:

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see Configure the HSM.
2. Enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, see Configure FIPS appliances in a high availability setup.
3. Import the FIPS key into the HSM of the FIPS card of the appliance. At the command prompt, type:

   ```
   import ssl fipskey serverkey -key ns-server.key -inform PEM
   ```

4. Add a certificate-key pair. At the command prompt, type:

   ```
   add certkey server -cert ns-server.cert -fipskey serverkey
   ```

5. Bind the certificate-key pair to the following internal services. At the command prompt, type:

   ```
   bind ssl service nsrpcs-127.0.0.1-3008 -certkeyname server
   ```

   ```
   bind ssl service nskrpcs-127.0.0.1-3009 -certkeyname server
   ```

   ```
   bind ssl service nsrpcs-::1l-3008 -certkeyname server
   ```

6. Enable secure RPC mode. At the command prompt, type:

   ```
   set ns rpcnode \<IP address\> -secure YES
   ```

   For more information about changing an RPC node password, see Change an RPC node password.

Configure secure RPC by using the GUI

To configure secure RPC by using the GUI, follow these steps:

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see Configure the HSM.
2. Enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, Configure FIPS appliances in a high availability setup.
3. Import the FIPS key into the HSM of the FIPS card of the appliance. For more information about importing a FIPS key, the Import an existing FIPS key section.
4. Navigate to **Traffic Management > SSL > Certificates**.
5. In the details pane, click Install.
6. In the Install Certificate dialog box, type the certificate details.
7. Click Create, and then click Close.
8. Navigate to **Traffic Management > Load Balancing > Services**.

9. In the details pane, on the Action tab, click Internal Services.
10. Select `nsrpcs-127.0.0.1-3008` from the list, and then click Open.
11. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
12. Select `nskrpcs-127.0.0.1-3009` from the list, and then click Open.
13. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
14. Select `nsrpcs-::11-3008` from the list, and then click Open.
15. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
16. Click OK.
17. Navigate to **System > Network > RPC**.
18. In the details pane, select the IP address, and click Open.
19. In the Configure RPC Node dialog box, select Secure.
20. Click OK.

Common network topologies

September 14, 2021

As described in the "Physical deployment mode" section in Where does a Citrix ADC appliance fit in the network?, you can deploy the Citrix ADC appliance either inline between the clients and servers or in one-arm mode. Inline mode uses a two-arm topology, which is the most common type of deployment.

Set up a common two-arm topology

In a two-arm topology, one network interface is connected to the client network and another network interface is connected to the server network, ensuring that all traffic flows through the appliance. This topology might require you to reconnect your hardware and also might result in a momentary downtime. The basic variations of two-arm topology are multiple subnets, typically with the appliance on a public subnet and the servers on a private subnet, and transparent mode, with both the appliance and the servers on the public network.

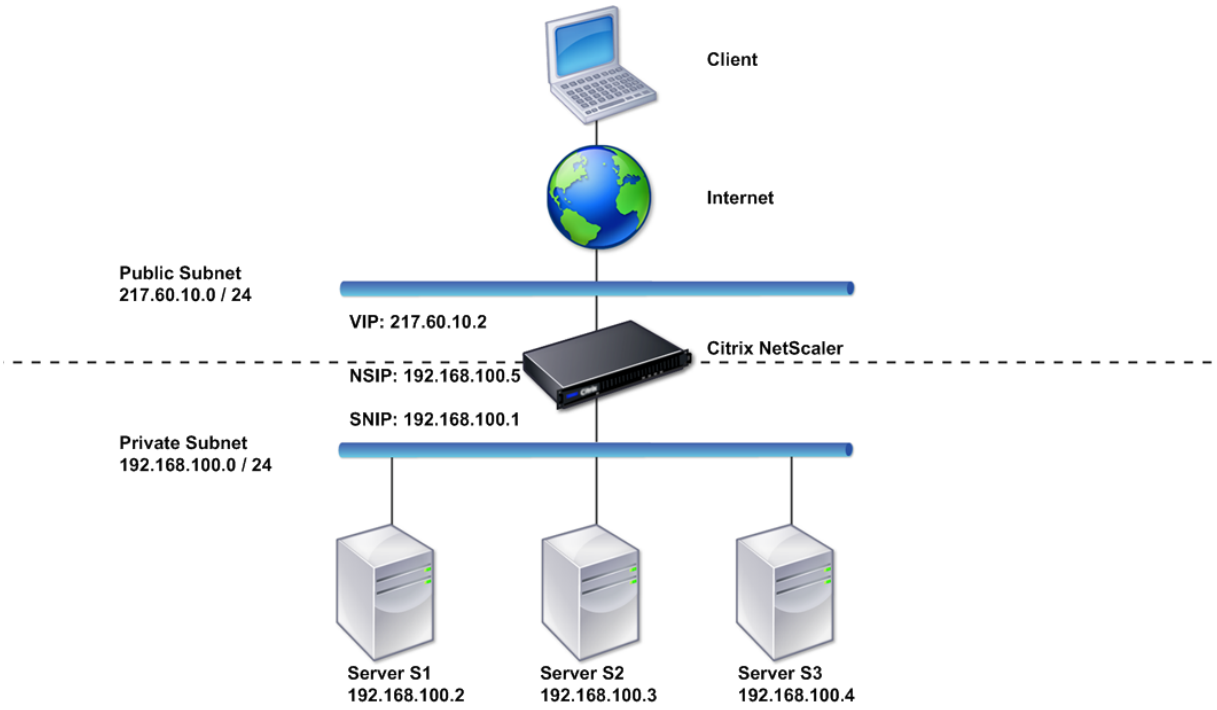
Set up a simple two-arm multiple subnet topology

One of the most commonly used topologies has the Citrix ADC appliance inline between the clients and the servers, with a virtual server configured to handle the client requests. This configuration is used when the clients and servers reside on different subnets. In most cases, the clients and servers reside on public and private subnets, respectively.

For example, consider an appliance deployed in two-arm mode for managing servers S1, S2, and S3, with a virtual server of type HTTP configured on the appliance, and with HTTP services running on the servers. The servers are on a private subnet and a SNIP is configured on the appliance to communicate with the servers. The Use SNIP (USNIP) option must be enabled on the appliance so that it uses the SNIP instead of the MIP.

As shown in the following figure, the VIP is on public subnet 217.60.10.0, and the NSIP, the servers, and the SNIP are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for Two-Arm Mode, Multiple Subnets



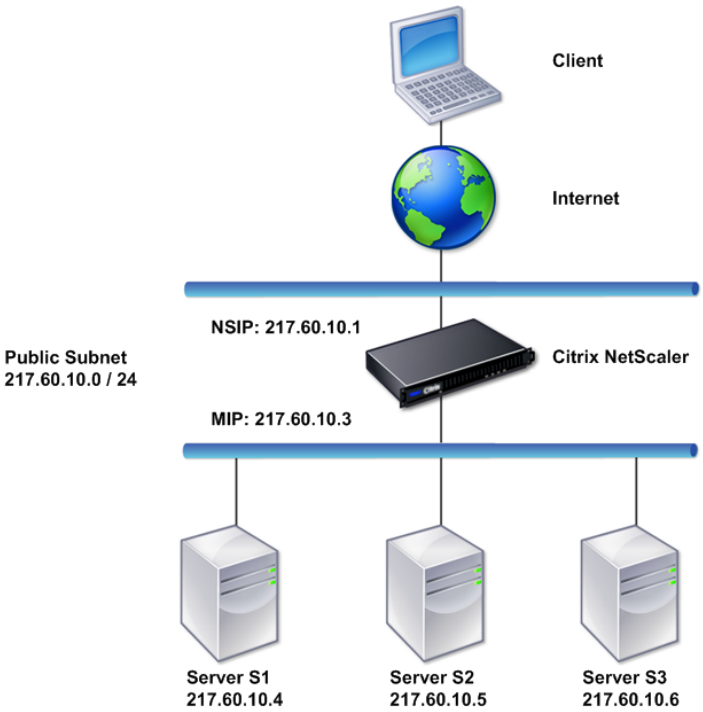
To deploy a Citrix ADC appliance in two-arm mode with multiple subnets, follow these steps:

1. Configure the NSIP and default gateway, as described in Configuring the NetScaler IP Address (NSIP).
2. Configure the SNIP, as described in Configuring Subnet IP Addresses.
3. Enable the USNIP option, as described in To enable or disable USNIP mode section.
4. Configure the virtual server and the services, as described in Creating a Virtual Server section and Configuring Services section.
5. Connect one of the network interfaces to a private subnet and the other interface to a public subnet.

Set up a simple two-arm transparent topology

Use transparent mode if the clients need to access the servers directly, with no intervening virtual server. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a Citrix ADC appliance is placed between the client and the server, so the traffic must pass through the appliance. You must enable L2 mode for bridging the packets. The NSIP and MIP are on the same public subnet, 217.60.10.0/24.

Figure 2. Topology Diagram for Two-Arm, Transparent Mode



To deploy a Citrix ADC appliance in two-arm, transparent mode, follow these steps

1. Configure the NSIP and default gateway, as described in Configuring the NetScaler IP Address (NSIP).
2. Enable L2 mode, as described in Enabling and Disabling Layer 2 Mode.
3. Configure the default gateway of the managed servers as the MIP.
4. Connect the network interfaces to the appropriate ports on the switch.

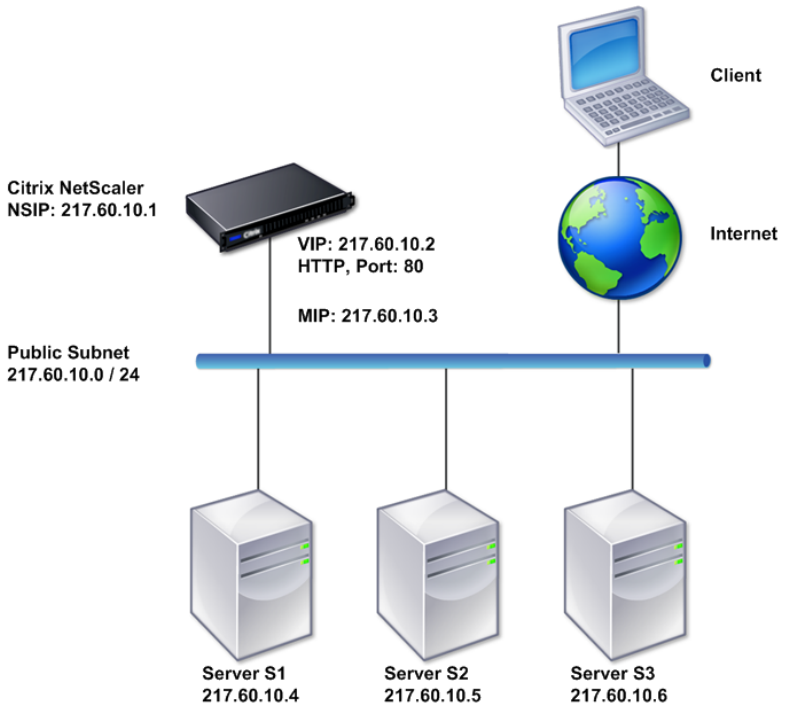
Set up common one-arm topologies

The two basic variations of one-arm topology are with a single subnet and with multiple subnets.

Set up a simple one-arm single subnet topology

You can use a one-arm topology with a single subnet when the clients and servers reside on the same subnet. For example, consider a Citrix ADC appliance deployed in one-arm mode for managing servers S1, S2, and S3. A virtual server of type HTTP is configured on an ADC appliance, and HTTP services are running on the servers. As shown in the following figure, the Citrix ADC IP address (NSIP), the Mapped IP address (MIP), and the server IP addresses are on the same public subnet, 217.60.10.0/24.

Figure 3. Topology Diagram for One-Arm Mode, Single Subnet



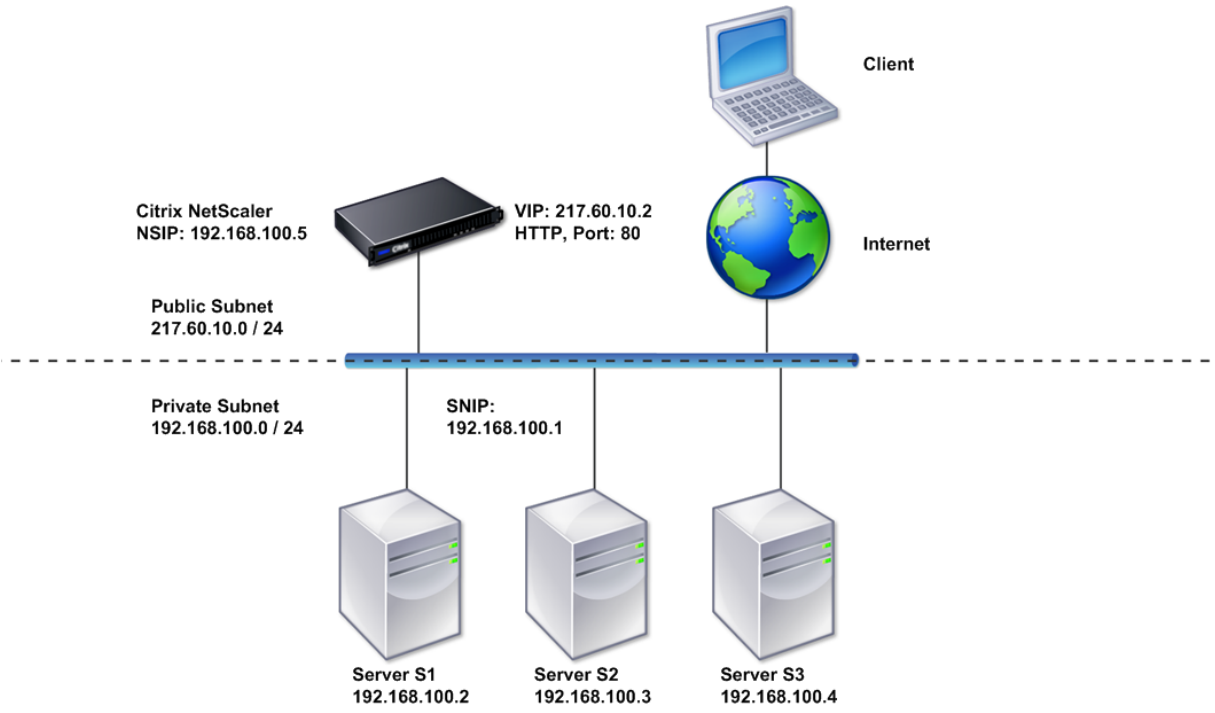
To deploy a Citrix ADC appliance in one-arm mode with a single subnet, follow these steps:

1. Configure the NSIP and the default gateway, as described in, as described in Configuring the Citrix ADC IP Address (NSIP).
2. Configure the virtual server and the services, as described in Creating a Virtual Server section and Configuring Services section.
3. Connect one of the network interfaces to the switch.

Set up a simple one-arm multiple subnet topology

You can use a one-arm topology with multiple subnets when the clients and servers reside on the different subnets. For example, consider a Citrix ADC appliance deployed in one-arm mode for managing servers S1, S2, and S3, with the servers connected to switch SW1 on the network. A virtual server of type HTTP is configured on the appliance, and HTTP services are running on the servers. These three servers are on the private subnet, so a subnet IP address (SNIP) is configured to communicate with them. The Use Subnet IP address (USNIP) option must be enabled so that the appliance uses the SNIP instead of a MIP. As shown in the following figure, the virtual IP address (VIP) is on public subnet 217.60.10.0/24; the NSIP, SNIP, and the server IP addresses are on private subnet 192.168.100.0/24.

Figure 4. Topology Diagram for One-Arm Mode, Multiple Subnets



To deploy a Citrix ADC appliance in one-arm mode with multiple subnets, follow these steps:

1. Configure the NSIP and the default gateway, as described in Configuring the NetScaler IP Address (NSIP).
2. Configure the SNIP and enable the USNIP option, as described in Configuring Subnet IP Addresses.
3. Configure the virtual server and the services, as described in Creating a Virtual Server section and Configuring Services section.

4. Connect one of the network interfaces to the switch.

System management settings

September 14, 2021

Once your initial configuration is in place, you can configure settings to define the behavior of the Citrix ADC appliance and facilitate connection management. You have a number of options for handling HTTP requests and responses. Routing, bridging, and MAC based forwarding modes are available for handling packets not addressed to the Citrix ADC appliance. You can define the characteristics of your network interfaces and can aggregate the interfaces. To prevent timing problems, you can synchronize the Citrix clock with a Network Time Protocol (NTP) server. The Citrix ADC appliance can operate in various DNS modes, including as an authoritative domain name server (ADNS). You can set up SNMP for system management and customize syslog logging of system events. Before deployment, verify that your configuration is complete and correct.

System settings

September 14, 2021

Configuration of system settings includes basic tasks such as configuring HTTP ports to enable connection keep-alive and server offload, setting the maximum number of connections for each server, and setting the maximum number of requests per connection. You can enable client IP address insertion for situations in which a proxy IP address is not suitable, and you can change the HTTP cookie version.

You can also configure a Citrix ADC appliance to open FTP connections on a controlled range of ports instead of ephemeral ports for data connections. This improves security, because opening all ports on the firewall is insecure. You can set the range anywhere from 1,024 to 64,000.

Before deployment, go through the verification checklists to verify your configuration. To configure HTTP parameters and the FTP port range, use the Citrix ADC GUI.

You can modify the types of HTTP parameters described in the following table.

Parameter Type: HTTP Port Information

Specifies: The web server HTTP ports used by your managed servers. If you specify the ports, the appliance performs request switching for any client request that has a destination port matching a specified port.

> Note: If an incoming client request is not destined for a service or a virtual server that is specifically configured on the appliance, the destination port in the request must match one of the globally configured HTTP ports. This allows the appliance to perform connection keep-alive and server off-load.

Parameter Type: Limits

Specifies: The maximum number of connections to each managed server, and the maximum number of requests sent over each connection. For example, if you set Max Connections to 500, and the appliance is managing three servers, it can open a maximum of 500 connections to each of the three servers. By default, the appliance can create an unlimited number of connections to any of the servers it manages. To specify an unlimited number of requests per connection, set Max Requests to 0.

> Note: If you are using the Apache HTTP server, you must set Max Connections equal to the value of the MaxClients parameter in the Apache httpd.conf file. Setting this parameter is optional for other web servers.

Parameter Type: Client IP Insertion

Specifies: Enable/disable insertion of the client's IP address into the HTTP request header. You can specify a name for the header field in the adjacent text box. When a web server managed by an appliance receives a subnet IP address, the server identifies it as the client's IP address. Some applications need the client's IP address for logging purposes or to dynamically determine the content to be served by the web server.

You can enable insertion of the actual client IP address into the HTTP header request sent from the client to one, some, or all servers managed by the appliance. You can then access the inserted address through a minor modification to the server (using an Apache module, ISAPI interface, or NSAPI interface).

Parameter Type: Cookie Version

Specifies: The HTTP cookie version to use when COOKIEINSERT persistence is configured on a virtual server. The default, version 0, is the most common type on the Internet. Alternatively, you can specify version 1.

Parameter Type: Requests/Responses

Specifies: Options for handling certain types of requests, and enable/disable logging of HTTP error responses.

Parameter Type: Server Header Insertion

Specifies: Insert a server header in Citrix ADC-generated HTTP responses.

To configure HTTP parameters by using the GUI, follow these steps:

1. In the navigation pane, expand **System**, and then click **Settings**.

2. In the details pane, under **Settings**, click **Change HTTP parameters**.
3. In the **Configure HTTP parameters** dialog box, specify values for some or all of the parameters that appear under the headings listed in the table above.
4. Click **OK**.

To set the FTP port range by using the GUI, follow these steps:

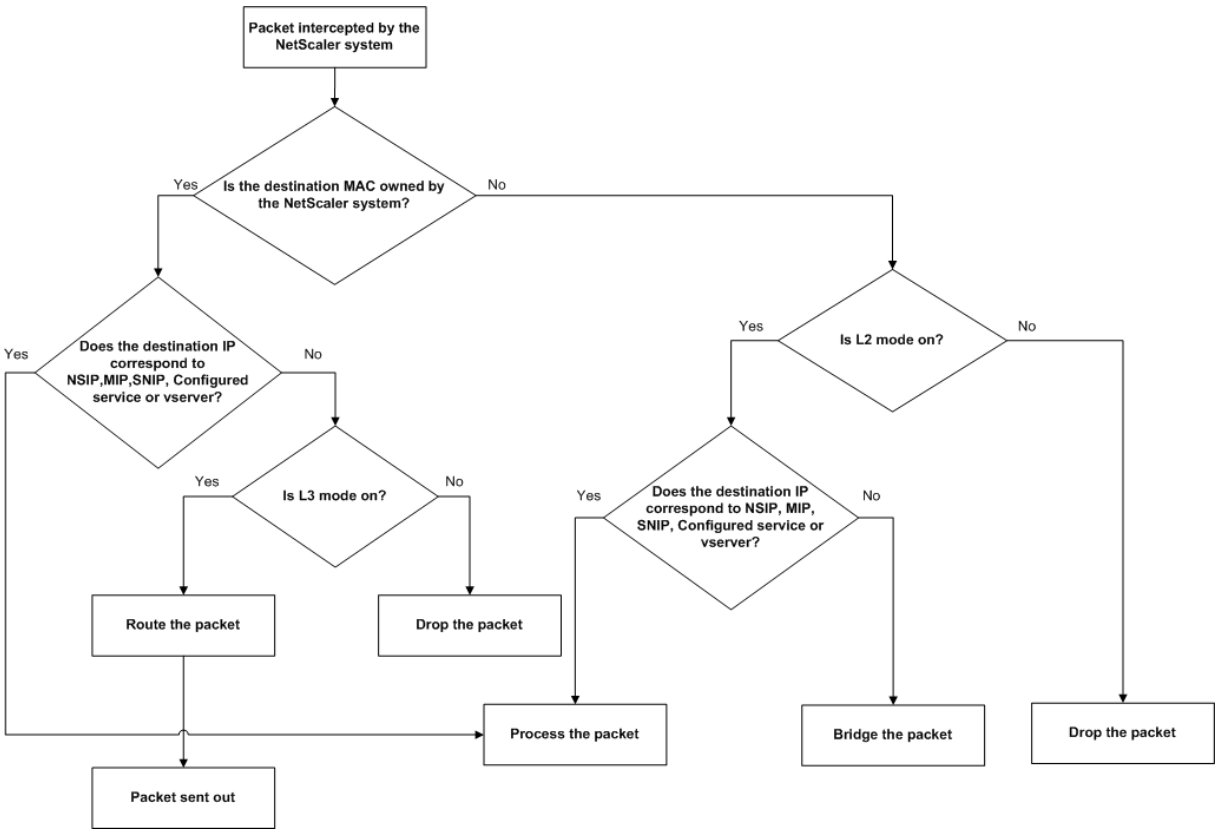
1. In the navigation pane, expand **System**, and then click **Settings**
2. In the details pane, under **Settings**, click **Change global system settings**.
3. Under **FTP Port Range,** in the **Start Port** and **End Port** text boxes, type the lowest and highest port numbers, respectively, for the range you want to specify (for example, 5000 and 6000).
4. Click **OK**.

Packet forwarding modes

September 14, 2021

The Citrix ADC appliance can either route or bridge packets that are not destined for an IP address owned by the appliance (that is, the IP address is not the NSIP, a MIP, a SNIP, a configured service, or a configured virtual server). By default, L3 mode (routing) is enabled and L2 mode (bridging) is disabled, but you can change the configuration. The following flow chart shows how the appliance evaluates packets and either processes, routes, bridges, or drops them.

Figure 1. Interaction between Layer 2 and Layer 3 Modes



An appliance can use the following modes to forward the packets it receives:

- Layer 2 (L2) Mode
- Layer 3 (L3) Mode
- MAC-Based Forwarding Mode

Enable and disable layer 2 mode

Layer 2 mode controls the Layer 2 forwarding (bridging) function. You can use this mode to configure a Citrix ADC appliance to behave as a Layer 2 device and bridge the packets that are not destined for it. When this mode is enabled, packets are not forwarded to any of the MAC addresses, because the packets can arrive on any interface of the appliance and each interface has its own MAC address.

With Layer 2 mode disabled (which is the default), the appliance drops packets that are not destined for one of its MAC address. If another Layer 2 device is installed in parallel with the appliance, Layer 2 mode must be disabled to prevent bridging (Layer 2) loops. You can use the configuration utility or the command line to enable Layer 2 mode.

Note: The appliance does not support spanning tree protocol. To avoid loops, if you enable L2 mode, do not connect two interfaces on the appliance to the same broadcast domain.

To enable or disable Layer 2 mode by using the CLI

At the command prompt, type the following commands to enable/disable Layer 2 mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

Examples

```
    \> enable ns mode l2
    Done
    \> show ns mode

    Mode Acronym Status
    \------- ------- ------
    1\) Fast Ramp FR ON
    2\) Layer 2 mode L2 ON
    .
    .
    .
    Done
    \>

    \> disable ns mode l2
    Done
    \> show ns mode

    Mode Acronym Status
    \------- ------- ------
    1\) Fast Ramp FR ON
    2\) Layer 2 mode L2 OFF
    .
    .
    .
    Done
    \>
<!--NeedCopy-->
```

To enable or disable Layer 2 mode by using the GUI

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes** and **Features**, click **Configure modes**.

3. In the **Configure Modes** dialog box, to enable Layer 2 mode, select the **Layer 2 Mode** check box. To disable Layer 2 mode, clear the check box.
4. Click **OK**. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click **Yes**.

Enable and disable layer 3 mode

Layer 3 mode controls the Layer 3 forwarding function. You can use this mode to configure a Citrix ADC appliance to look at its routing table and forward packets that are not destined for it. With Layer 3 mode enabled (which is the default), the appliance performs route table lookups and forwards all packets that are not destined for any appliance-owned IP address. If you disable Layer 3 mode, the appliance drops these packets.

To enable or disable Layer 3 mode by using the CLI

At the command prompt, type the following commands to enable/disable Layer 3 mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

Examples

```
\> enable ns mode l3
Done
\> show ns mode

Mode Acronym Status
\------- ------- ------
1\) Fast Ramp FR ON
2\) Layer 2 mode L2 OFF
.
.
.
9\) Layer 3 mode (ip forwarding) L3 ON
.
.
.
Done
\>

\> disable ns mode l3
```



```
20      Done
21      \> show ns mode
22
23      Mode Acronym Status
24      \------- ------- ------
25      1\) Fast Ramp FR ON
26      2\) Layer 2 mode L2 OFF
27      .
28      .
29      .
30      9\) Layer 3 mode (ip forwarding) L3 OFF
31      .
32      .
33      .
34      Done
35      \>
36  <!--NeedCopy-->
```

To enable or disable Layer 3 mode by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In the Configure Modes dialog box, to enable Layer 3 mode, select the Layer 3 Mode (IP Forwarding) check box. To disable Layer 3 mode, clear the check box.
4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

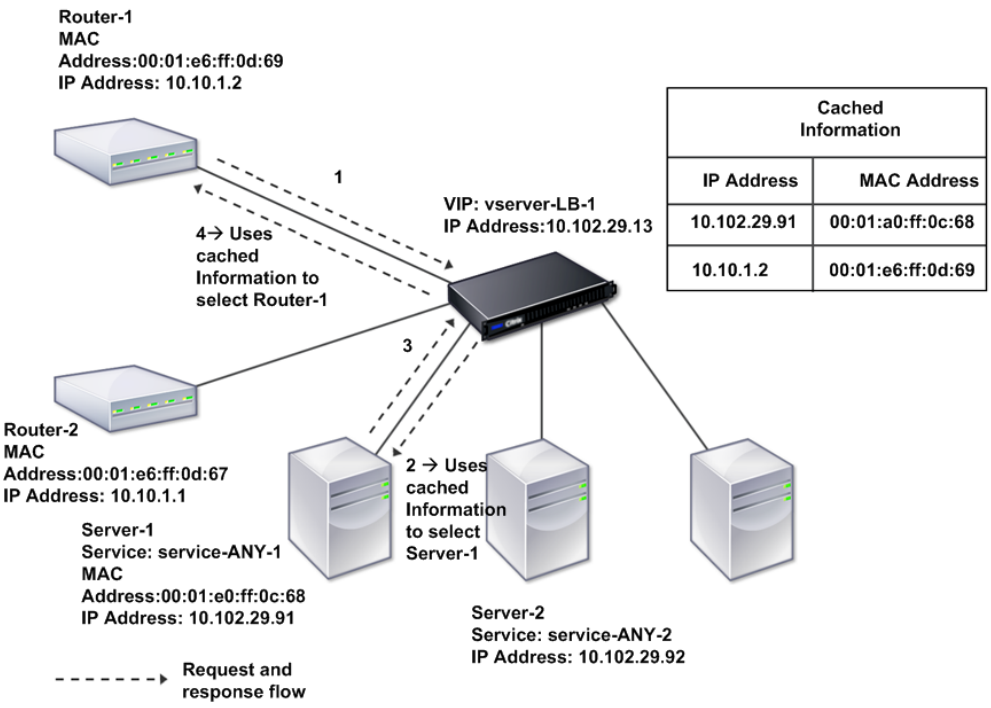
Enable and disable MAC based forwarding mode

You can use MAC-based forwarding to process traffic more efficiently and avoid multiple-route or ARP lookups when forwarding packets, because the Citrix ADC appliance remembers the MAC address of the source. To avoid multiple lookups, the appliance caches the source MAC address of every connection for which it performs an ARP lookup, and it returns the data to the same MAC address.

MAC-based forwarding is useful when you use VPN devices because the appliance ensures that all traffic flowing through a particular VPN passes through the same VPN device.

The following figure shows the process of MAC-based forwarding.

Figure 2. MAC-Based Forwarding Process



When MAC-based forwarding is enabled, the appliance caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server responds through an appliance, the appliance sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when an appliance initiates a connection, it uses the route and ARP tables for the lookup function. To enable MAC-based forwarding, use the configuration utility or the command line.

Some deployments require the incoming and outgoing paths to flow through different routers. In these situations, MAC-based forwarding breaks the topology design. For a global server load balancing (GSLB) site that requires the incoming and outgoing paths to flow through different routers, you must disable MAC-based forwarding and use the appliance's default router as the outgoing router.

With MAC-based forwarding disabled and Layer 2 or Layer 3 connectivity enabled, a route table can specify separate routers for outgoing and incoming connections. To disable MAC-based forwarding, use the configuration utility or the command line.

To enable or disable MAC-based forwarding by using the CLI

At the command prompt, type the following commands to enable/disable MAC-based forwarding mode and verify that it has been enabled/disabled:

- <enable ns mode <Mode>
- <disable ns mode <Mode>
- <show ns mode
  Example

  “‘ pre codeblock

  ```
  enable ns mode mbf
  Done
  show ns mode
  ```

  ```
  1   Mode                          Acronym           Status
  2   -------                       -------           ------   1)  Fast
      Ramp                     FR                ON   2)  Layer 2
      mode                  L2                OFF   .   .   .   6)
      MAC-based forwarding        MBF               ON   .   .   .
        Done  >
  ```

  ```
  disable ns mode mbf
  Done
  show ns mode
  ```

  ```
  1   Mode                          Acronym           Status
  2   -------                       -------           ------   1)  Fast
      Ramp                     FR                ON   2)  Layer 2
      mode                  L2                OFF   .   .   .   6)
      MAC-based forwarding        MBF               OFF   .   .   .
         Done  >  <!--NeedCopy--> ```
  ```

To enable or disable MAC-based forwarding by using the GUI

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features** group, click **Configure modes**.
3. In the **Configure Modes** dialog box, to enable MAC-based forwarding mode, select the **MAC Based Forwarding** check box. To disable MAC-based forwarding mode, clear the check box.
4. Click **OK**. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click **Yes**.

Network interfaces

September 14, 2021

The Citrix ADC interfaces are numbered in slot/port notation. In addition to modifying the characteristics of individual interfaces, you can configure virtual LANs to restrict traffic to specific groups of hosts. You can also aggregate links into high-speed channels.

Virtual LANs

The Citrix ADC appliance supports (Layer 2) port and IEEE802.1Q tagged virtual LANs (VLANs). VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface to belong to multiple VLANs by using IEEE 802.1q tagging.

You can bind your configured VLANs to IP subnets. The ADC appliance (if it is configured as the default router for the hosts on the subnets) then performs IP forwarding between these VLANs.

The Citrix ADC appliance supports the following types of VLANs.

- Default VLAN

  By default, the network interfaces on a Citrix ADC appliance are included in a single, port-based VLAN as untagged network interfaces. This default VLAN has a VID of 1 and exists permanently. It cannot be deleted, and its VID cannot be changed.

- Port-Based VLANs

  A set of network interfaces that share a common, exclusive, Layer 2 broadcast domain define the membership of a port-based VLAN. You can configure multiple port-based VLANs. When you add an interface to a new VLAN as an untagged member, it is automatically removed from the default VLAN.

- Tagged VLAN

  A network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of only one VLAN (its native VLAN). The untagged network interface forwards the frames for the native VLAN as untagged frames. A tagged network interface can be a part of more than one VLAN. When you configure tagging, be sure that both ends of the link have matching VLAN settings. You can use the configuration utility to define a tagged VLAN (nsvlan) that can have any ports bound as tagged members of the VLAN. Configuring this VLAN requires a reboot of the ADC appliance and therefore must be done during initial network configuration.

Link aggregate channels

Link aggregation combines incoming data from multiple ports into a single high speed link. Configuring the link aggregate channel increases the capacity and availability of the communication channel between a Citrix ADC appliance and other connected devices. An aggregated link is also referred to as a channel.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. A network interface can be bound to only one channel. Binding a network interface to a link aggregate channel changes the VLAN configuration. That is, binding network interfaces to a channel removes them from the VLANs that they originally belonged to and adds them to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you have bound network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then you bind them to link aggregate channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them to VLAN 2.

Note: You can also use Link Aggregation Control Protocol (LACP) to configure link aggregation. For more information, see Configuring Link Aggregation by Using the Link Aggregation Control Protocol.

Clock synchronization

September 14, 2021

You can configure your Citrix ADC appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network. NTP uses User Datagram Protocol (UDP) port 123 as its transport layer. You have to add NTP servers in the NTP configuration file so that the appliance periodically gets updates from these servers.

If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site at http://www.ntp.org.

To configure clock synchronization on your appliance, follow these steps:

1. Log on to the command line and enter the shell command.
2. At the shell prompt, copy the ntp.conf file from the /etc directory to the /nsconfig directory. If the file already exists in the /nsconfig directory, make sure that you remove the following entries from the ntp.conf file:

   ```
   restrict localhost

   restrict 127.0.0.2
   ```


These entries are required only if you want to run the device as a time server. However, this feature is not supported on the Citrix ADC appliance.

3. Edit /nsconfig/ntp.conf by typing the IP address for the desired NTP server under the file's server and restrict entries.
4. Create a file named rc.netscaler in the /nsconfig directory, if the file does not already exist in the directory.
5. Edit /nsconfig/rc.netscaler by adding the following entry: `/usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &`

   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   Note: If the time difference between the Citrix ADC appliance and the time server is more than 1000 sec, the ntpd service terminates with a message to the ADC log. To avoid this, you need to start ntpd with the -g option, which forcibly syncs the time. Add the following entry in /nsconfig/rc.netscaler:

   ```
   /usr/sbin/ntpd -g -c /nsconfig/ntp.conf -l /var/log/ntpd.log &
   ```

   If you do not want to forcibly sync the time when there is a large difference, you can set the date manually and then start ntpd again. You can check the time difference between the appliance and the time server by running the following command in the shell:

   ```
   1 ntpdate -q <IP address or domain name of the NTP server>
   2 <!--NeedCopy-->
   ```

6. Reboot the appliance to enable clock synchronization.

   Note: If you want to start time synchronization before you restart the appliance, enter the following command (which you added to the rc.netscaler file in step 5) at the shell prompt:

   ```
   1 /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ ntpd.log &
   2 <!--NeedCopy-->
   ```

DNS configuration

September 14, 2021

You can configure a Citrix ADC appliance to function as an Authoritative Domain Name Server (ADNS), DNS proxy server, End Resolver, or Forwarder. You can add DNS resource records such as SRV Records, AAAA Records, A Records, MX Records, NS Records, CNAME Records, PTR Records, and SOA Records. Also, the appliance can balance the load on external DNS servers.

A common practice is to configure an appliance as a forwarder. For this configuration, you need to add external name servers. After you have added the external servers, you should verify that your configuration is correct.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

When adding name servers, you can specify IP addresses or virtual IP addresses (VIPs). If you use IP addresses, the appliance load balances requests to the configured name servers in a round robin manner. If you use VIPs, you can specify any load balancing method.

Add a name server by using the CLI

At the command prompt, type the following commands to add a name server and verify the configuration:

- <add dns nameServer <IP>
- <show dns nameServer <IP>

  Example

  “‘ pre codeblock

  ```
  add dns nameServer 10.102.29.10
  Done
  show dns nameServer 10.102.29.10
  1) 10.102.29.10 - State: DOWN
  Done
  “‘
  ```

Add a name server by using the GUI

1. Navigate to **Traffic Management** > **DNS** > **Name Servers**.
2. In the details pane, click **Add**.
3. In the **Create Name Server** dialog box, select **IP Address**.
4. In the **IP Address** text box, type the IP address of the name server (for example, 10.102.29.10). If you are adding an external name server, clear the **Local** check box.
5. Click **Create**, and then click **Close**.
6. Verify that the name server you added appears in the **Name Servers** pane.

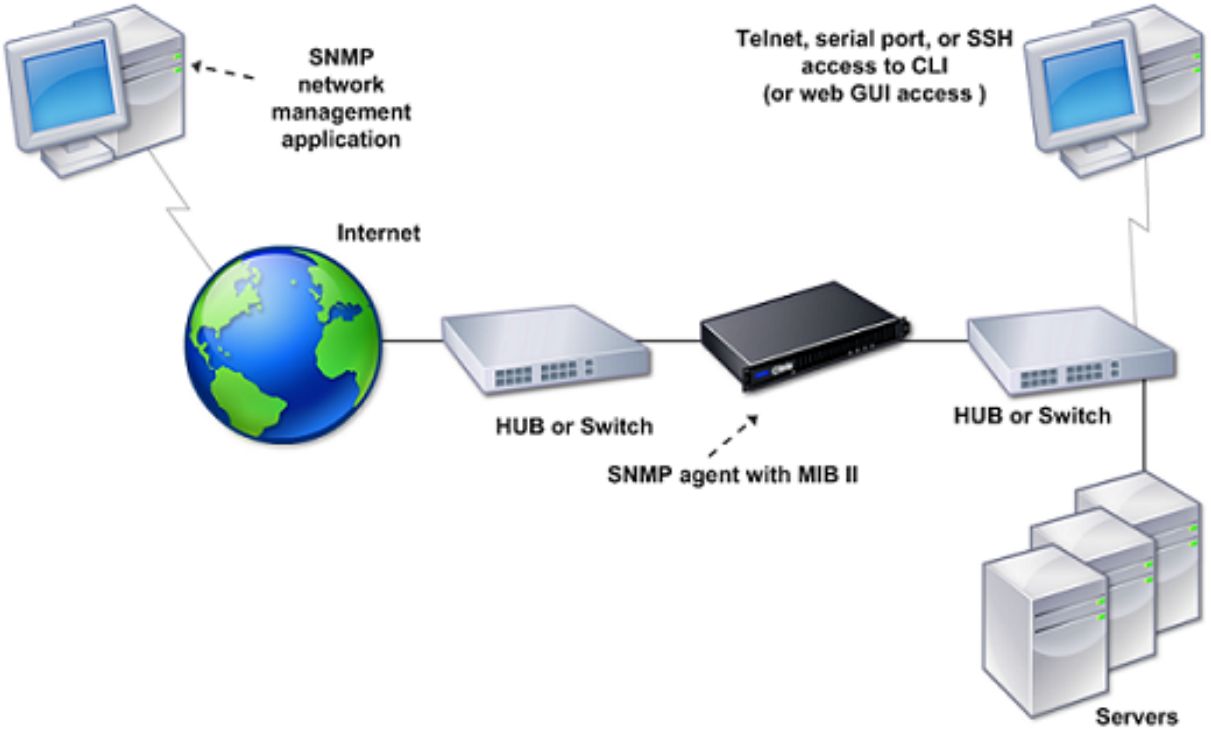
SNMP configuration

September 14, 2021

The Simple Network Management Protocol (SNMP) network management application, running on an external computer, queries the SNMP agent on the Citrix ADC appliance. The agent searches the management information base (MIB) for data requested by the network management application and sends the data to the application.

SNMP monitoring uses traps messages and alarms. SNMP traps messages are asynchronous events that the agent generates to signal abnormal conditions, which are indicated by alarms. For example, if you want to be informed when CPU utilization is above 90 percent, you can set up an alarm for that condition. The following figure shows a network with a Citrix ADC appliance that has SNMP enabled and configured.

Figure 1. SNMP on the Citrix ADC appliance



The SNMP agent on a Citrix ADC appliance supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3). Because it operates in bilingual mode, the agent can handle SNMPv2 queries, such as Get-Bulk, and SNMPv1 queries. The SNMP agent also sends traps compliant with SNMPv2 and supports SNMPv2 data types, such as counter64. SNMPv1 managers (programs on other servers that request SNMP information from the ADC appliance) use the NS-MIB-smiv1.mib file when processing SNMP queries. SNMPv2 managers use the NS-MIB-smiv2.mib file.

The Citrix ADC appliance supports the following enterprise-specific MIBs:

- A subset of standard MIB-2 groups. Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.
- A system enterprise MIB. Provides system-specific configuration and statistics.

To configure SNMP, you specify which managers can query the SNMP agent, add SNMP trap listeners that will receive the SNMP trap messages, and configure SNMP Alarms.

Add SNMP managers

You can configure a workstation running a management application that complies with SNMP version 1, 2, or 3 to access an appliance. Such a workstation is called an SNMP manager. If you do not specify an SNMP manager on the appliance, the appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds to SNMP queries from only those specific IP addresses. When specifying the IP address of an SNMP manager, you can use the netmask parameter to grant access from entire subnets. You can add a maximum of 100 SNMP managers or networks. To add an SNMP manager by using the CLI

At the command prompt, type the following commands to add an SNMP manager and verify the configuration:

```
add snmp manager <IPAddress> ... [-netmask <netmask>]
show snmp manager <IPAddress>
```

Example:

```
1 add snmp manager 10.102.29.5 -netmask 255.255.255.255
2 Done
3 show snmp manager 10.102.29.5
4 10.102.29.5 255.255.255.255
5 Done
6 <!--NeedCopy-->
```

To add an SNMP manager by using the GUI:

1. In the navigation pane, expand **System**, expand **SNMP**, and then click **Managers**.
2. In the details pane, click **Add**.
3. In the **Add SNMP Manager** dialog box, in the **IP Address** text box, type the IP address of the workstation running the management application (for example, 10.102.29.5).
4. Click **Create**, and then click **Close**.
5. Verify that the SNMP manager you added appears in the **Details** section at the bottom of the pane.

Add SNMP traps listeners

After configuring the alarms, you need to specify the trap listener to which the appliance will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap

listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

To add an SNMP trap listener by using the CLI

At the command prompt, type the following command to add an SNMP trap and verify that it has been added:

- `add snmp trap specific <IP>`
- `show snmp trap`

Example:

```
Trap type: SPECIFIC
Destination IP: 10.102.29.3
TD: 0
Destination Port: 162
Source IP: NetScaler IP
Version: V2
Min-Severity: -
AllPartition: DISABLED
Community: public
<!--NeedCopy-->
```

To add an SNMP trap listener by using the GUI

1. In the navigation pane, expand System, expand **SNMP**, and then click **Traps**.
2. In the details pane, click **Add**.
3. In the **Create SNMP Trap Destination** dialog box, in the **Destination IP Address** text box, type the IP address (for example, 10.102.29.3).
4. Click **Create** and then click **Close**.
5. Verify that the SNMP trap you added appears in the **Details** section at the bottom of the pane.

Configure SNMP alarms

You configure alarms so that the appliance generates a trap message when an event corresponding to one of the alarms occurs. Configuring an alarm consists of enabling the alarm and setting the severity level at which a trap is generated. There are five severity levels: Critical, Major, Minor, Warning, and Informational. A trap is sent only when the severity of the alarm matches the severity specified for the trap.

Some alarms are enabled by default. If you disable an SNMP alarm, the appliance will not generate trap messages when corresponding events occur. For example, if you disable the Login-Failure SNMP alarm, the appliance will not generate a trap message when a login failure occurs.

To enable or disable an alarm by using the CLI

At the command prompt, type the following commands to enable or disable an alarm and verify that it has been enabled or disabled:

```
set snmp alarm <trapName> [-state ENABLED      DISABLED ]
```

-
- show snmp alarm <trapName>

  Example

```
set snmp alarm LOGIN-FAILURE -state ENABLED
Done
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
\----- --------------- ---------------- ---- -------- ---------
   --------
LOGIN-FAILURE N/A N/A N/A ENABLED - ENABLED
Done
<!--NeedCopy-->
```

To set the severity of the alarm by using the CLI

At the command prompt, type the following commands to set the severity of the alarm and verify that the severity has been set correctly:

- `set snmp alarm <trapName> [-severity <severity>]`
- `show snmp alarm <trapName>`

Example:

```
set snmp alarm LOGIN-FAILURE -severity Major
Done
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
\----- --------------- ---------------- ---- -------- ---------
   --------
LOGIN-FAILURE N/A N/A N/A ENABLED Major ENABLED
```



```
7 Done
8 <!--NeedCopy-->
```

To configure alarms by using the GUI

1. In the navigation pane, expand **System**, expand SNMP, and then click **Alarms**.
2. In the details pane, select an alarm (for example, LOGIN-FAILURE), and then click **Open**.
3. In the **Configure SNMP Alarm** dialog box, to enable the alarm, select Enabled in the **State** drop-down list. To disable the alarm, select Disabled.
4. In the **Severity** drop-down list, select a severity option (for example, Major).
5. Click **OK**, and then click **Close**.
6. Verify that the parameters for the SNMP alarm you configured are correctly configured by viewing the **Details** section at the bottom of the pane.

Verify configuration

September 14, 2021

After you've finished configuring your system, complete the following checklists to verify your configuration.

Configuration checklist

- The build running is:
- There are no incompatibility issues. (Incompatibility issues are documented in the build's release notes.)
- The port settings (speed, duplex, flow control, monitoring) are the same as the switch's port.
- Enough SNIP IP addresses have been configured to support all server-side connections during peak times.
  - The number of configured SNIP IP addresses is:____
  - The expected number of simultaneous server connections is:
    [ ] 62,000 [ ] 124,000 [ ] Other____

Topology configuration checklist

The routes have been used to resolve servers on other subnets.

The routes entered are:

_____

- If the Citrix ADC appliance is in a public-private topology, reverse NAT has been configured.
- The failover (high availability) settings configured on the ADC appliance resolve in a one arm or two-arm configuration. All unused network interfaces have been disabled:

_____

- If the ADC appliance is placed behind an external load balancer, then the load balancing policy on the external load balancer is not “least connection”.

  The load balancing policy configured on the external load balancer is:

  _____

- If the ADC appliance is placed in front of a firewall, the session time-out on the firewall is set to a value greater than or equal to 300 seconds.

  Note: The TCP idle connection timeout on a Citrix ADC appliance is 360 seconds. If the timeout on the firewall is also set to 300 seconds or more, then the appliance can perform TCP connection multiplexing effectively because connections will not be closed earlier.

  The value configured for the session time-out is: _____

Server configuration checklist

- “Keep-alive” has been enabled on all the servers.

  The value configured for the keep-alive time-out is: _____

- The default gateway has been set to the correct value. (The default gateway should either be a Citrix ADC appliance or upstream router.) The default gateway is:

  _____

- The server port settings (speed, duplex, flow control, monitoring) are the same as the switch port settings.

  _____

- If the Microsoft® Internet Information Server is used, buffering is enabled on the server.
- If an Apache Server is used, the MaxConn (maximum number of connections) parameter is configured on the server and on the Citrix ADC appliance.

  The MaxConn (maximum number of connections) value that has been set is:

  _____

- If a Netscape Enterprise Server is used, the maximum requests per connection parameter is set on the Citrix ADC appliance. The maximum requests per connection value that has been set is:

  _____

Software features configuration checklist

- Does the Layer 2 mode feature need to be disabled? (Disable if another Layer 2 device is working in parallel with a Citrix ADC appliance.)

  Reason for enabling or disabling:

  _____

- Does the MAC-based forwarding feature need to be disabled? (If the MAC address used by return traffic is different, it should be disabled.)

  Reason for enabling or disabling:

  _____

- Does host-based reuse need to be disabled? (Is there virtual hosting on the servers?)

  Reason for enabling or disabling:

  _____

- Do the default settings of the surge protection feature need to be changed?

  Reason for changing or not changing:

  _____

Access checklist

- The system IPs can be pinged from the client-side network.
- The system IPs can be pinged from the server-side network.
- The managed server(s) can be pinged through the Citrix ADC.
- Internet hosts can be pinged from the managed servers.
- The managed server(s) can be accessed through the browser.
- The Internet can be accessed from managed server(s) using the browser.
- The system can be accessed using SSH.
- Admin access to all managed server(s) is working.

> Note: When you are using the ping utility, ensure that the pinged server has ICMP ECHO enabled, or your ping will not succeed.

Firewall checklist

The following firewall requirements have been met:

- UDP 161 (SNMP)
- UDP 162 (SNMP trap)

- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

Load balance traffic on a Citrix ADC appliance

September 14, 2021

The load balancing feature distributes client requests across multiple servers to optimize resource utilization. In a real-world scenario with a limited number of servers providing service to a large number of clients, a server can become overloaded and degrade the performance of the server farm. A Citrix ADC appliance uses load balancing criteria to prevent bottlenecks by forwarding each client request to the server best suited to handle the request when it arrives.

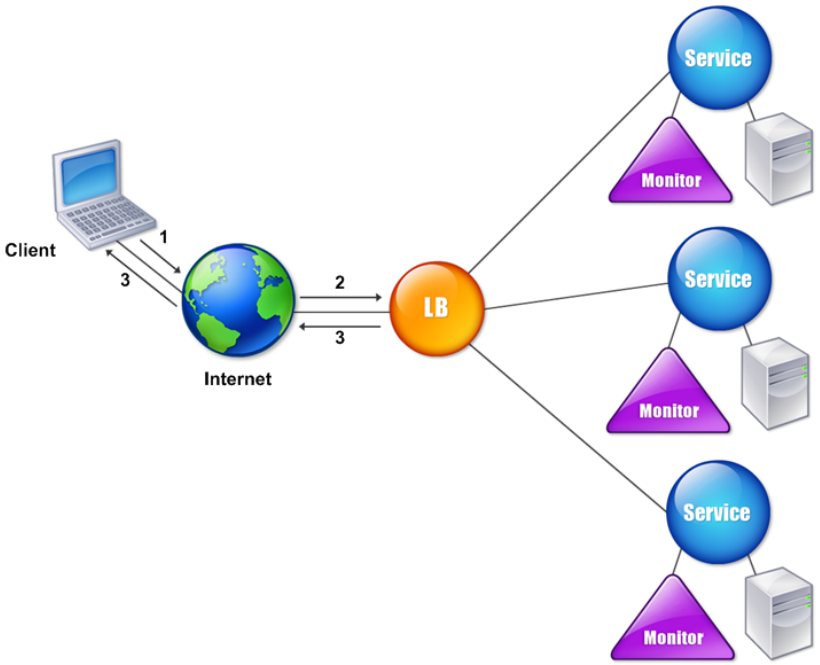
To configure load balancing, you define a virtual server to proxy multiple servers in a server farm and balance the load among them.

When a client initiates a connection to the server, a virtual server terminates the client connection and initiates a new connection with the selected server, or reuses an existing connection with the server, to perform load balancing. The load balancing feature provides traffic management from Layer 4 (TCP and UDP) through Layer 7 (FTP, HTTP, and HTTPS).

The Citrix ADC appliance uses a number of algorithms, called load balancing methods, to determine how to distribute the load among the servers. The default load balancing method is the Least Connections method.

A typical load balancing deployment consists of the entities described in the following figure.

Figure 1. Load Balancing Architecture



The entities function as follows:

- **Virtual server**. An entity that is represented by an IP address, a port, and a protocol. The virtual server IP address (VIP) is usually a public IP address. The client sends connection requests to this IP address. The virtual server represents a bank of servers.
- **Service**. A logical representation of a server or an application running on a server. Identifies the server's IP address, a port, and a protocol. The services are bound to the virtual servers.
- **Server object**. An entity that is represented by an IP address. The server object is created when you create a service. The IP address of the service is taken as the name of the server object. You can also create a server object and then create services by using the server object.
- **Monitor**. An entity that tracks the health of the services. The appliance periodically probes the servers using the monitor bound to each service. If a server does not respond within a specified response timeout, and the specified number of probes fails, the service is marked DOWN. The appliance then performs load balancing among the remaining services.

Load balancing

September 14, 2021

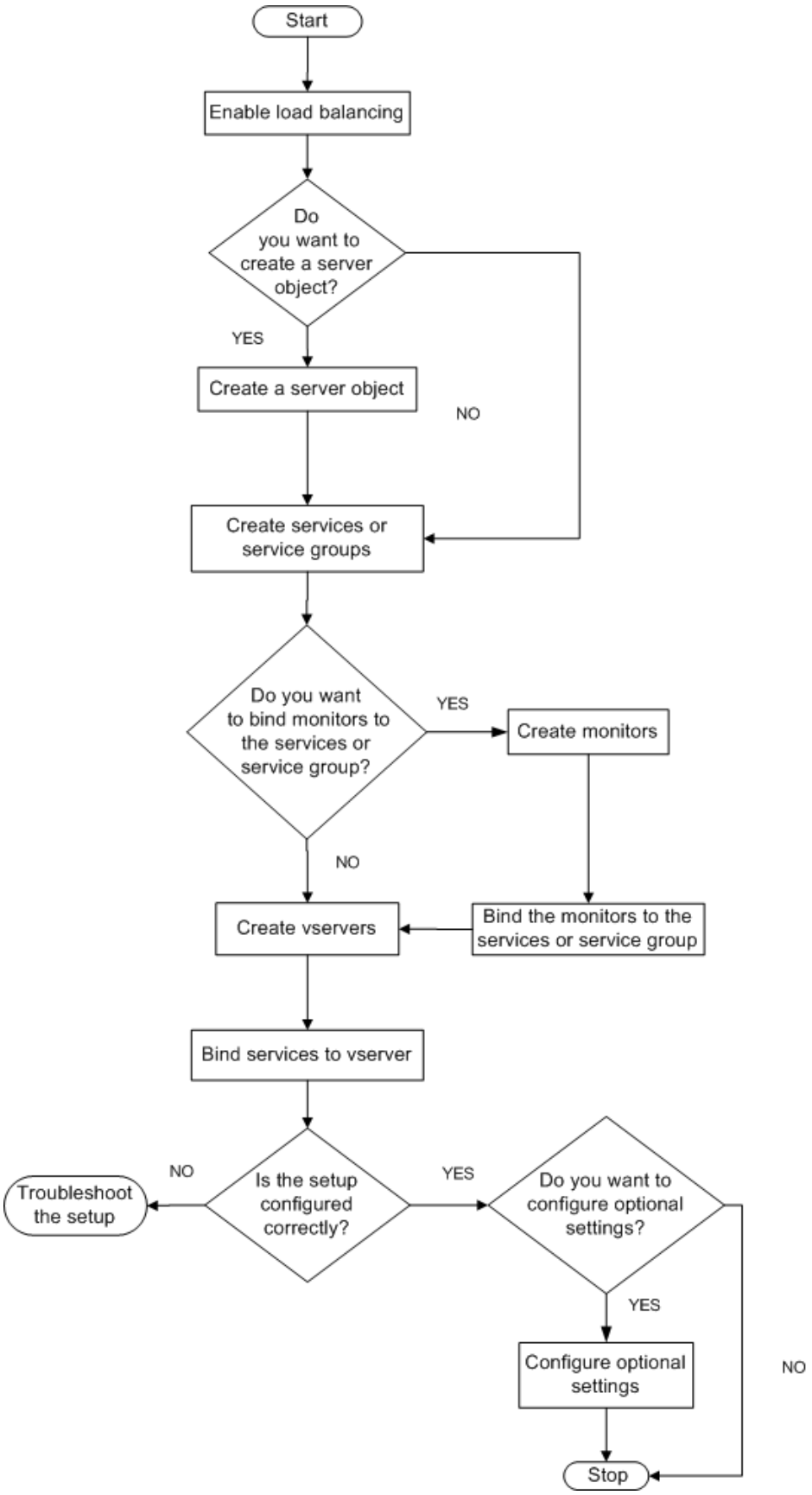
To configure load balancing, you must first create services. Then, you create virtual servers and bind the services to the virtual servers. By default, the Citrix ADC appliance binds a monitor to each service. After binding the services, verify your configuration by making sure that all of the settings are correct.

> Note: After you deploy the configuration, you can display statistics that show how the entities in the configuration are performing. Use the statistical utility or the stat lb vserver <vserverName> command.

Optionally, you can assign weights to a service. The load balancing method then uses the assigned weight to select a service. For getting started, however, you can limit optional tasks to configuring some basic persistence settings, for sessions that must maintain a connection to a particular server, and some basic configuration-protection settings.

The following flow chart illustrates the sequence of the configuration tasks.

Figure 1. Sequence of Tasks to Configure Load Balancing



Enable load balancing

Before configuring load balancing, make sure that the load balancing feature is enabled.

To enable load balancing by using the CLI

At the command prompt, type the following commands to enable load balancing and verify that it is enabled:

- enable feature lb
- show feature

Example

“‘ pre codeblock

> enable feature lb
> Done
> show feature

```
1   Feature                  Acronym          Status
2   -------                  -------          ------   1)  Web
       Logging               WL               OFF   2)  Surge
       Protection         SP               OFF   3)  Load Balancing
                 LB               ON   .   .   .   9)  SSL
       Offloading            SSL              ON   .   .   .  Done
       <!--NeedCopy--> ```
```

To enable load balancing by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click OK.
4. In the Enable/Disable Feature(s)? message, click Yes.

Configure services and a virtual server

When you have identified the services you want to load balance, you can implement your initial load balancing configuration by creating the service objects, creating a load balancing virtual server, and binding the service objects to the virtual server.

To implement the initial load balancing configuration by using the CLI

At the command prompt, type the following commands to implement and verify the initial configuration:

- <add service <name> <IPaddress> <serviceType> <port>
- <add lb vserver <vServerName> <serviceType> [<IPaddress> <port>]
- <bind lb vserver <name> <serviceName>
- <show service bindings <serviceName>

Example

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
 Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
 Done
> bind lb vserver vserver-LB-1 service-HTTP-1
 Done
> show service bindings service-HTTP-1
    service-HTTP-1 (10.102.29.5:80) - State : DOWN

      1)      vserver-LB-1 (10.102.29.60:80) - State : DOWN
 Done
<!--NeedCopy-->
```

To implement the initial load balancing configuration by using the GUI

1. Navigate to Traffic Management > Load Balancing.
2. In the details pane, under Getting Started, click Load Balancing wizard, and follow the instructions to create a basic load balancing setup.
3. Return to the navigation pane, expand Load Balancing, and then click Virtual Servers.
4. Select the virtual server that you configured and verify that the parameters displayed at the bottom of the page are correctly configured.
5. Click Open.
6. Verify that each service is bound to the virtual server by confirming that the Active check box is selected for each service on the Services tab.

Persistence settings

September 14, 2021

You must configure persistence on a virtual server if you want to maintain the states of connections on the servers represented by that virtual server (for example, connections used in e-commerce). The appliance then uses the configured load balancing method for the initial selection of a server, but forwards to that same server all subsequent requests from the same client.

If persistence is configured, it overrides the load balancing methods once the server has been selected. If the configured persistence applies to a service that is down, the appliance uses the load balancing methods to select a new service, and the new service becomes persistent for subsequent requests from the client. If the selected service is in an Out Of Service state, it continues to serve the outstanding requests but does not accept new requests or connections. After the shutdown period elapses, the existing connections are closed. The following table lists the types of persistence that you can configure.

Persistence Type	Persistent Connections
Source IP, SSL Session ID, Rule, DESTIP, SRCIPDESTIP	250K
CookieInsert, URL passive, Custom Server ID	Memory limit. In case of CookieInsert, if time out is not 0, any number of connections is allowed until limited by memory.

Table 1. Limitations on Number of Simultaneous Persistent Connections

If the configured persistence cannot be maintained because of a lack of resources on an appliance, the load balancing methods are used for server selection. Persistence is maintained for a configured period of time, depending on the persistence type. Some persistence types are specific to certain virtual servers. The following table shows the relationship.

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL Passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
SRCIPDESTIP	N/A	N/A	YES	YES	N/A
DESTIP	N/A	N/A	YES	YES	N/A

Table 2. Persistence Types Available for Each Type of Virtual Server

You can also specify persistence for a group of virtual servers. When you enable persistence on the group, the client requests are directed to the same selected server regardless of which virtual server in the group receives the client request. When the configured time for persistence elapses, any virtual server in the group can be selected for incoming client requests.

Two commonly used persistence types are persistence based on cookies and persistence based on server IDs in URLs.

Configure persistence based on cookies

When you enable persistence based on cookies, the the Citrix ADC appliance adds an HTTP cookie into the Set-Cookie header field of the HTTP response. The cookie contains information about the service to which the HTTP requests must be sent. The client stores the cookie and includes it in all subsequent requests, and the ADC uses it to select the service for those requests. You can use this type of persistence on virtual servers of type HTTP or HTTPS.

The Citrix ADC appliance inserts the cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

where:

- <<NSC_XXXX> is the virtual server ID that is derived from the virtual server name.
- <<ServiceIP> is the hexadecimal value of the IP address of the service.
- <<ServicePort> is the hexadecimal value of the port of the service.

The ADC encrypts ServiceIP and ServicePort when it inserts a cookie, and decrypts them when it receives a cookie.

Note: If the client is not allowed to store the HTTP cookie, the subsequent requests do not have the HTTP cookie, and persistence is not honored.

By default, the ADC appliance sends HTTP cookie version 0, in compliance with the Netscape specification. It can also send version 1, in compliance with RFC 2109.

You can configure a timeout value for persistence that is based on HTTP cookies. Note the following:

- If HTTP cookie version 0 is used, the Citrix ADC appliance inserts the absolute Coordinated Universal Time (GMT) of the cookie's expiration (the expires attribute of the HTTP cookie), calculated as the sum of the current GMT time on an ADC appliance, and the timeout value.
- If an HTTP cookie version 1 is used, the ADC appliance inserts a relative expiration time (Max-Age attribute of the HTTP cookie). In this case, the client software calculates the actual expiration time.

Note: Most client software currently installed (Microsoft Internet Explorer and Netscape browsers) understand HTTP cookie version 0; however, some HTTP proxies understand HTTP cookie version 1.

If you set the timeout value to 0, the ADC appliance does not specify the expiration time, regardless of the HTTP cookie version used. The expiration time then depends on the client software, and such cookies are not valid if that software is shut down. This persistence type does not consume any system resources. Therefore, it can accommodate an unlimited number of persistent clients.

An administrator can change the HTTP cookie version.

To change the HTTP cookie version by using the CLI

At the command prompt, type;

```
set ns param [-cookieversion ( 0 | 1 )]
<!--NeedCopy-->
```

Example:

```
set ns param -cookieversion 1
<!--NeedCopy-->
```

To change the HTTP cookie version by using the GUI

1. Navigate to System > Settings.
2. In the details pane, click Change HTTP Parameters.
3. In the Configure HTTP Parameters dialog box, under Cookie, select Version 0 or Version 1.

Note: For information about the parameters, see Configure persistence based on cookies.

To configure persistence based on cookies by using the CLI

At the command prompt, type the following commands to configure persistence based on cookies and verify the configuration:

```
set lb vserver <name> -persistenceType COOKIEINSERT

```



```
show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
Done
show lb vserver vserver-LB-1
    vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
    .
    .
    .
    Persistence: COOKIEINSERT (version 0)
    Persistence Timeout: 2 min
    .
    .
    .
 Done
<!--NeedCopy-->
```

To configure persistence based on cookies by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Method and Persistence tab, in the Persistence list, select COOKIEINSERT.
4. In the Time-out (min) text box, type the time-out value (for example, 2).
5. Click OK.
6. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane.

Configure persistence based on server IDs in URLs

The Citrix ADC appliance can maintain persistence based on the server IDs in the URLs. In a technique called URL passive persistence, the ADC extracts the server ID from the server response and embeds it in the URL query of the client request. The server ID is an IP address and port specified as a hexadecimal number. The ADC extracts the server ID from subsequent client requests and uses it to select the server.

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. For more information about

expressions, see Policy Configuration and Reference.

Note: If the server ID cannot be extracted from the client requests, server selection is based on the load balancing method.

Example: Payload Expression

The expression, URLQUERY contains sid= configures the system to extract the server ID from the URL query of a client request, after matching token sid=. Thus, a request with the URL `http://www.citrix.com/index.asp?\\&sid;=c0a864100050` is directed to the server with the IP address10.102.29.10 and port 80.

The timeout value does not affect this type of persistence, which is maintained as long as the server ID can be extracted from the client requests. This persistence type does not consume any system resources, so it can accommodate an unlimited number of persistent clients.

> Note: For information about the parameters, see Load Balancing.

To configure persistence based on server IDs in URLs by using the CLI

At the command prompt, type the following commands to configure persistence based on server IDs in URLs and verify the configuration:

```
set lb vserver <name> -persistenceType URLPASSIVE

<show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
Done
show lb vserver vserver-LB-1
    vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
    .
    .
    .
    Persistence: URLPASSIVE
    Persistence Timeout: 2 min
    .
    .
    .
Done
<!--NeedCopy-->
```


To configure persistence based on server IDs in URLs by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Method and Persistence tab, in the Persistence list, select URLPASSIVE.
4. In the Time-out (min) text box, type the time-out value (for example, 2).
5. In the Rule text box, enter a valid expression. Alternatively, click Configure next to the Rule text box and use the Create Expression dialog box to create an expression.
6. Click OK.
7. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane.

Configure features to protect the load balancing configuration

September 14, 2021

You can configure URL redirection to provide notifications of virtual server malfunctions, and you can configure backup virtual servers to take over if a primary virtual server becomes unavailable.

Configure URL redirection

You can configure a redirect URL to communicate the status of the appliance in the event that a virtual server of type HTTP or HTTPS is down or disabled. This URL can be a local or remote link. The appliance uses HTTP 302 redirect.

Redirects can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. In this case, a redirect is used when both the primary and backup virtual servers are down.

To configure a virtual server to redirect client requests to a URL by using the CLI

At the command prompt, type the following commands to configure a virtual server to redirect client requests to a URL and verify the configuration:


```
set lb vserver <name> -redirectURL <URL>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
    > set lb vserver vserver-LB-1 -redirectURL <http://www.newdomain.
       com/mysite/maintenance>
     Done
    > show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
        .
        .
        .
        Redirect URL: <http://www.newdomain.com/mysite/maintenance>
        .
        .
        .
     Done
    >
<!--NeedCopy-->
```

To configure a virtual server to redirect client requests to a URL by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure URL redirection (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Redirect URL text box, type the URL (for example, `http://www.newdomain.com/mysite/maintenance`), and then click OK.
4. Verify that the redirect URL you configured for the server appears in the Details section at the bottom of the pane.

Configure backup virtual servers

If the primary virtual server is down or disabled, the appliance can direct the connections or client requests to a backup virtual server that forwards the client traffic to the services. The appliance can also send a notification message to the client regarding the site outage or maintenance. The backup virtual server is a proxy and is transparent to the client.

You can configure a backup virtual server when you create a virtual server or when you change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating a cascaded backup virtual server. The maximum depth of cascading backup virtual servers is 10. The appliance searches for a backup virtual server that is up and accesses that virtual server to deliver the content.

You can configure URL redirection on the primary for use when the primary and the backup virtual servers are down or have reached their thresholds for handling requests.

Note: If no backup virtual server exists, an error message appears, unless the virtual server is configured with a redirect URL. If both a backup virtual server and a redirect URL are configured, the backup virtual server takes precedence.

To configure a backup virtual server by using the CLI

At the command prompt, type the following commands to configure a backup server and verify the configuration:

```
set lb vserver <name> [-backupVserver <string>]

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
    > set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
     Done
    > show lb vserver vserver-LB-1
        vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
        .
        .
        .
        Backup: vserver-LB-2
        .
        .
        .
     Done
    >
<!--NeedCopy-->
```


To set up a backup virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the backup virtual server (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Backup Virtual Server list, select the backup virtual server (for example, vserver-LB-2, and then click OK.
4. Verify that the backup virtual server you configured appears in the Details section at the bottom of the pane.

   Note: If the primary server goes down and then comes back up, and you want the backup virtual server to function as the primary server until you explicitly reestablish the primary virtual server, select the Disable Primary When Down check box.

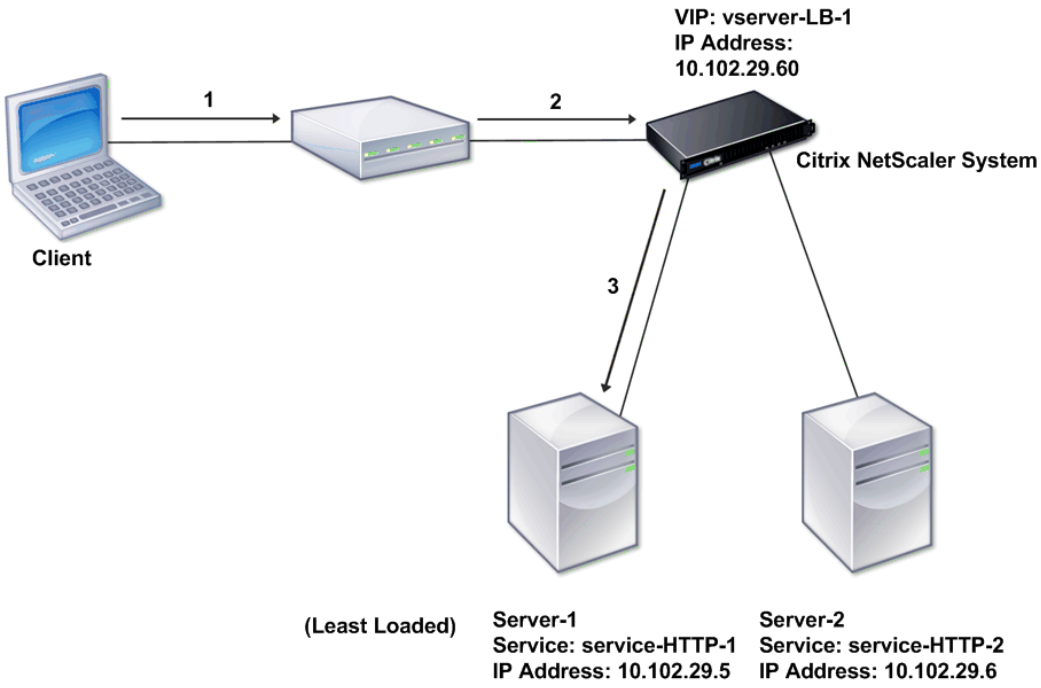
A typical load balancing scenario

September 14, 2021

In a load balancing setup, the Citrix ADC appliances are logically located between the client and the server farm, and they manage traffic flow to the servers.

The following figure shows the topology of a basic load balancing configuration.

Figure 1. Basic Load Balancing Topology

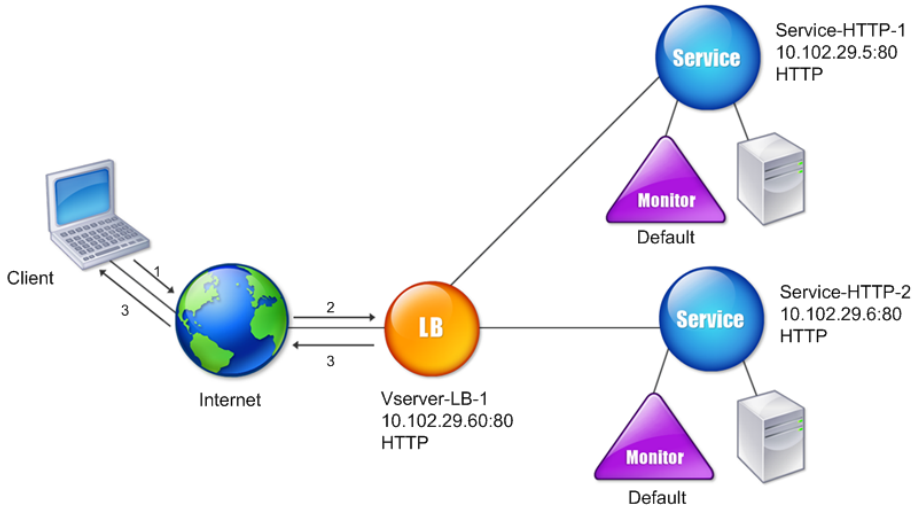


The virtual server selects the service and assigns it to serve client requests. Consider the scenario in the preceding figure, where the services service-HTTP-1 and service-HTTP-2 are created and bound to the virtual server named virtual server-LB-1. Virtual server-LB-1 forwards the client request to either service-HTTP-1 or service-HTTP-2. The system selects the service for each request by using the Least Connections load balancing method. The following table lists the names and values of the basic entities that must be configured on the system.

Table 1. LB Configuration Parameter Values

The following figure shows the load balancing sample values and required parameters that are described in the preceding table.

Figure 2. Load Balancing Entity Model



The following tables list the commands used to configure this load balancing setup by using the command line interface.

Task	Command
To enable load balancing	enable feature lb
To create a service named service-HTTP-1	add service service-HTTP-1 10.102.29.5 HTTP 80
To create a service named service-HTTP-2	add service service-HTTP-2 10.102.29.6 HTTP 80
To create a virtual server named vserver-LB-1	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
To bind a service named service-HTTP-1 to a virtual server named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-1
To bind a service named service-HTTP-2 to a virtual server named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-2

Table 2. Initial Configuration Tasks

For more information about the initial configuration tasks, see Setting Up Basic Load Balancing.

Task	Command
To view the properties of a virtual server named vserver-LB-1	show lb vserver vserver-LB-1
To view the statistics of a virtual server named vserver-LB-1	stat lb vserver vserver-LB-1
To view the properties of a service named service-HTTP-1	show service service-HTTP-1
To view the statistics of a service named service-HTTP-1	stat service service-HTTP-1
To view the bindings of a service named service-HTTP-1	show service bindings service-HTTP-1

Table 3. Verification Tasks

Task	Command
To configure persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
To configure COOKIEINSERT persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
To configure URLPassive persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
To configure a virtual server to redirect the client request to a URL on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
To set a backup virtual server on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

Table 4. Customization Tasks

For more information about configuring persistence, see Choosing and Configuring Persistence Settings. For information about configuring a virtual server to redirect a client request to a URL and setting up a backup virtual server, see Configuring Features to Protect the Load Balancing Configuration.

Use case: How to force Secure and HttpOnly cookie options for websites using the Citrix ADC appliance

September 14, 2021

The web administrators may force the Secure, or HttpOnly, or both the flags on the Session ID and the authentication cookies that are generated by the web applications. You can modify the Set-cookie headers to include these two options by using an HTTP load balancing virtual server and rewrite policies on a Citrix ADC appliance.

- **HttpOnly** - This option on a cookie causes the web browsers to return the cookie using the HTTP or HTTPS protocol only. The non-HTTP methods such as JavaScript document.cookie references cannot access the cookie. This option helps in preventing cookie theft due to cross-site scripting.

  > NOTE
  >
  > You cannot use the HttpOnly option when a web application requires access to Cookie contents by using a client-side script, such as JavaScript or a client-side Java Applet. You can use the method mentioned in this document to rewrite only the server-generated cookies and not the cookies generated by Citrix ADC appliance. For example, AppFirewall, persistence, VPN session cookies and so on.

- **Secure** - This option on a cookie causes the web browsers to return only the cookie value when the transmission is encrypted by SSL. This option can be used to prevent cookie theft through connection eavesdropping.

> NOTE
>
> The following procedure is not applicable for VPN virtual servers.

To configure the Citrix ADC appliance to force the Secure and HttpOnly flags for an existing HTTP virtual server by using CLI

1. Create a rewrite action.

   This example is configured to set both Secure and HttpOnly flags. If either one is missing, modify it as necessary for other combinations.

```
1 add rewrite action act_cookie_Secure replace_all http.RES.
      full_Header "\"Secure; HttpOnly; path=/\"" -search "regex(re!(
      path=/\\; Secure; HttpOnly)|(path=/\\; Secure)|(path=/\\;
      HttpOnly)|(path=/)!)" -bypassSafetyCheck YES
2 <!--NeedCopy-->
```


This policy replaces all instances of “path=/”, “path=/; Secure”, “path=/; Secure; HttpOnly” and “path=/; HttpOnly” with “Secure; HttpOnly; path=/”. This regular expression (regex) fails if the case doesn’t match.

2. Create a rewrite policy to trigger the action.

```
add rewrite policy rw_force_secure_cookie "http.RES.HEADER(\"Set-Cookie\").EXISTS" act_cookie_Secure
<!--NeedCopy-->
```

3. Bind the rewrite policy to the virtual server to be secured. If `Secure` option is used, an SSL virtual server must be used.

```
bind lb vserver mySSLVServer -policyName rw_force_secure_cookie -priority 100 -gotoPriorityExpression NEXT -type RESPONSE
<!--NeedCopy-->
```

Examples:

The following example shows the cookie before setting the httpOnly flag

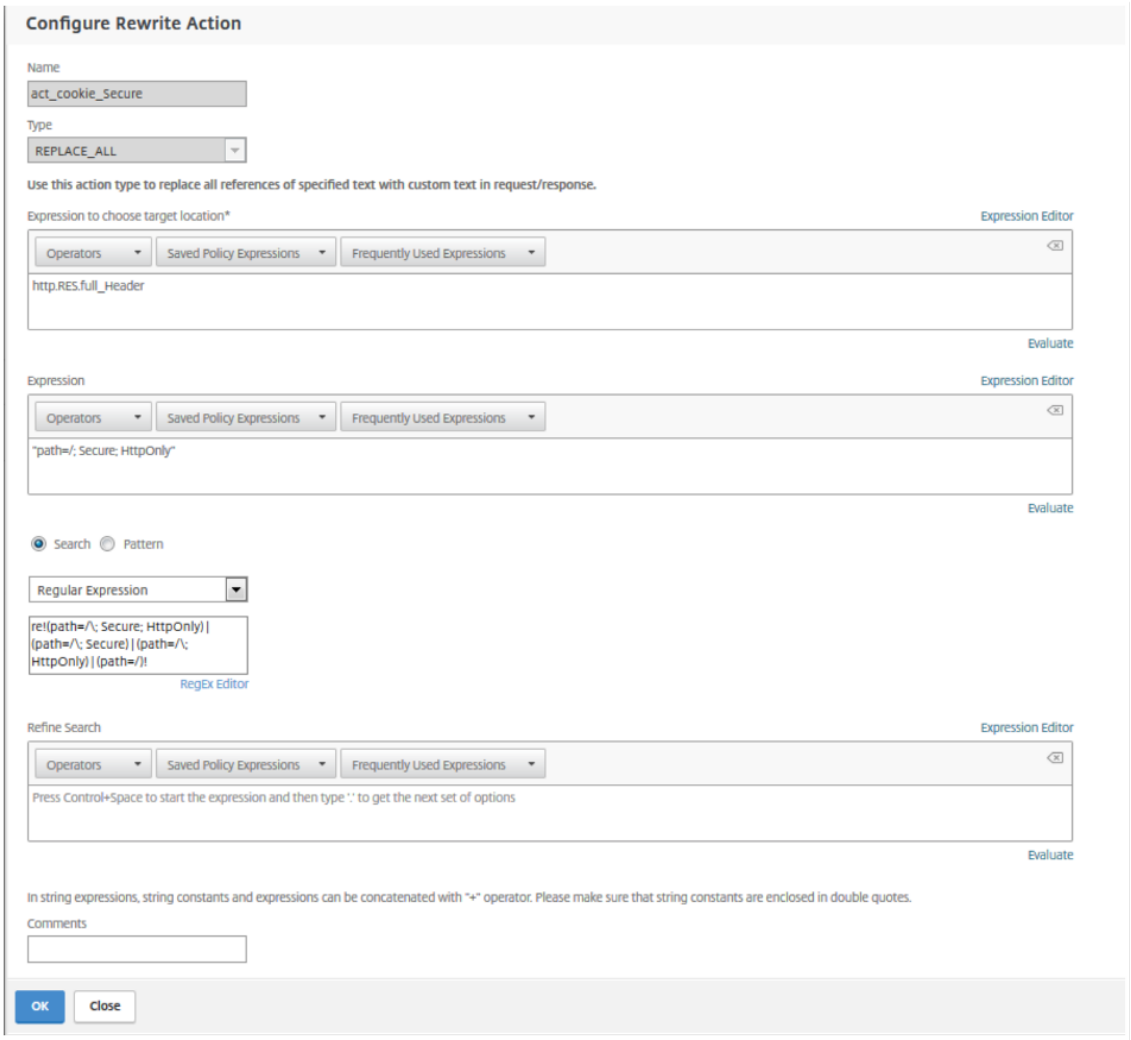
```
Set-Cookie: CtxsAuthId=C5614491; path=/Citrix/ProdWeb
<!--NeedCopy-->
```

The following example shows the cookie after setting the httpOnly flag

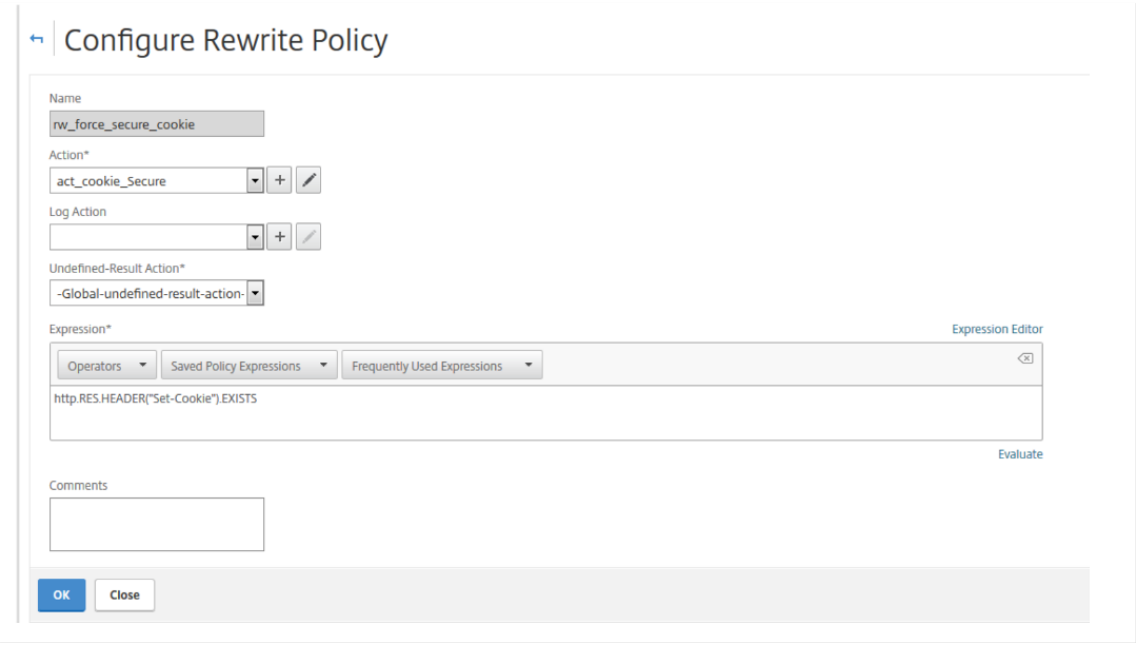
```
Set-Cookie: CtxsAuthId=C5614491; Secure; HttpOnly; path=/Citrix/ProdWeb/
<!--NeedCopy-->
```

To configure the Citrix ADC appliance to force the Secure and HttpOnly flags for an existing HTTP virtual server by using GUI

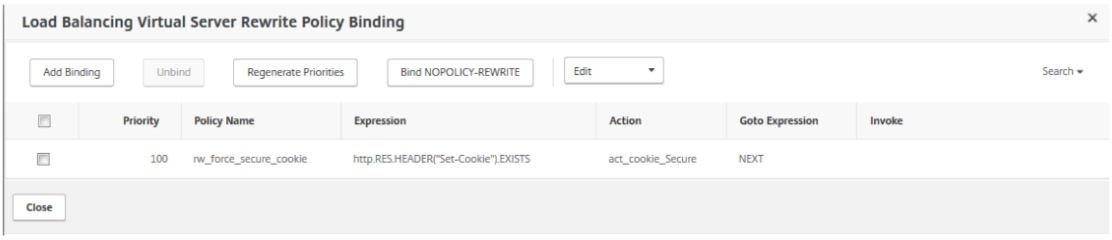
1. Navigate to **AppExpert > Rewrite > Actions**, and click **Add** to add a new rewrite action.



2. Navigate to **AppExpert > Rewrite > Policies**, and click **Add** to add a new rewrite policy.



3. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and then bind the rewrite (response) policy to the corresponding SSL virtual server.



Accelerate load balanced traffic by using compression

September 14, 2021

Compression is a popular means of optimizing bandwidth usage, and most web browsers support compressed data. If you enable the compression feature, the Citrix ADC appliance intercepts requests from clients and determines whether the client can accept compressed content. After receiving the HTTP response from the server, the appliance examines the content to determine whether it is compressible. If the content is compressible, the appliance compresses it, modifies the response header to indicate the type of compression performed, and forwards the compressed content to the client.

Citrix ADC compression is a policy-based feature. A policy filters requests and responses to identify responses to be compressed, and specifies the type of compression to apply to each response. The appliance provides several built-in policies to compress common MIME types such as text/html, text/plain,

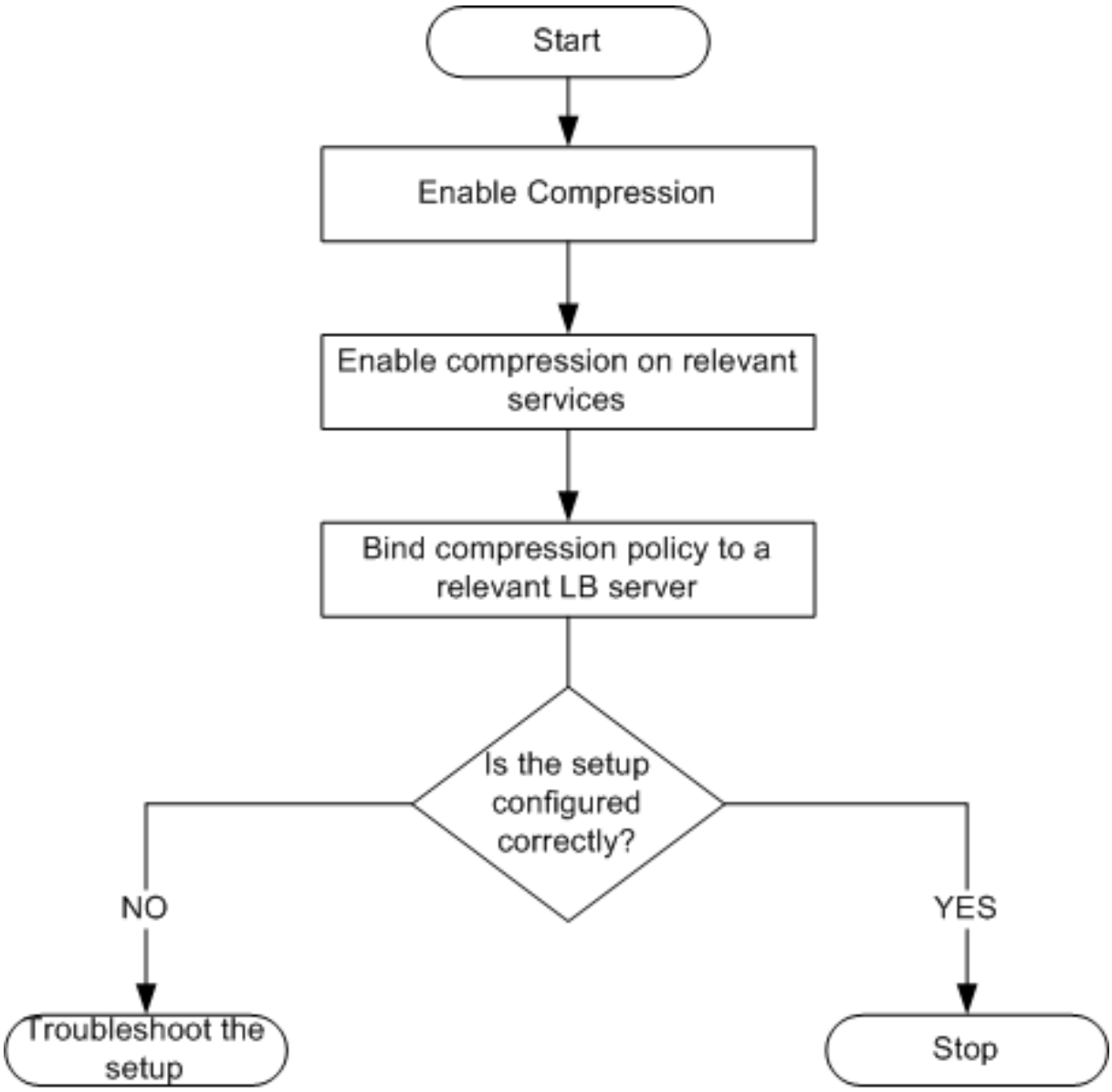
text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint. You can also create custom policies. The appliance does not compress compressed MIME types such as application/octet-stream, binary, bytes, and compressed image formats such as GIF and JPEG.

To configure compression, you must enable it globally and on each service that will provide responses that you want compressed. If you have configured virtual servers for load balancing or content switching, you should bind the polices to the virtual servers. Otherwise, the policies apply to all traffic that passes through the appliance.

Compression configuration task sequence

The following flow chart shows the sequence of tasks for configuring basic compression in a load balancing setup.

Figure 1. Sequence of Tasks to Configure Compression



Note: The steps in the above figure assume that load balancing has already been configured.

Enable compression

By default, compression is not enabled. You must enable the compression feature to allow compression of HTTP responses that are sent to the client.

To enable compression by using the CLI

At the command prompt, type the following commands to enable compression and verify the configuration:

- enable ns feature CMP
- show ns feature

```
    > enable ns feature CMP


    Done

    > show ns feature


           Feature                        Acronym              Status

           -------                        -------              ------

    1)     Web Logging                    WL                   ON

    2)     Surge Protection               SP                   OFF

    .

    7) Compression Control CMP ON

    8)     Priority Queuing               PQ                   OFF

    .

    Done
<!--NeedCopy-->
```


To enable compression by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Compression check box, and then click OK.
4. In the Enable/Disable Feature(s)? dialog box, click Yes.

Configure services to compress data

In addition to enabling compression globally, you must enable it on each service that will deliver files to be compressed.

To enable compression on a service by using the CLI

At the command prompt, type the following commands to enable compression on a service and verify the configuration:

- set service <name> -CMP YES
- show service <name>

```
> show service SVC_HTTP1

SVC_HTTP1 (10.102.29.18:80) - HTTP

State: UP

Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)

Time since last state change: 0 days, 03:03:37.200

Server Name: 10.102.29.18

Server ID : 0   Monitor Threshold : 0

Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits

```



```
24
25 Use Source IP: NO
26
27
28 Client Keepalive(CKA): NO
29
30
31 Access Down Service: NO
32
33
34 TCP Buffering(TCPB): NO
35
36
37 HTTP Compression(CMP): YES
38
39
40 Idle timeout: Client: 180 sec   Server: 360 sec
41
42
43 Client IP: DISABLED
44
45
46 Cacheable: NO
47
48
49 SC: OFF
50
51
52 SP: OFF
53
54
55 Down state flush: ENABLED
56
57 1)      Monitor Name: tcp-default
58
59
60 State: DOWN     Weight: 1
61
62
63 Probes: 1095    Failed [Total: 1095 Current: 1095]
64
65
66 Last response: Failure - TCP syn sent, reset received.
67
68
```



```
Response Time: N/A


Done

<!--NeedCopy-->
```

To enable compression on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select the service for which you want to configure compression (for example, service-HTTP-1), and then click Open.
3. On the Advanced tab, under Settings, select the Compression check box, and then click OK.
4. Verify that, when the service is selected, HTTP Compression(CMP): ON appears in the **Details** section at the bottom of the pane.

Bind a compression policy to a virtual server

If you bind a policy to a virtual server, the policy is evaluated only by the services associated with that virtual server. You can bind compression policies to a virtual server either from the Configure Virtual Server (Load Balancing) dialog box or from the Compression Policy Manager dialog box. This topic includes instructions to bind compression policies to a load balancing virtual server by using the Configure Virtual Server (Load Balancing) dialog box.

To bind or unbind a compression policy to a virtual server by using the command line

At the command prompt, type the following commands to bind or unbind a compression policy to a load balancing virtual server and verify the configuration:

(bind	unbind) lb vserver <name> -policyName <string>

-
- show lb vserver <name>

Example:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> showlbvserverlbvip
```



```
4
5  lbvip(8.7.6.6:80)-HTTPType:ADDRESS
6  State:UP
7  LaststatechangewasatThuMay2805:37:212009(+685ms)
8  Timesincelaststatechange:19days,04:26:50.470
9  EffectiveState:UP
10 ClientIdleTimeout:180sec
11 Downstateflush:ENABLED
12 DisablePrimaryVserverOnDown:DISABLED
13 PortRewrite:DISABLED
14 No.ofBoundServices:1(Total)1(Active)
15 ConfiguredMethod:LEASTCONNECTION
16 CurrentMethod:RoundRobin,Reason:Boundservice'sstatechangedtoUP
17 Mode:IP
18 Persistence:NONE
19 VserverIPandPortinsertion:OFF
20 Push:DISABLEDPushVServer:
21 PushMultiClients:NO
22 PushLabelRule:
23
24 BoundServiceGroups:
25 1)GroupName:Service-Group-1
26
27 1)Service-Group-1(10.102.29.252:80)-HTTPState:UPWeight:1
28
29 1)Policy:ns_cmp_msappPriority:0
30
31 Done
32
33 <!--NeedCopy-->
```

To bind or unbind a compression policy to a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server to which you want to bind or unbind a compression policy (for example, Vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Policies tab, click Compression.
4. Do one of the following:
   - To bind a compression policy, click Insert Policy, and then select the policy you want to bind to the virtual server.
   - To unbind a compression policy, click the name of the policy you want to unbind from the virtual server, and then click Unbind Policy.

5. Click OK.

Secure load balanced traffic by using SSL

September 14, 2021

The Citrix ADC SSL offload feature transparently improves the performance of websites that conduct SSL transactions. By offloading CPU-intensive SSL encryption and decryption tasks from the local web server to the appliance, SSL offloading ensures secure delivery of web applications without the performance penalty incurred when the server processes the SSL data. Once the SSL traffic is decrypted, it can be processed by all standard services. The SSL protocol works seamlessly with various types of HTTP and TCP data and provides a secure channel for transactions using such data.

To configure SSL, you must first enable it. Then, you configure HTTP or TCP services and an SSL virtual server on the appliance, and bind the services to the virtual server. You must also add a certificate-key pair and bind it to the SSL virtual server. If you use Outlook Web Access servers, you must create an action to enable SSL support and a policy to apply the action. An SSL virtual server intercepts incoming encrypted traffic and decrypts it by using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the appliance for appropriate processing.

For detailed information about SSL offloading, see SSL offload and acceleration.

SSL configuration task sequence

To configure SSL, you must first enable it. Then, you must create an SSL virtual server and HTTP or TCP services on the Citrix ADC appliance. Finally, you must bind a valid SSL certificate and the configured services to the SSL virtual server.

An SSL virtual server intercepts incoming encrypted traffic and decrypts it using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the Citrix ADC appliance for appropriate processing.

The following flow chart shows the sequence of tasks for configuring a basic SSL offload setup.

Figure 1. Sequence of Tasks to Configure SSL Offloading

Enable SSL offload

First enable the SSL feature. You can configure SSL-based entities on the appliance without enabling the SSL feature, but they will not work until you enable SSL.

Enable SSL by using the CLI

At the command prompt, type the following commands to enable SSL Offload and verify the configuration:

```
-  enable ns feature SSL
-  show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature ssl

Done


> show ns feature


Feature Acronym Status


------- ------- ------


1) Web Logging WL ON


2) SurgeProtection SP OFF


3) Load Balancing LB ON . . .


 9) SSL Offloading SSL ON


10) Global Server Load Balancing GSLB ON . .


```



```
30 Done >
31 <!--NeedCopy-->
```

Enable SSL by using the GUI

Follow these steps:

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes and Features**, click **Change basic features**.
3. Select the **SSL Offloading** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** message box, click **Yes**.

Create HTTP services

A service on the appliance represents an application on a server. Once configured, services are in the disabled state until the appliance can reach the server on the network and monitor its status. This topic covers the steps to create an HTTP service.

Note: For TCP traffic, perform the following procedures, but create TCP services instead of HTTP services.

Add an HTTP service by using the CLI

At the command prompt, type the following commands to add an HTTP service and verify the configuration:

```
1 -  add service <name> (<IP> | <serverName>) <serviceType> <port>
2 -  show service <name>
3 <!--NeedCopy-->
```

Example:

```
1 > add service SVC_HTTP1 10.102.29.18 HTTP 80
2
3
4  Done
5
6
7 > show service SVC_HTTP1
8
9
10         SVC_HTTP1 (10.102.29.18:80) - HTTP
11
```



```
12
13          State: UP
14
15
16          Last state change was at Wed Jul 15 06:13:05 2009
17
18
19          Time since last state change: 0 days, 00:00:15.350
20
21
22          Server Name: 10.102.29.18
23
24
25          Server ID : 0   Monitor Threshold : 0
26
27
28          Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
29
30
31          Use Source IP: NO
32
33
34          Client Keepalive(CKA): NO
35
36
37          Access Down Service: NO
38
39
40          TCP Buffering(TCPB): NO
41
42
43          HTTP Compression(CMP): YES
44
45
46          Idle timeout: Client: 180 sec   Server: 360 sec
47
48
49          Client IP: DISABLED
50
51
52          Cacheable: NO
53
54
55          SC: OFF
56
```



```
57
58          SP: OFF
59
60
61          Down state flush: ENABLED
62
63
64
65
66
67  1)      Monitor Name: tcp-default
68
69
70                  State: UP       Weight: 1
71
72
73                  Probes: 4       Failed [Total: 0 Current: 0]
74
75
76                  Last response: Success - TCP syn+ack received.
77
78
79                  Response Time: N/A
80
81
82   Done
83  <!--NeedCopy-->
```

Add an HTTP service by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL Offload > Services**.
2. In the details pane, click **Add**.
3. In the **Create Service** dialog box, type the name of the service, IP address, and port (for example, SVC_HTTP1, 10.102.29.18, and 80).
4. In the **Protocol** list, select the type of the service (for example, HTTP).
5. Click **Create**, and then click **Close**. The HTTP service you configured appears in the Services page.
6. Verify that the parameters you configured are correctly configured by selecting the service and viewing the Details section at the bottom of the pane.

Add an SSL based virtual server

In a basic SSL offloading setup, the SSL virtual server intercepts encrypted traffic, decrypts it, and sends the clear text messages to the services that are bound to the virtual server. Offloading CPU-intensive SSL processing to the appliance allows the back-end servers to process a greater number of requests.

Add an SSL-based virtual server by using the CLI

At the command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

```
-  add lb vserver <name> <serviceType> [<IPAddress> <port>]
-  show lb vserver <name>
<!--NeedCopy-->
```

> **Caution:** To ensure secure connections, you must bind a valid SSL certificate to the SSL-based virtual server before you enable it.

Example:

```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
  Done


  > show lb vserver vserver-SSL-1


  vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS


  State: DOWN[Certkey not bound] Last state change was at Tue Jun 16
     06:33:08 2009 (+176 ms)


  Time since last state change: 0 days, 00:03:44.120


  Effective State: DOWN Client Idle Timeout: 180 sec


  Down state flush: ENABLED


  Disable Primary Vserver On Down : DISABLED
```



```
24
25
26    No. of Bound Services : 0 (Total) 0 (Active)
27
28
29    Configured Method: LEASTCONNECTION Mode: IP
30
31
32    Persistence: NONE
33
34
35    Vserver IP and Port insertion: OFF
36
37
38    Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:
          Done
39 <!--NeedCopy-->
```

Add an SSL-based virtual server by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL Offload > Virtual Servers**.
2. In the details pane, click **Add**.
3. In the **Create Virtual Server (SSL Offload)** dialog box, type the name of the virtual server, IP address, and port.
4. In the **Protocol** list, select the type of the virtual server, for example, SSL.
5. Click **Create**, and then click **Close**.
6. Verify that the parameters you configured are correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane. The virtual server is marked as DOWN because a certificate-key pair and services have not been bound to it.

> **Caution:** To ensure secure connections, you must bind a valid SSL certificate to the SSL-based virtual server before you enable it.

Bind services to the SSL virtual server

After decrypting the incoming data, the SSL virtual server forwards the data to the services that you have bound to the virtual server.

Data transfer between the appliance and the servers can be encrypted or in clear text. If the data transfer between the appliance and the servers is encrypted, the entire transaction is secure from end

to end. For more information about configuring the system for end-to-end security, see SSL offload and acceleration.

Bind a service to a virtual server by using the CLI

At the command prompt, type the following commands to bind a service to the SSL virtual server and verify the configuration:

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1



  Done


  > show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) -
     SSL Type:

  ADDRESS State: DOWN[Certkey not bound]

  Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)

  Time since last state change: 0 days, 00:31:53.70

  Effective State: DOWN Client Idle

  Timeout: 180 sec

  Down state flush: ENABLED Disable Primary Vserver On Down :

  DISABLED No. of Bound Services : 1 (Total) 0 (Active)
```



```
31
32
33    Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver
          IP and
34
35
36    Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients:
         NO Push Label Rule:
37
38
39
40
41
42    1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
43
44
45    State: DOWN Weight: 1
46
47
48    Done
49  <!--NeedCopy-->
```

Bind a service to a virtual server by using the GUI

1. Navigate to **Traffic Management > SSL Offload > Virtual Servers**.
2. In the details pane, select a virtual server, and then click **Open**.
3. On the **Services** tab, in the **Active** column, select the check boxes next to the services that you want to bind to the selected virtual server.
4. Click **OK**.
5. Verify that the Number of Bound Services counter in the Details section at the bottom of the pane is incremented by the number of services that you bound to the virtual server.

Add a certificate-key pair

An SSL certificate is an integral element of the SSL Key-Exchange and encryption/decryption process. The certificate is used during an SSL handshake to establish the identity of the SSL server. You can use a valid, existing SSL certificate that you have on the Citrix ADC appliance, or you can create your own SSL certificate. The appliance supports RSA certificates of up to 4096 bits.

ECDSA certificates with only the following curves are supported:

- prime256v1 (P_256 on the ADC)
- secp384r1 (P_384 on the ADC)

- secp521r1 (P_521 on the ADC; supported on VPX only)
- secp224r1 (P_224 on the ADC; supported on VPX only)

> **Note:** Citrix recommends that you use a valid SSL certificate that has been issued by a trusted certificate authority. Invalid certificates and self-created certificates are not compatible with all SSL clients.

Before a certificate can be used for SSL processing, you must pair it with its corresponding key. The certificate key pair is then bound to the virtual server and used for SSL processing.

Add a certificate key pair by using the CLI

> **Note:** For information about creating an ECDSA certificate-key pair, see Create an ECDSA certificate-key pair.

At the command prompt, type the following commands to create a certificate key pair and verify the configuration:

```
1 -  add ssl certKey <certkeyName> -cert <string> [-key <string>]
2 -  show sslcertkey <name>
3 <!--NeedCopy-->
```

Example:

```
1  > add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
2
3   Done
4
5
6  > show sslcertkey CertKey-SSL-1
7
8
9     Name: CertKey-SSL-1 Status: Valid,
10
11
12    Days to expiration:4811 Version: 3
13
14
15    Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer:
          C=US,ST=California,L=San
16
17
18    Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
19
20
```



```
21     Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17
          21:26:47 2022 GMT
22
23
24     Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,
          CN=d efault Public Key
25
26
27     Algorithm: rsaEncryption Public Key
28
29
30     size: 1024
31
32
33   Done
34 <!--NeedCopy-->
```

Add a certificate key pair by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL > Certificates**.
2. In the details pane, click **Add**.
3. In the **Install Certificate** dialog box, in the Certificate-Key Pair Name text box, type a name for the certificate key pair you want to add, for example, Certkey-SSL-1.
4. Under **Details**, in Certificate File Name, click **Browse (Appliance)** to locate the certificate. Both the certificate and the key are stored in the /nsconfig/ssl/ folder on the appliance. To use a certificate present on the local system, select Local.
5. Select the certificate you want to use, and then click **Select**.
6. In Private Key File Name, click **Browse (Appliance)** to locate the private key file. To use a private key present on the local system, select Local.
7. Select the key you want to use and click **Select**. To encrypt the key used in the certificate key pair, type the password to be used for encryption in the Password text box.
8. Click **Install**.
9. Double-click the certificate key pair and, in the Certificate Details window, verify that the parameters have been configured correctly and saved.

Bind an SSL certificate key pair to the virtual server

After you pairing an SSL certificate with its corresponding key, bind the certificate-key pair to the SSL virtual server so that it can be used for SSL processing. Secure sessions require establishing a connection between the client computer and an SSL-based virtual server on the appliance. SSL processing

is then carried out on the incoming traffic at the virtual server. Therefore, before enabling the SSL virtual server on the appliance, you need to bind a valid SSL certificate to the SSL virtual server.

Bind an SSL certificate key pair to a virtual server by using the CLI

At the command prompt, type the following commands to bind an SSL certificate key pair to a virtual server and verify the configuration:

```
-  bind ssl vserver <vServerName> -certkeyName <string>
-  show ssl vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1

Done

> show ssl vserver Vserver-SSL-1



     Advanced SSL configuration for VServer Vserver-SSL-1:

     DH: DISABLED

     Ephemeral RSA: ENABLED Refresh Count: 0

     Session Reuse: ENABLED Timeout: 120 seconds

     Cipher Redirect: ENABLED

     SSLv2 Redirect: ENABLED

     ClearText Port: 0

```



```
32
33      Client Auth: DISABLED
34
35
36      SSL Redirect: DISABLED
37
38
39      Non FIPS Ciphers: DISABLED
40
41
42      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
43
44
45
46
47
48 1) CertKey Name: CertKey-SSL-1 Server Certificate
49
50
51 1) Cipher Name: DEFAULT
52
53
54    Description: Predefined Cipher Alias
55
56
57 Done
58 <!--NeedCopy-->
```

Bind an SSL certificate key pair to a virtual server by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL Offload > Virtual Servers**.
2. Select the virtual server to which you want to bind the certificate key pair, for example, Vserver-SSL-1, and click Open.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, on the **SSL Settings** tab, under **Available**, select the certificate key pair that you want to bind to the virtual server. Then click **Add**.
4. Click **OK**.
5. Verify that the certificate key pair that you selected appears in the Configured area.

Configure support for Outlook web access

If you use Outlook Web Access (OWA) servers on your Citrix ADC appliance, you must configure the appliance to insert a special header field, FRONT-END-HTTPS: ON, in HTTP requests directed to the OWA servers, so that the servers generate URL links as `https://` instead of `http://`.

Note: You can enable OWA support for HTTP-based SSL virtual servers and services only. You cannot apply it for TCP-based SSL virtual servers and services.

To configure OWA support, do the following:

- Create an SSL action to enable OWA support.
- Create an SSL policy.
- Bind the policy to the SSL virtual server.

Create an SSL action to enable OWA support

Before you can enable Outlook Web Access (OWA) support, you must create an SSL action. SSL actions are bound to SSL policies and triggered when incoming data matches the rule specified by the policy.

Create an SSL action to enable OWA support by using the CLI

At the command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

```
1 -  add ssl action <name> -OWASupport ENABLED
2 -  show SSL action <name>
3 <!--NeedCopy-->
```

Example:

```
1     > add ssl action Action-SSL-OWA -OWASupport enabled
2
3
4
5
6     Done
7
8
9     > show SSL action Action-SSL-OWA
10
11
12    Name: Action-SSL-OWA
13
14
```



```
    Data Insertion Action: OWA


    Support: ENABLED


    Done
<!--NeedCopy-->
```

Create an SSL action to enable OWA support by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL > Policies**.
2. In the details pane, on the **Actions** tab, click **Add**.
3. In the **Create SSL Action** dialog box, in the Name text box, type Action-SSL-OWA.
4. Under Outlook Web Access, select **Enabled**.
5. Click **Create**, and then click **Close**.
6. Verify that Action-SSL-OWA appears in the **SSL Actions** page.

Create SSL policies

SSL policies are created by using the policy infrastructure. Each SSL policy has an SSL action bound to it, and the action is carried out when incoming traffic matches the rule that has been configured in the policy.

Create an SSL policy by using the CLI

At the command prompt, type the following commands to configure an SSL policy and verify the configuration:

```
-  add ssl policy <name> -rule <expression> -reqAction <string>
-  show ssl policy <name>
<!--NeedCopy-->
```

Example:

```
> add ssl policy Policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA

Done

> show ssl policy Policy-SSL-1

```



```
Name: Policy-SSL-1 Rule: ns_true

Action: Action-SSL-OWA Hits: 0

Policy is bound to following entities

1) PRIORITY : 0

Done
<!--NeedCopy-->
```

Create an SSL policy by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL > Policies**.
2. In the details pane, click **Add**.
3. In the **Create SSL Policy** dialog box, in the Name text box, type the name of the SSL Policy (for example, Policy-SSL-1).
4. In **Request** Action, select the configured SSL action that you want to associate with this policy (for example, Action-SSL-OWA). The ns_true general expression applies the policy to all successful SSL handshake traffic. However, to filter specific responses, you can create policies with a higher level of detail. For more information about configuring granular policy expressions, see SSL actions and policies.
5. In **Named Expressions**, choose the built-in general expression ns_true and click **Add Expression**. The expression ns_true now appears in the Expression text box.
6. Click **Create**, and then click **Close**.
7. Verify that the policy is correctly configured by selecting the policy and viewing the Details section at the bottom of the pane.

Bind the SSL policy to the SSL virtual server

After you configure an SSL policy for Outlook Web Access, bind the policy to a virtual server that will intercept incoming Outlook traffic. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

Bind an SSL policy to an SSL virtual server by using the CLI

At the command prompt, type the following commands to bind an SSL policy to an SSL virtual server and verify the configuration:


```
-  bind ssl vserver <vServerName> -policyName <string>
-  show ssl vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1

 Done

> show ssl vserver Vserver-SSL-1

Advanced SSL configuration for VServer Vserver-SSL-1:

DH: DISABLED

Ephemeral RSA: ENABLED

Refresh Count: 0

Session Reuse: ENABLED

Timeout: 120 seconds

Cipher Redirect: ENABLED

SSLv2 Redirect: ENABLED

ClearText Port: 0

Client Auth: DISABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: CertKey-SSL-1 Server Certificate

1) Policy Name: Policy-SSL-1 Priority: 0

1) Cipher Name: DEFAULT Description: Predefined Cipher Alias

```



```
Done
<!--NeedCopy-->
```

Bind an SSL policy to an SSL virtual server by using the GUI

Follow these steps:

1. Navigate to **Traffic Management > SSL Offload > Virtual Servers**.
2. In the details pane, select the virtual server (for example, Vserver-SSL-1), and then click **Open**.
3. In the **Configure Virtual Server (SSL Offload)** dialog box, click **Insert Policy**, and then select the policy that you want to bind to the SSL virtual server. Optionally, you can double-click the Priority field and type a new priority level.
4. Click **OK**.

Features at a glance

September 14, 2021

Citrix ADC features can be configured independently or in combinations to address specific needs. Although some features fit more than one category, the numerous Citrix ADC features can generally be categorized as application switching and traffic management features, application acceleration features, and application security and firewall features, and an application visibility feature.

To understand the order in which the features perform their processing, see Processing Order of Features section.

Application switching and traffic management features

September 14, 2021

Below are the application switching and traffic management features.

SSL Offloading

Transparently offloads SSL encryption and decryption from web servers, freeing server resources to service content requests. SSL places a heavy burden on an application's performance and can render many optimization measures ineffective. SSL offload and acceleration allow all the benefits of Citrix Request Switching technology to be applied to SSL traffic, ensuring secure delivery of web applications without degrading end-user performance.

For more information, see SSL offload and acceleration.

Access Control Lists

Compares incoming packets to Access Control Lists (ACLs). If a packet matches an ACL rule, the action specified in the rule is applied to the packet. Otherwise, the default action (ALLOW) is applied and the packet is processed normally. For the appliance to compare incoming packets to the ACLs, you have to apply the ACLs. All ACLs are enabled by default, but you have to apply them in order for the Citrix ADC appliance to compare incoming packets against them. If an ACL is not required to be a part of the lookup table, but still needs to be retained in the configuration, it should be disabled before the ACLs are applied. An ADC appliance does not compare incoming packets to disabled ACLs.

For more information, see Access Control List.

Load Balancing

Load balancing decisions are based on a variety of algorithms, including round robin, least connections, weighted least bandwidth, weighted least packets, minimum response time, and hashing based on URL, domain source IP, or destination IP. Both the TCP and UDP protocols are supported, so the Citrix ADC appliance can load balance all traffic that uses those protocols as the underlying carrier (for example, HTTP, HTTPS, UDP, DNS, NNTP, and general firewall traffic). In addition, the ADC appliance can maintain session persistence based on source IP, cookie, server, group, or SSL session. It allows users to apply custom Extended Content Verification (ECV) to servers, caches, firewalls and other infrastructure devices to ensure that these systems are functioning properly and are providing the right content to users. It can also perform health checks using ping, TCP, or HTTP URL, and the user can create monitors based on Perl scripts.
To provide high-scale WAN optimization, the CloudBridge appliances deployed at data centers can be load balanced through Citrix ADC appliances. The bandwidth and number of concurrent sessions can be improved significantly.

For more information, see Load Balancing.

Traffic Domains

Traffic domains provide a way to create logical ADC partitions within a single Citrix ADC appliance. They enable you to segment network traffic for different applications. You can use traffic domains to create multiple isolated environments whose resources do not interact with each other. An application belonging to a specific traffic domain communicates only with entities, and processes traffic, within that domain. Traffic belonging to one traffic domain cannot cross the boundary of another traffic domain. Therefore, you can use duplicate IP addresses on the appliance as long as an addresses is not duplicated within the same domain.

For more information, see Traffic Domains.

Network Address Translation

Network address translation (NAT) involves modification of the source and/or destination IP addresses, and/or the TCP/UDP port numbers, of IP packets that pass through the Citrix ADC appliance. Enabling NAT on the appliance enhances the security of your private network, and protects it from a public network such as the Internet, by modifying your network's source IP addresses when data passes through the Citrix ADC appliance.

The Citrix ADC appliance supports the following types of network address translation:

INAT: In Inbound NAT (INAT), an IP address (usually public) configured on the Citrix ADC appliance listens to connection requests on behalf of a server. For a request packet received by the appliance on a public IP address, the ADC replaces the destination IP address with the private IP address of the server. In other words, the appliance acts as a proxy between clients and the server. INAT configuration involves INAT rules, which define a 1:1 relationship between the IP address on the Citrix ADC appliance and the IP address of the server.

RNAT: In Reverse Network Address Translation (RNAT), for a session initiated by a server, the Citrix ADC appliance replaces the source IP address in the packets generated by the server with an IP address (type SNIP) configured on the appliance. The appliance thereby prevents exposure of the server's IP address in any of the packets generated by the server. An RNAT configuration involves an RNAT rule, which specifies a condition. The appliance performs RNAT processing on those packets that match the condition.

Stateless NAT46 Translation: Stateless NAT46 enables communication between IPv4 and IPv6 networks, by way of IPv4 to IPv6 packet translation and vice versa, without maintaining any session information on the Citrix ADC appliance. A stateless NAT46 configuration involves an IPv4-IPv6 INAT rule and an NAT46 IPv6 prefix.

Stateful NAT64 Translation: The stateful NAT64 feature enables communication between IPv4 clients and IPv6 servers through IPv6 to IPv4 packet translation, and vice versa, while maintaining session information on the Citrix ADC appliance. A stateful NAT64 configuration involves an NAT64 rule and an NAT64 IPv6 prefix.

For more information, see Configuring Network Address Translation.

Multipath TCP Support

Citrix ADC appliances support Multipath TCP (MPTCP). MPTCP is a TCP/IP protocol extension that identifies and uses multiple paths available between hosts to maintain the TCP session. You must enable MPTCP on a TCP profile and bind it to a virtual server. When MPTCP is enabled, the virtual server func-

tions as an MPTCP gateway and converts MPTCP connections with the clients to TCP connections that it maintains with the servers.

For more information, see MPTCP (Multi-Path TCP).

Content Switching

Determines the server to which to send the request on the basis of configured content switching policies. Policy rules can be based on the IP address, URL, and HTTP headers. This allows switching decisions to be based on user and device characteristics such as who the user is, what type of agent is being used, and what content the user requested.

For more information, see Content Switching.

Global Server Load Balancing (GSLB)

Extends the traffic management capabilities of a NetScaler to include distributed Internet sites and global enterprises. Whether installations are spread across multiple network locations or multiple clusters in a single location, the NetScaler maintains availability and distributes traffic across them. It makes intelligent DNS decisions to prevent users from being sent to a site that is down or overloaded. When the proximity-based GSLB method is enabled, the NetScaler can make load balancing decisions based on the proximity of the client's local DNS server (LDNS) in relation to different sites. The main benefit of the proximity-based GSLB method is faster response time resulting from the selection of the closest available site.

For more information, see Global Server Load Balancing.

Dynamic Routing

Enables routers to obtain topology information, routes, and IP addresses from neighboring routers automatically. When dynamic routing is enabled, the corresponding routing process listens to route updates and advertises routes. The routing processes can also be placed in passive mode. Routing protocols enable an upstream router to load balance traffic to identical virtual servers hosted on two standalone NetScaler units using the Equal Cost Multipath technique.

For more information, see Configuring Dynamic Routes.

Link Load Balancing

Load balances multiple WAN links and provides link failover, further optimizing network performance and ensuring business continuity. Ensures that network connections remain highly available, by applying intelligent traffic control and health checks to distribute traffic efficiently across upstream

routers. Identifies the best WAN link to route both incoming and outbound traffic based on policies and network conditions, and protects applications against WAN or Internet link failure by providing rapid fault detection and failover.

For more information, see Link Load Balancing.

TCP Optimization

You can use TCP profiles to optimize TCP traffic. TCP profiles define the way that NetScaler virtual servers process TCP traffic. Administrators can use the built-in TCP profiles or configure custom profiles. After defining a TCP profile, you can bind it to a single virtual server or to multiple virtual servers.

Some of the key optimization features that can be enabled by TCP profiles are:

- TCP keep-alive—Checks the operational status of the peers at specified time intervals to prevent the link from being broken.
- Selective Acknowledgment (SACK)— Improves the performance of data transmission, especially in long fat networks (LFNs).
- TCP window scaling— Allows efficient transfer of data over long fat networks (LFNs).

For more information on TCP Profiles, see Configuring TCP Profiles.

CloudBridge Connector

The Citrix NetScaler CloudBridge Connector feature, a fundamental part of the Citrix OpenCloud framework, is a tool used to build a cloud-extended data center. The OpenCloud Bridge enables you to connect one or more Citrix ADC appliances or NetScaler virtual appliances on the cloud-to your network without reconfiguring your network. Cloud hosted applications appear as though they are running on one contiguous enterprise network. The primary purpose of the OpenCloud Bridge is to enable companies to move their applications to the cloud while reducing costs and the risk of application failure. In addition, the OpenCloud Bridge increases network security in cloud environments. An OpenCloud Bridge is a Layer-2 network bridge that connects a Citrix ADC appliance or NetScaler virtual appliance on a cloud instance to a Citrix ADC appliance or NetScaler virtual appliance on your LAN. The connection is made through a tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. Then Internet Protocol security (IPsec) protocol suite is used to secure the communication between the peers in the OpenCloud Bridge.

For more information, see CloudBridge.

DataStream

The NetScaler DataStream feature provides an intelligent mechanism for request switching at the database layer by distributing requests on the basis of the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, user names, character sets, and packet size.

You can configure load balancing to switch requests according to load balancing algorithms, or you can elaborate the switching criteria by configuring content switching to make a decision based on SQL query parameters, such as user name, database names, and command parameters. You can further configure monitors to track the states of database servers.

The advanced policy infrastructure on the Citrix ADC appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with MYSQL.CLIENT and MYSQL.REQ) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

> Note: DataStream is supported for MySQL and MS SQL databases.

For more information, see DataStream.

Application acceleration features

September 14, 2021

- AppCompress

  Uses the gzip compression protocol to provide transparent compression for HTML and text files. The typical 4:1 compression ratio yields up to 50% reduction in bandwidth requirements out of the data center. It also results in significantly improved end-user response time, because it reduces the amount of data that must be delivered to the user's browser.

- Cache Redirection

  Manages the flow of traffic to a reverse proxy, transparent proxy, or forward proxy cache farm. Inspects all requests, and identifies non-cacheable requests and sends them directly to the origin servers over persistent connections. By intelligently redirecting non-cacheable requests back to the origin web servers, the Citrix ADC appliance frees cache resources and increases cache hit rates while reducing overall bandwidth consumption and response delays for these requests.

  For more information, see Cache Redirection.

- AppCache

  Helps optimize web content and application data delivery by providing a fast in-memory HTTP/1.1 and HTTP/1.0 compliant web caching for both static and dynamic content. This on-board cache stores the results of incoming application requests even when an incoming request is secured or the data compressed, and then reuses the data to fulfill subsequent requests for the same information. By serving data directly from the on-board cache, the appliance can reduce page regeneration times by eliminating the need to funnel static and dynamic content requests to the server.

  For more information, see Integrated Caching.

- TCP Buffering

  Buffers the server's response and delivers it to the client at the client's speed, thus offloading the server faster and thereby improving the performance of web sites.

Application security and firewall features

September 14, 2021

Below are the security and firewall features.

Denial of service (DoS) attack defense

Detects and stops malicious distributed denial-of-service (DDoS) attacks and other types of malicious attacks before they reach your servers, preventing them from affecting network and application performance. The Citrix ADC appliance identifies legitimate clients and elevates their priority, leaving suspect clients unable to consume a disproportionate percentage of resources and cripple your site. The appliance provides application-level protection from the following types of malicious attacks:

- SYN flood attacks
- Pipeline attacks
- Teardrop attacks
- Land attacks
- Fraggle attacks
- Zombie connection attacks

The appliance aggressively defends against these types of attacks by preventing the allocation of server resources for these connections. This insulates servers from the overwhelming flood of packets associated with these events.

The appliance also protects network resources from ICMP based attacks by using ICMP rate limiting and aggressive ICMP packet inspection. It performs strong IP reassembly, drops a variety of suspicious and malformed packets, and applies Access Control Lists (ACLs) to site traffic for further protection.

For more information, see HTTP Denial-of-Service Protection.

Content Filtering

Provides protection from malicious attacks for web sites at the Layer 7 level. The appliance inspects each incoming request according to user-configured rules based on HTTP headers, and performs the action the user configured. Actions can include resetting the connection, dropping the request, or sending an error message to the user's browser. This allows the appliance to screen unwanted requests and reduces your servers' exposure to attacks.

This feature can also analyze HTTP GET and POST requests and filter out known bad signatures, allowing it to defend your servers against HTTP-based attacks.

For more information, see Content Filtering.

Responder

Functions like an advanced filter and can be used to generate responses from the appliance to the client. Some common uses of this feature are generation of redirect responses, user defined responses, and resets.

For more information, see Responder.

Rewrite

Modifies HTTP headers and body text. You can use the rewrite feature to add HTTP headers to an HTTP request or response, make modifications to individual HTTP headers, or delete HTTP headers. It also enables you to modify the HTTP body in requests and responses.

When the appliance receives a request or sends a response, it checks for rewrite rules, and if applicable rules exist, it applies them to the request or response before passing it on to the web server or client computer.

For more information, see Rewrite.

Priority Queuing

Prioritizes user requests to ensure that the most important traffic is serviced first during surges in request volume. You can establish priority based on request URLs, cookies, or a variety of other fac-

tors. The appliance places requests in a three-tier queue based on their configured priority, enabling business-critical transactions to flow smoothly even during surges or site attacks.

For more information, see Priority Queuing.

Surge Protection

Regulates the flow of user requests to servers and controls the number of users that can simultaneously access the resources on the servers, queuing any additional requests once your servers have reached their capacity. By controlling the rate at which connections can be established, the appliance blocks surges in requests from being passed on to your servers, thus preventing site overload.

For more information, see Surge Protection.

Citrix Gateway

Citrix Gateway is a secure application access solution that provides administrators granular application-level policy and action controls to secure access to applications and data while allowing users to work from anywhere. It gives IT administrators a single point of control and tools to help ensure compliance with regulations and the highest levels of information security across and outside the enterprise. At the same time, it empowers users with a single point of access—optimized for roles, devices, and networks—to the enterprise applications and data they need. This unique combination of capabilities helps maximize the productivity of today's mobile workforce.

For more information, see Citrix Gateway.

Application Firewall

Protects applications from misuse by hackers and malware, such as cross site scripting attacks, buffer overflow attacks, SQL injection attacks, and forceful browsing, by filtering traffic between each protected web server and users that connect to any web site on that web server. The application firewall examines all traffic for evidence of attacks on web server security or misuse of web server resources, and takes the appropriate action to prevent these attacks from succeeding.

For more information, see Application Firewall.

Application visibility feature

September 14, 2021

- Citrix Application Delivery Management

  Citrix Application Delivery Management (ADM) is a high performance collector that provides end-to-end user experience visibility across Web and HDX (ICA) traffic. It collects HTTP and ICA AppFlow records generated by Citrix ADC appliances and populates analytical reports covering Layer 3 to Layer 7 statistics. Citrix ADM provides in-depth analysis for the last five minutes of real-time data, and for historical data collected for the last one hour, one day, one week, and one month.

  HDX (ICA) analytic dashboard enables you to drill down from HDX Users, Applications, Desktops, and even from gateway-level information. Similarly, HTTP analytics provide a bird's eye view of Web Applications, URLs Accessed, Client IP Addresses and Server IP Addresses, and other dashboards. The administrator can drill down and identify the pain points from any of these dashboards, as appropriate for the use case.

- Enhanced Application Visibility Using AppFlow

  The Citrix ADC appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.

  Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

  AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

  To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

  You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

  For more information, see AppFlow.

- Stream Analytics

  The performance of your website or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you must be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify

the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources help you determine whether those resources must be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

If the website or application does not change frequently, you can use products that collect statistical data, and then manually analyze the statistics and optimize the delivery of content. However, if you do not want to perform manual optimizations, or if your website or application is dynamic in nature, you need infrastructure that can not only collect statistical data but can also automatically optimize the delivery of resources based on the statistics. On the Citrix ADC appliance, this functionality is provided by the Stream Analytics feature. The feature operates on a single Citrix ADC appliance and collects run-time statistics based on the criteria that you define. When used with Citrix ADC policies, the feature also provides you with the infrastructure that you need for automatic, real-time traffic optimization.

For more information, see Action Analytics.

Citrix ADC Solutions

September 14, 2021

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

This section includes the following solutions.

- Setting up Citrix ADC for Citrix Virtual Apps and Desktops
- Global Server Load Balancing (GSLB) Powered Zone Preference
- Anycast support in Citrix ADC
- Deploy digital advertising platform on AWS with Citrix ADC
- Enhancing Clickstream analytics in AWS using Citrix ADC
- Citrix ADC in a Private Cloud Managed by Microsoft Windows Azure Pack and Cisco ACI

Setting up Citrix ADC for Citrix Virtual Apps and Desktops

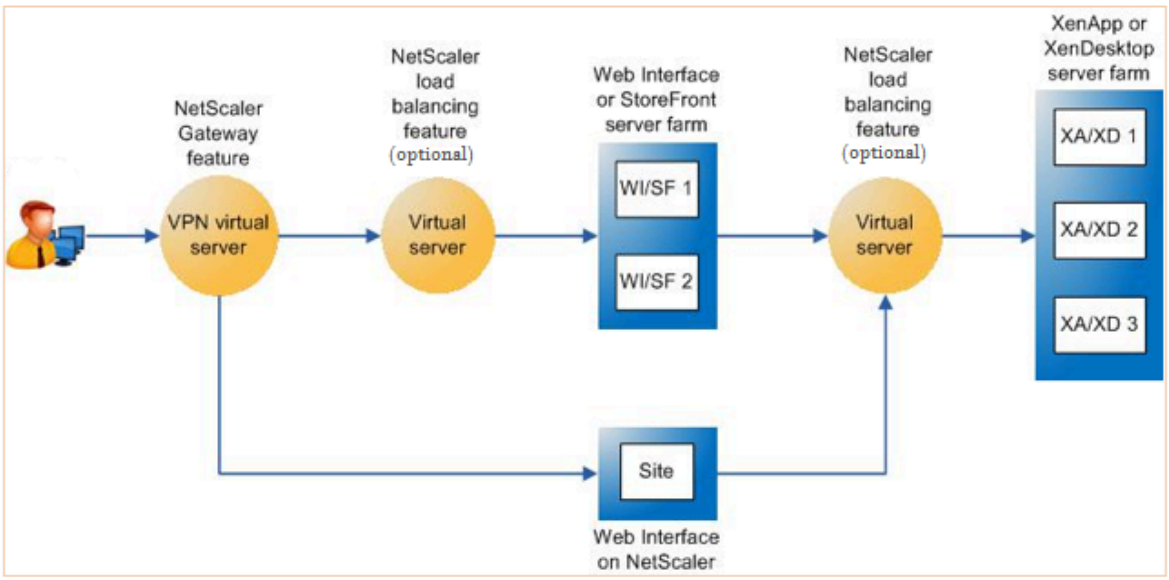
September 14, 2021

A Citrix ADC appliance can provide load balanced, secure remote access to your Citrix Virtual Apps and Desktops applications. You can use the Citrix ADC load balancing feature to distribute traffic across

the Citrix Virtual Apps and Desktops server. You can use the Citrix Gateway feature to provide secure remote access to the servers.

Citrix ADC can also accelerate and optimize the traffic flow and offer visibility features that are useful for Citrix Virtual Apps and Desktops deployments.

Figure 1. Citrix ADC appliance in Citrix Virtual Apps and Desktops setup



The preceding figure shows the components involved in this deployment:

- **NetScaler Gateway.** Provides the URL for user access, and provides security by authenticating the users.
- **Citrix ADC load balancing virtual server.** Load balances the traffic for the Web Interface or StoreFront servers. You can also deploy a load balancing virtual server in front of the Citrix Virtual Apps and Desktop servers to load balance key components such as XML Broker and Desktop Delivery Controller (DDC) server.
- **Web Interface or StoreFront or Web Interface on Citrix ADC.** Provides the interface through which you can access the applications.

  Note: Web Interface on Citrix ADC (WIonNS) is a customization of the Web Interface product, hosted on the Citrix ADC appliance.
- **Citrix Virtual Apps and Desktops.** Provides the applications that your users want to access.

To set up the Citrix ADC for Citrix Virtual Apps and Desktops by using the Citrix ADC GUI

Prerequisites

- Citrix Virtual Apps and Desktop servers are configured and available.
- Web Interface, StoreFront, or Web Interface on Citrix ADC servers are configured and available.
- You have a working knowledge of Citrix Gateway, Citrix ADC, Citrix Virtual Apps and Desktops, and StoreFront/Web Interface/Web Interface on Citrix ADC.
- Make sure that you have configured a virtual server and a service and bound the service to the virtual server. For more information, see:
  - Load balance XenDesktop
  - Load balance XenApp

Procedure:

1. Log on to the Citrix ADC appliance and on the **Configuration** tab click **XenApp and XenDesktop**.
2. On the **Details** pane, click **Get Started**. If the setup exists on the Citrix ADC, click the **Edit** link corresponding to each of the section that you want to modify.
3. Select the product (StoreFront, Web Interface, or Web Interface on Citrix ADC) that in your deployment provides the interface for access to the Citrix Virtual Apps and Desktops applications.
4. Set up secure remote access.
   a) In the **NetScaler Gateway Settings** section, specify the details for the VPN virtual server and click **Continue**.
   b) In the **Server Certificate** section, choose an existing certificate or install a new certificate and click **Continue**.
   c) In the **Authentication** section, configure the primary authentication mechanism to be used and specify the server details or use an existing server and click **Continue**.
   d) In the **StoreFront** section, specify the details of the server that provides the interface for accessing the applications and click **Continue**.
   e) You can use one of the following as your StoreFront server.
      i. LB virtual server pointing to multiple SF servers.
      ii. Web Interface or StoreFront server directly reachable from the Citrix ADC appliance.
      iii. Web Interface on Citrix ADC.
5. Click **Done** to complete the configuration.

Global Server Load Balancing (GSLB) Powered Zone Preference

September 14, 2021

GSLB powered zone preference is a feature that integrates Citrix Virtual Apps and Desktops, StoreFront, and Citrix ADC to provide clients access to the most optimized data center based on the client location.

In a distributed Citrix Virtual Apps and Desktops deployment, StoreFront might not select an optimal data center when multiple equivalent resources are available from multiple data centers. In such cases, StoreFront randomly selects a data center. It can send the request to any of the Citrix Virtual Apps and Desktops servers in any data center, regardless of proximity to the client making the request.

The client IP address is examined when an HTTP request arrives at the Citrix Gateway appliance. The real client IP address is used to create the data center preference list that is forwarded to StoreFront. If the Citrix ADC appliance is configured to insert the zone preference header, StoreFront 3.5 or later can use the information provided by the appliance to reorder the list of delivery controllers and connect to an optimal delivery controller in the same zone as the client. StoreFront selects the optimal gateway VPN virtual server for the selected data center zone, adds this information to the ICA file with appropriate IP addresses, and sends it to the client. StoreFront then tries to launch applications hosted on the preferred data center's delivery controllers before trying to contact equivalent controllers in other data centers.

For more information about configuring this solution, click here.

For a video overview about GSLB powered zone preference solution, click https://www.youtube.com/watch?v=Y8DELum0Xp0.

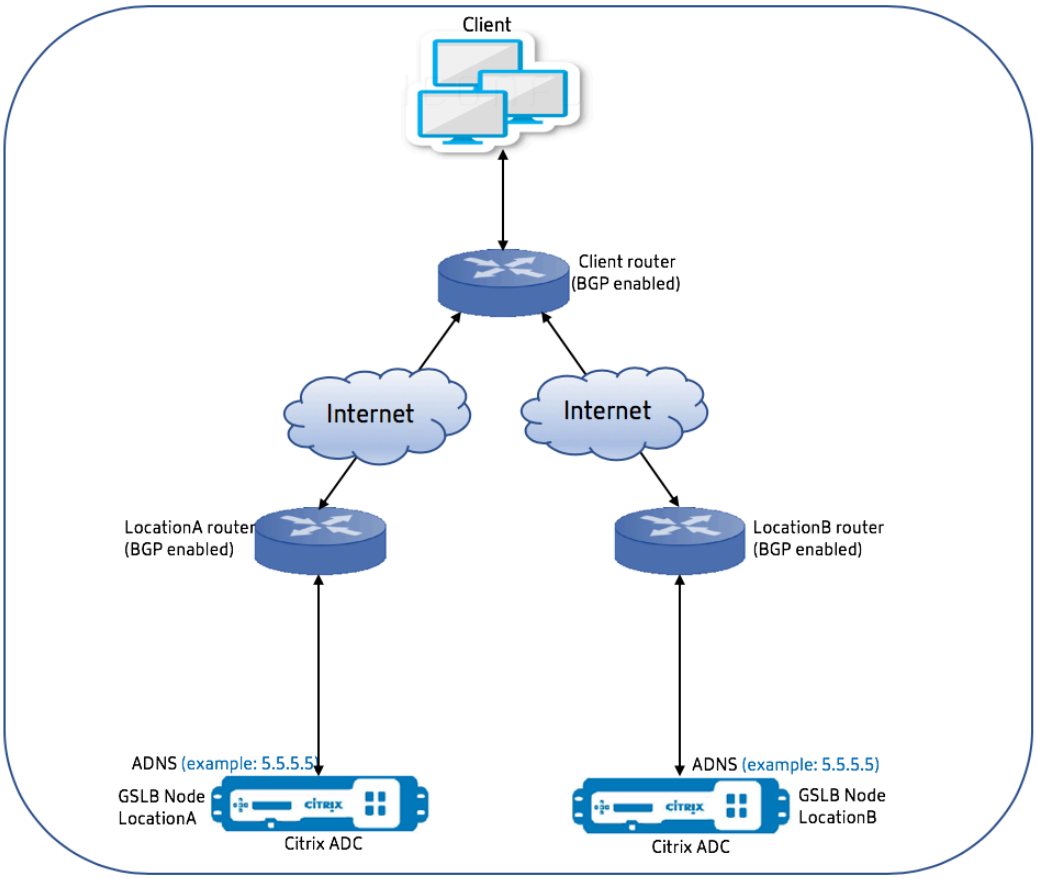
Anycast support in Citrix ADC

September 14, 2021

Anycast is a type of network where a set of servers shares an IP address. The client request is directed to the topographically closest server based on their routing tables. This routing reduces latency issues, ensures high availability, and minimizes downtime.

Citrix ADC supports anycast network with Global Server Load Balancing (GSLB) and DNS features.

The following diagram illustrates a topology diagram of Anycast in Citrix ADC.



Anycast GSLB

The Citrix ADC GSLB feature provides load balancing across globally distributed sites along with disaster recovery and ensures continuous availability of applications.

During an outage, GSLB provides immediate disaster recovery by routing traffic to the closest or the best performing data center. However, GSLB cannot control the following:

- How the DNS traffic is routed to GSLB nodes in different geographical locations.
- How much latency is getting added while DNS queries get routed to GSLB nodes.

In a typical GSLB setup, each data center has a GSLB node configured with the site-specific Authoritative Domain Name Server (ADNS) to receive DNS queries. Each site's ADNS is configured as the nameserver in the DNS resolver. As the number of GSLB nodes increases, the number of nameserver records also increase. In such cases, if there is a failure of a data center, LDNS has to retry resolution with a different nameserver. This retry adds to the latency in DNS resolution.
Also, every time a GSLB node is added, the nameserver records must be updated.

To overcome these drawbacks, you can use Anycast ADNS. In Anycast ADNS, a single ADNS IP address is used for all GSLB nodes and the DNS traffic is routed to GSLB nodes using dynamic routing.

For example, if a GSLB site is DOWN, the routing table is updated and route to this site is removed. As a result, The DNS queries are not sent to the sites that are DOWN. As a result, there are no retries.

If a new GSLB node is added, the new node is assigned the same ADNS IP address. The dynamic routing automatically updates the routing tables with routes to new sites based on the routing algorithms. Hence, you do not have to update the DNS name server records. The rollout of new GSLB sites is made simpler and faster with Anycast.

How to configure an ADNS IP address in an anycast mode

Enable host routing on the ADNS IP in a Citrix ADC appliance, and set the appropriate Route Health Injection (RHI) level. Mostly, there would not be any virtual servers on the ADNS IP and therefore RHI level must be selected as NONE. Enabling host route on the ADNS IP makes it a kernel route. You can then enable the dynamic routing of choice and configure the routing protocol to redistribute the kernel routes.

ADNS IP configuration – Example

At the command prompt, type;

```
add service adns_public 5.5.5.5 ADNS 53

set ip 5.5.5.5 -hostRoute ENABLED -vserverRHILevel ALL_VSERVERS
<!--NeedCopy-->
```

BGP configuration in GSLB site – Example

```
Site1#sh run
!
hostname Site1
!
log syslog
log record-priority
!
ns route-install bgp
!
interface lo0
 ip address 127.0.0.1/8
 ipv6 address fe80::1/64
 ipv6 address ::1/128
!
interface vlan0
 ip address 10.102.148.94/25
```



```
 ipv6 address fe80::e84c:f4ff:fe74:4588/64
!
interface vlan2
 ip address 172.18.30.15/24
!
router bgp 5
 redistribute kernel -----> redistributing the kernel routes
 neighbor 172.18.30.30 remote-as 4
 neighbor 172.18.30.30 advertisement-interval 1
 neighbor 172.18.30.30 timers 4 16
!
End

Site1#
<!--NeedCopy-->
```

GSLB site routing table - Example

```
Site1#sh ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, I - Intranet
       * - candidate default

K       5.5.5.5/32 via 0.0.0.0 ---------------------------------------->
    Kernel Route for ADNS
C       10.102.148.0/25 is directly connected, vlan0
C       127.0.0.0/8 is directly connected, lo0
B       172.18.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       172.18.20.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
C       172.18.30.0/24 is directly connected, vlan2
B       192.168.3.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.5.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h

Gateway of last resort is not set
Site1#
<!--NeedCopy-->
```


Anycast DNS

You can use Anycast DNS for DNS proxy virtual servers on Citrix ADC. When there are multiple DNS name servers configured, the DNS resolver responds based on round robin method. For example, if the resolver does not receive any response from the first server, it switches to the second server after the configured timeout value expires. The switching from first server to second server adds to the latency in DNS resolution. If the DNS resolvers are configured with Anycast, then this latency can be eliminated.

DNS configuration – Example

At the command prompt, type;

```
add lb vserver dns DNS 5.5.5.50 53

set ip 5.5.5.50 -hostRoute ENABLED -vserverRHILevel ALL_VSERVERS
<!--NeedCopy-->
```

Deploy digital advertising platform on AWS with Citrix ADC

September 14, 2021

With the evolving nature of digital platforms, a wide range of advertising applications are available. For example, social media, direct mail, videos, banners, pops, interstitials, rich media and so on. Advertisers are embracing video advertising networks at a fast pace, constituting nearly 40% of the advertisement traffic. But with more usage of mobiles by the modern users, running video ads on the mobile platform has seen considerable surge.

The digital advertising platforms face several challenges. Some of the challenges are:

- Security threats
- High operational costs
- Wide range of devices are available to send traffic over the internet. The different protocols for real-time communication pose the following challenges:
  - webRTC
  - Adaptive streaming
  - UDP for video, where WebRTC uses UDP over HTTP

To deal with the complex behavior of advertising platforms, Citrix ADC solution with its whole suite of capabilities and features well integrated with AWS, provides an instant, secure, and reliable access to digital advertisement inventory, anywhere and anytime. Citrix ADC plays a critical role in delivering the SaaS and web apps for digital platforms.

Digital advertising platform integration with Citrix ADC

Digital advertising platform overview

The digital advertising platform consists of the following key components:

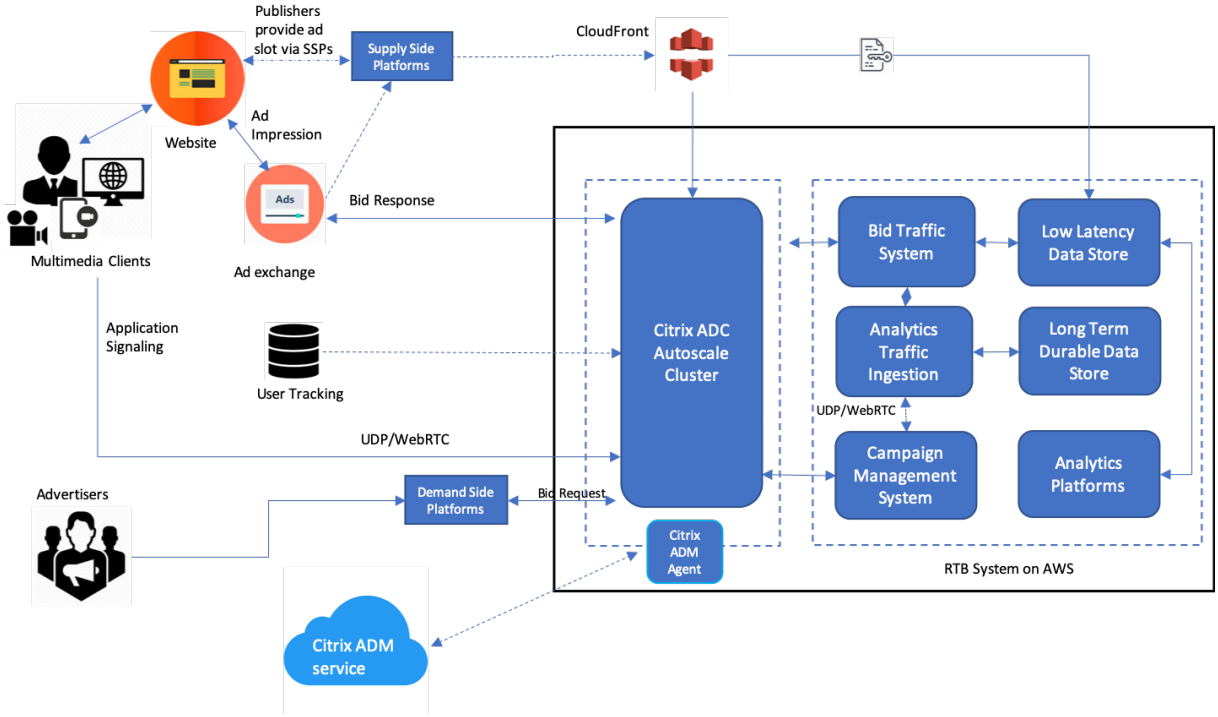
- Ad exchange
- Ad network
- Demand side platform (DSP)
- Supply side platform (SSP)
- Real time bidding (RTB) systems

An overview of the process followed in an advertising system is as follows.

- The first transaction happens when the user visits the website.
- This triggers a bid/advertisement request (including user's demographic information) that is sent to the ad server or publisher contacting an ad exchange.
- The Ad publishers send the advertisement request to an Ad exchange through SSPs.
- The Ad exchange submits this request and the accompanying data to DSP telling there is an impression or advertisement request available. Therefore, multiple advertisers can automatically submit bids in real time to place their advertisements.
- Meanwhile, the advertisers must set up their campaigns in DSP. Use the information about the user from Data Management Platform (DMP) to assess the amount they are willing to pay for delivering an advertisement to the user.
- DSPs submit these real-time bids on each advertisement impression because it is served to the advertisement exchange.
- Whichever bidder bids the most within a time period set by the Ad exchange or SSPs, gets an advertisement slot by the publishers to serve their advertisements. Otherwise, they lose the opportunity to get the right advertisement for their key demographic.

How digital advertising platform is integrated with Citrix ADC

The following diagram illustrates how the different components of advertising platform communicate with Citrix ADC and Citrix Application Delivery Management (ADM) to serve online advertisements.



How Citrix ADC contributes

In the advertisement publishing process, Citrix ADC solution helps in handling and processing the inconsistent influx of bid traffic. It acts as an entry point for all traffic to ensure scalability and availability across the availability zones. To cater to the elastic nature of advertising traffic, it is deployed in an autoscaling group in front of web applications and database servers.

The advertising platform on AWS with Citrix ADC solution allows you to get the real-time performance, high scalability, and high availability across the globe. You can buy and sell rich media, video, mobile, and native advertisements in real time. It reduces the overall operational cost and latency involved in running an Advertising platform. It is the best performing proxy with the rich capabilities of gracefully removing the back-end servers during Autoscale, connection multiplexing, and ensuring that the end-user traffic never gets impacted. Citrix ADC supports load balancing the HTTP, UDP, WebRTC, and RTSP protocols that are used in the advertising platforms.

Citrix ADC fits coherently into the AWS environment with the following key attributes:

- Content switching – Switch to the right platform based on host name.
- Security protection – Use web application firewall (WAF) functionality, rate limiting (through Client IP), and protection against DDoS attacks.
- Autoscaling of both front-end and back-end traffic.
- End-to-end visibility, and anomaly detection across ADC appliances by utilizing ADM.
- Low latency.

How Citrix ADM contributes

Citrix ADC utilizes Citrix ADM to overcome the following challenges faced by the digital advertising platforms:

- Identify the trend deviations from expected performance
- Real-time application performance analysis
- Capacity monitoring

Advantages of advertising platform integration with Citrix ADC and ADM

The Citrix ADC solution offers the following capabilities and benefits to a digital advertising platform vendor.

Low cost

- Integrated with AWS Autoscaling service, the Citrix ADC VPX instance can scale up or down your front-end and back-end resources automatically. This provides a zero-touch configuration catering to the elasticity of advertising platforms.
- Consolidation of delivering all types of traffic from a single point.

For more information on AWS autoscaling, see Add back-end AWS Autoscaling service.

High availability

- If one Availability Zone becomes unavailable, Citrix ADC applies its fault tolerance ability to autodetect the servers in another availability zone, without any traffic interruption.
- Also, It gracefully terminates servers avoiding the loss of client connections.

For more information, see How high availability on AWS works.

Application performance analytics

Citrix ADM intelligent analytics and application performance analytics ensures to:

- Gain visibility into the issues (server response anomalies, 5XX errors, and so on) plaguing the end user experience.
- Alert the administrator to take corrective actions immediately.

For more information, see Performance indicators for application analytics.

Rich firewall security

Most common security vulnerabilities occur in web applications rather in networks. It is vital to protect your web applications from unauthorized access such as bots, data thefts, and application layer attacks.

Citrix ADC provides comprehensive and integrated Layer 4 to Layer 7 security that includes:

- Web App Firewall (WAF) to protect your web applications, identify, and mitigate malicious bots with regularly updated bot signatures and behaviour-based detection.
- Rate limiting to prevent an advertising platform from being overwhelmed.

For more information, see Citrix Web App Firewall.

Select the right AWS instance type for advertising platform

Choose the right AWS instance type for ADC depending on the following two factors:

- Number of users simultaneously accessing the advertising platform.
- Average number of users on the platform.

The Citrix ADC can be deployed in various EC2 instances, which include c5, c5n, m5, and so on. For advertising platforms, use the following AWS instance types:

- c5 or c5n is appropriate for handling SSL heavy traffic.
- c5.large can handle up to 1000 SSL TPS.

For more information, see VPX-AWS support matrix.

Enhancing Clickstream analytics in AWS using Citrix ADC

September 14, 2021

Customers are increasingly accessing the company products through various applications such as Mobile apps, SaaS apps and so on. Therefore, applications can become a landmine of customer experience data. To track customer behavior online, customer-centric companies form data-driven profiles for each of their customers using this customer behavior data.

A clickstream is a sequence or stream of events that represent user actions (clicks) on a website or a mobile application. However, the scope of clickstream extends beyond clicks. It includes product searches, impressions, purchases, and any such events that might be of relevance to the business. Mere collecting and storing the customer experience data is not of much value. There is a need to distribute the highly complex data seamlessly to the right vendors at the right time. Businesses can derive value from the data and quickly take conscious decisions to improve upon their strategies. Therefore companies increasingly use clickstream analytics to glean insights into the customer experience journey of the apps.

This document provides you a good understanding on why Clickstream data is of utmost importance, how it is collected, stored, distributed, and transformed into meaningful and actionable analytics.

Citrix ADC integrates with Citrix ADM, and adds value to AWS services such as Amazon Kinesis Data Firehose to equip businesses with the best-in-class analytics solution that revolves around user's Clickstreams.

This Citrix ADC solution helps you to solve complex business issues efficiently and with extreme simplicity. Citrix ADC and AWS Kinesis help to capture the issues with the poorly designed workflow. Citrix ADM helps to capture web app and network performance related issues by applying relevant filters. Conjunction of Citrix ADC with Citrix ADM and AWS Kinesis helps you to manage and analyze the huge influx of clickstream data in each phase. This solution is highly available, scalable, robust, and ensures the delivery is continuous and secure. Thus, you can derive actionable insights.

Why businesses opt for Clickstream analytics?

Businesses opt for clickstream primarily to understand how users interact with the application, and to get insights on improving the goals of the application. Clickstream Analytics is an information retrieval use case that tracks your user's behavior, navigation habits, and so on. Clickstream analytics gives you information on:

- Which link your customers are clicking more often and at what point in time.
- Where was the visitor before reaching my website?
- How much time did the visitor spend on each page?
- When and where did the visitor click the "back" button on the web browser?
- What items did the visitor add to (or remove from) their shopping cart?
- From which page did the visitor exit my website?

Analytics service to manage Clickstream data using Amazon Kinesis

You can use Amazon Kinesis to perform clickstream analytics. Amazon kinesis enables clickstream analytics with the following services:

- Amazon Kinesis Data Firehose
- Amazon Kinesis Data Analytics
- Amazon Kinesis Data Streams

With Amazon Kinesis, you can collect and analyze your huge data sets at any scale. AWS Kinesis can handle data from various sources, such as:

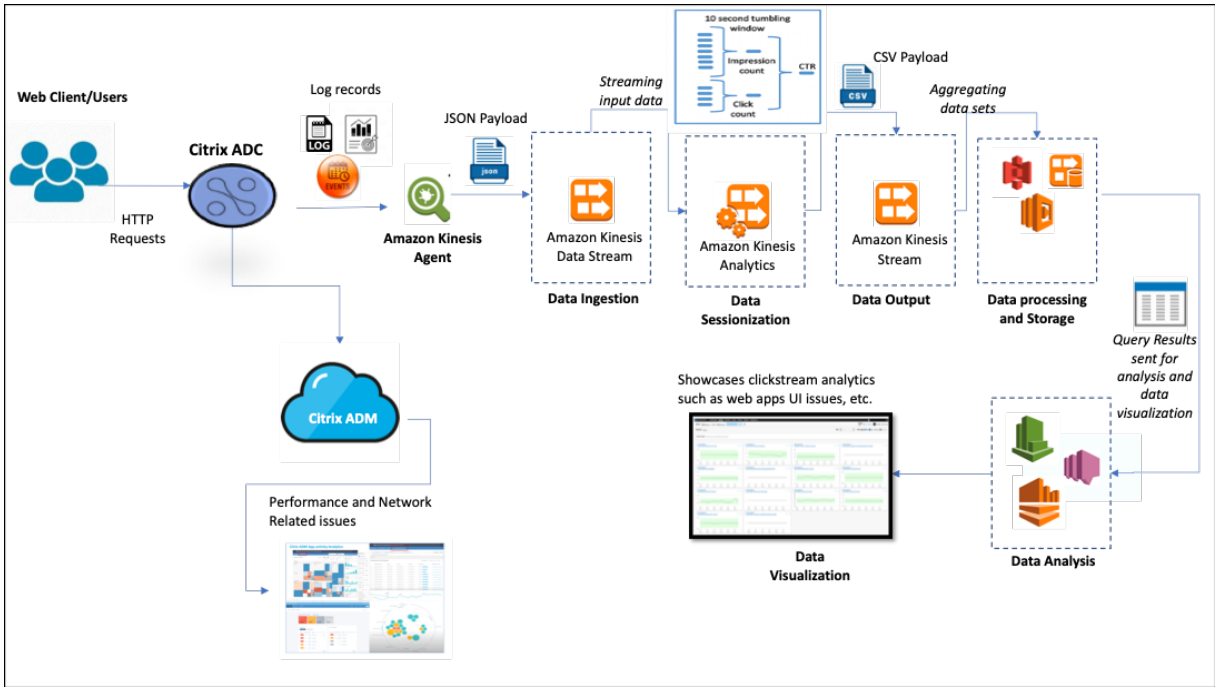
- Mobile and web applications (for example, Gaming, ecommerce)
- IoT devices
- Social networking applications
- Financial trading services

- Geospatial services

How Citrix ADC enables Clickstream analytics

The Citrix ADC solution collates and delivers information securely on the activities of users, such as, websites visited, the bandwidth spent, navigation flow. Companies analyze this high throughput and continuous clickstream data to corroborate the effectiveness of the following:

- Site layout
- Marketing campaigns
- New application features



With the Citrix ADC's ability to provide a resilient network protection for enterprise environments, the server cost is reduced manifolds by offloading computationally intensive tasks, and running sessions on this data. Thereby helping companies to identify events in real-time with high availability, security, and low latency always.

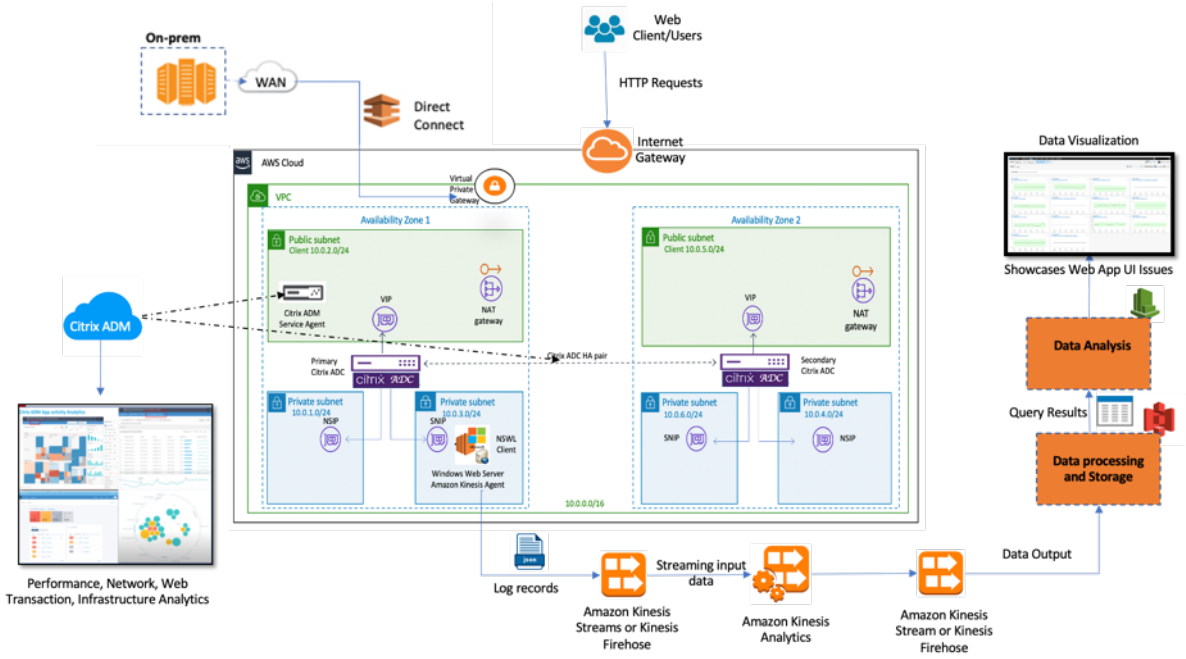
For configuration information, see Configure the Citrix ADC solution for clickstream analytics.

How Citrix ADC and Citrix ADM complement the AWS environment

The following diagram illustrates the end-to-end user workflow to perform Clickstream analytics in AWS infrastructure. This diagram helps you understand the following processes:

- How user interacts with Citrix ADC
- How Citrix ADC captures user's actions and generates clickstream data

- How the clickstream data is delivered to AWS services (Amazon Kinesis)
- How Amazon Kinesis processes the data logs and stores them to produce meaningful clickstream analytics



The Citrix ADC seamlessly integrates into the AWS environment and Citrix ADM that helps businesses to be compatible with variable volume and diverse nature of the clickstream data. It provides services to load and analyze streaming knowledge with simplicity. You can also create custom streaming knowledge applications for specialized desires.

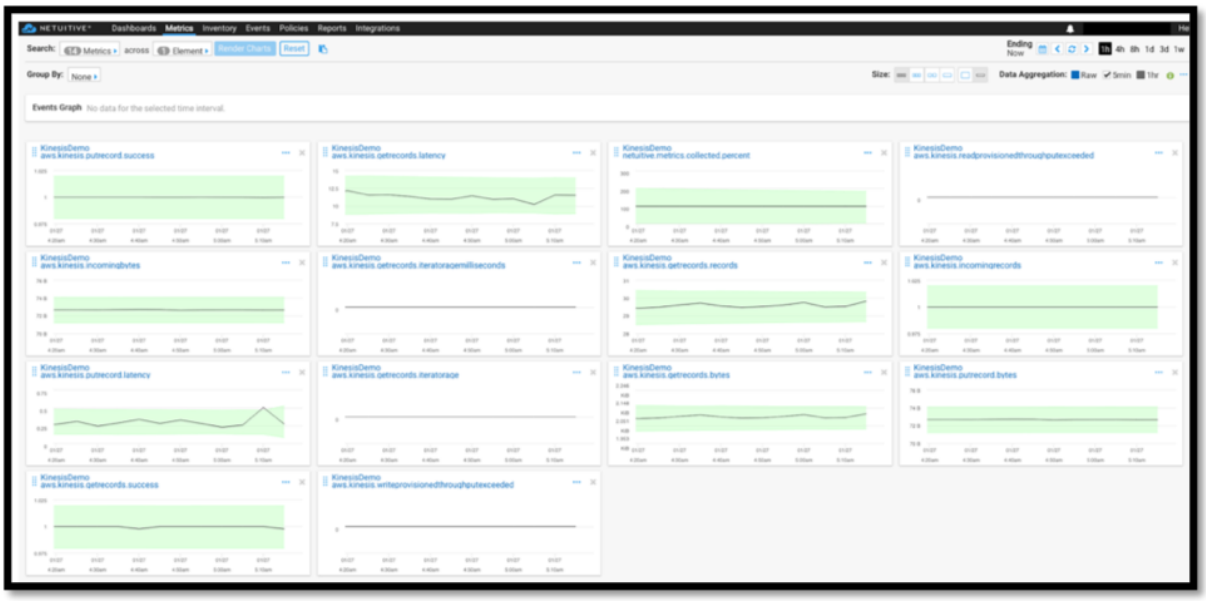
Amazon Kinesis

The AWS environment has different services that perform analytics on the user events, logs, and metrics captured by Citrix ADC. The data can be website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events.

- Amazon Kinesis Data Streams perform analytics in scenarios that involve scalable and durable real-time data streaming that can continuously capture GB of data per second from several sources.
- Amazon Kinesis Data Analytics can be used for scenarios with lower latency between the session generation because it takes less time to aggregate various data sets.
- Amazon Kinesis Agent for Microsoft Windows collects, parses, filters, and streams input data to Kinesis data streams.
- Once the data is up in the cloud, you can implement the exact data pipeline to get the results you want. For example, you can use this information in Amazon Quick Sight, which is a visualization tool that is used to build dashboards.

The AWS Kinesis dashboard provides the following offerings:

- Showcases web apps UI issues
- Near real-time visualizations of web usage metrics such as events per hour, visitor count, and referrers.
- Session-wise analysis



Citrix ADM Analytics

By utilizing Citrix ADM with Citrix ADC, you can get a single-pane-of-glass view across all the business environments. Citrix ADC captured logs are fed into Citrix ADM, that treats your individual applications as a single entity. You can gain valuable insights and effectively troubleshoot issues with the following ADM capabilities:

- Intelligent analytics
- Web transaction analytics
- Anomaly detection
- Performance and network-related issues

The following ADM service dashboard helps you gain valuable insights to effectively troubleshoot the issues.

How Citrix ADM correlates with Clickstream analytics

Clickstream analytics data can be correlated with ADM analytics to describe, predict, and improve application's performance.

For more information on Citrix ADM, see Citrix ADM

For example, an organization while analyzing their logs notice that most of the users are abandoning their sites. But to find the root cause behind this user behavior, they need to find out which part of their application is performing bad. With clickstream analytics data and ADM analytics, you can derive the following insights to analyze the reason behind users abandoning a site:

- Is the user abandoning due to latency, 5xx errors?
- Are there any SSL Handshake errors?
- Is there some part of the application that has performance or network related issues?
- Is there a 404 error, or the page loading time takes forever to respond, and so on.
- Are customers facing server response anomalies?

Citrix ADM service provides Web Insights that allow IT administrators to speed up solving issues with the following features:

- Provides integrated and real-time monitoring of all web applications that served by the Citrix

ADC.

- Get a holistic view on the application performance w.r.t. time, latency, and the usual user's behavior through observability tools (such as global service graph).
- Perform intelligent analytics to understand server response anomalies.
- SSL insights contribute towards resolving 5xx and 4xx errors.
- To maintain records of all web sessions that include:
  - Detailed logs of every web transaction
  - Search capability to find relevant logs
  - Ability to isolate an ADC-to-end user vs. ADC-to-server problem

Types of data exported by ADC for Clickstream analytics

Citrix ADC captures the different sources that generate varied forms of data, which are as follows:

- Web server logs

  Web server logging feature sends logs of HTTP and HTTPS requests to a client system for storage and retrieval. These logs contain huge amount of data, which is difficult to comprehend and make sense out of it. Analytical tools help in understanding and bring value from it. For configuration details, see the **Web logging configuration section** in this document.

- Syslogs

  The primary use of syslogs is for systems management. Proactive syslog monitoring pays off because it significantly reduces downtime of servers and other devices in your infrastructure. Syslog identifies critical network issues and reports them proactively.

- Access logs

  The access logs store information about events that occurred on your web server. For instance, when someone visits your website, a log is recorded and stored to provide the web server administrator with information such as the IP address of the visitor, what pages they were viewing, status codes, browser used. To access logs might be overwhelming, if there is lack of appropriate knowledge to understand them.
  You can program your system to integrate with:

  - Citrix ADC for seamless delivery
  - Kinesis for actionable insights that is useful for businesses

- Audit logs

  The Audit Logging feature enables you to log the Citrix ADC states and status information collected by various modules in the kernel and in the user-level daemons.

- Error logs

The error logs file is an aid for administrators to provide more information regarding a specific error that has occurred on the web server.

Configure the Citrix ADC solution for clickstream analytics

The Web server logging feature enables you to send logs of HTTP and HTTPS requests to a client system for storage and retrieval.

To configure the Citrix ADC for web server logging you must:

- Enable web logging feature
- Configure the size of the buffer to temporarily store the log entries because the Web log server runs on the Citrix ADC.

To configure web server logging by using CLI:

1. Enable the web server logging feature.

```
enable ns feature WL
<!--NeedCopy-->
```

2. [Optional] Modify/Configure the buffer size for storing the logged information.

```
set ns weblogparam -bufferSizeMB 60
<!--NeedCopy-->
```

3. Install the Citrix ADC web logging (NSWL) client. For more information, see Installing the Citrix ADC web logging (NSWL) client
4. Install the NSWL client on Windows by performing the following operations on the system where you downloaded the package.
   a) Extract and copy the nswl_win-< release number >-< build number >.zip file from the package to a Windows system on which you want to install the NSWL client.
   b) On the Windows system, unzip the file in a directory (referred as < NSWL-HOME>). Bin, samples, and other directories are extracted.
   c) At the command prompt, run the following command from the < NSWL-HOME >\bin directory:

```
nswl -install -f < path of the log.conf file >\log.conf
<!--NeedCopy-->
```

> **Note:**
>
> To uninstall the NSWL client, at the command prompt, run the following command from

the < NSWL-HOME >\bin directory:

```
nswl -remove
<!--NeedCopy-->
```

5. After you install the NSWL client, configure the NSWL client using the NSWL executable. These configurations are stored in the NSWL client configuration file (log.conf).

   Run the following commands from the directory in which the NSWL executable is located:

```
\ns\bin
<!--NeedCopy-->
```

6. In the NSWL client configuration file (log.conf), add the Citrix ADC IP address (NSIP) from which the NSWL client collects logs by running the following in the client system command prompt:

```
nswl -addns -f < Path to the configuration(log.conf) file >\log.
   conf
<!--NeedCopy-->
```

7. Input the Citrix ADC appliance's NSIP (IP address), user name as `nsroot` and password as "the instance id/your set password" so that:

   - NSWL client connects to the ADC after you add the NetScaler IP address (NSIP) to the NSWL configuration file
   - ADC buffers the HTTP and HTTPS request log entries before sending them to the client.
   - The client can filter the entries(by modifying log.conf file) before storing them.

Note

Change the default password for Citrix ADC and then proceed with the configuration. Type the following command to change the password:

```
set system user nsroot -password <your password>
<!--NeedCopy-->
```

Configuring the Amazon Kinesis agent

Perform the following steps in the AWS web console to configure the Amazon Kinesis agent:

1. Create a configuration file (appsettings.json) and deploy it. Configuration files define sets of sources, sinks, and pipes that connect sources to sinks, along with optional transformations.

   The following example is a complete `appsettings.json` configuration file that configures Kinesis Agent to stream Windows application log events to Kinesis Data Firehose.


```
{

"Sources": [
    {

        "Id": "NSWLog",
        "SourceType": "DirectorySource",
        "Directory": "C:\\Users\\Administrator\\Downloads\\
           nswl_win-13.0-52.24\\bin",
        "FileNameFilter": "*.log"
        "RecordParser": "TimeStamp",
        "TimestampFormat": "yyyy-MM-dddd HH:mm:ss.ffff", //
           Optional parameter required only by the timestamp
           record parser
        "TimeZoneKind": "UTC", //Local or UTC
        "SkipLines": 0 //Skip a number of lines at the beginning
           of each file
     }

    ],
"Sinks": [
    {

        "Id": "ApplicationLogKinesisFirehoseSink",
        "SinkType": "KinesisFirehose",
        "StreamName": "Delivery-ik-logs",
        "AccessKey": "Your Access Key",
        "SecretKey": "YourSecretKey",
        "Region": "ap-south-1"
     }

    ],
"Pipes": [
    {

    "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink
       ",
    "SourceRef": "ApplicationLogSource",
    "SinkRef": "ApplicationLogKinesisFirehoseSink"
     }

    ],
"Telemetrics":
    {
```



```
40
41      "off": "true"
42       }
43
44   }
45
46 <!--NeedCopy-->
```

2. Set up a Kinesis Agent on data sources to collect data and send it continuously to Amazon Kinesis Firehose/Kinesis Data Analytics. For more information, see Getting Started with Amazon Kinesis Agent for Microsoft Windows.
3. Create an end-to-end data delivery stream using Amazon Kinesis Firehose. The delivery stream transmits your data from the agent to the destination. The destination includes Amazon Kinesis Analytics, Amazon Redshift, Amazon Elasticsearch service, and Amazon S3. For the Source, choose **Direct PUT or other sources** to create a Kinesis Data Firehose delivery stream.
4. Process the incoming log data using SQL queries in Amazon Kinesis Analytics.
5. Load processed data from Kinesis Analytics to Amazon Elasticsearch Service to index the data.
6. Analyze and visualize the processed data using Visualization tools, such as Kibana and AWS QuickInsight Services.

References

- View and Export syslog messages
- Citrix Networking for Hybrid Multi Cloud
- Writing to AWK Kinesis Data Streams using Kinesis Agent

Citrix ADC in a Private Cloud Managed by Microsoft Windows Azure Pack and Cisco ACI

September 14, 2021

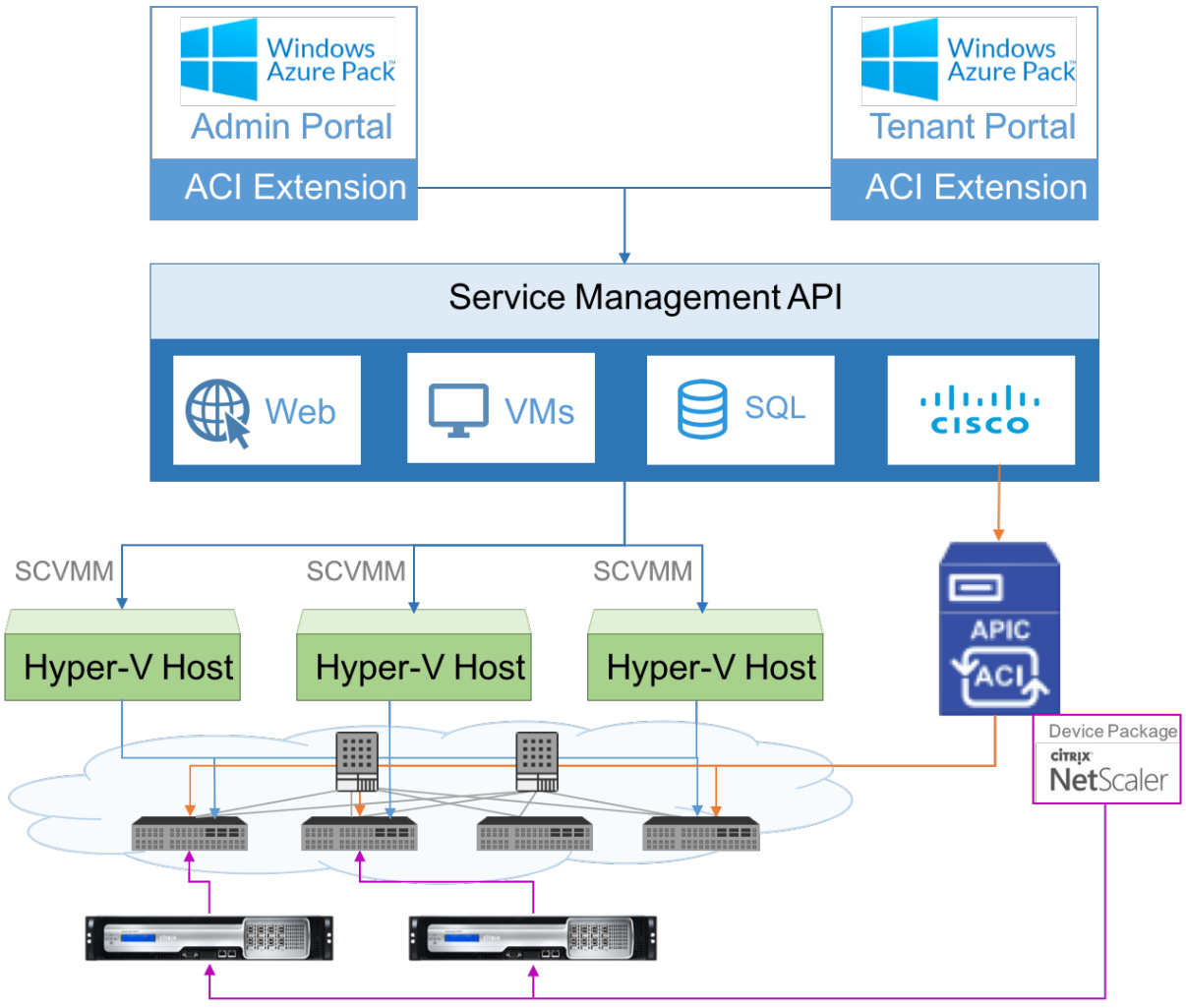
You can use a Citrix ADC appliance for load balancing in a private cloud that is managed through Microsoft Windows Azure Pack. The network for the private cloud is automated by using Cisco ACI and Citrix ADC.

This solution involves many integration points, such as Windows Azure Pack (WAP) to Cisco APIC, Cisco APIC to System Center Virtual Machine Manager (SCVMM), and Cisco APIC to Citrix ADC. As a tenant in the private cloud, you can enable NAT, provision network services, and add a load balancer.

WAP supports tenant and administrator portals where an administrator can perform administrative tasks such as ACI registration, VIP range, Citrix ADC device association with virtual machine cloud, tenant user account creation. Tenants can log on to the WAP Tenant Portal and configure the network, bridge domains, and Virtual Routing and Forwarding (VRFs), and make use of the Citrix ADC load balancing and RNAT features.

> **Important**
>
> - In this solution, the Citrix ADC appliance provides only basic load balancing.
> - Tenants can deploy multiple VIP addresses with different ports for the same network, but must ensure that the IP and port combination is unique.
> - The Citrix ADC device package supports only single-context deployment. Each Tenant gets a dedicated Citrix ADC instance.
> - WAP supports Citrix ADC MPX appliances and Citrix ADC VPX virtual appliances, including Citrix ADC VPX instances deployed on the Citrix ADC SDX platform.

The following illustration provides an overview of the solution:



Prerequisites

Make sure that:

- You have conceptual knowledge of Cisco ACI components and Citrix ADCs.
  - For more information about Cisco ACI and its components, see the product documentation at: http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html.
  - For more information about the Citrix ADCs, see the Citrix ADC product documentation at http://docs.citrix.com/.
- All the required components of Cisco ACI, including Cisco APIC in the data center, are set up and configured. For more information about Cisco ACI and its components, see the product documentation at: http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html.
- You know how to integrate Cisco ACI with Microsoft Windows Azure Pack. See the product documentation at: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/2-x/virtualization/b_ACI_Virtualization_Guide_2_2_1.html.
- You have conceptual knowledge of Microsoft Windows Azure Pack. See the product documentation at: https://www.microsoft.com/en-in/cloud-platform/windows-azure-pack.
- You have installed Citrix ADC software release 11.1 or later.
- You configure Citrix ADCs in Cisco ACI, so that they can be managed by using Cisco APIC.
- From Cisco APIC, make sure that:
  - Management connectivity of Cisco APIC to Citrix ADC is established.
  - You upload the Citrix ADC device package version *11.1–52.3* and register the Citrix ADC device in Cisco ACI by using Cisco APIC.
  - You configure the Citrix ADC appliance in Cisco APIC's common tenant and make sure that there are no faults in Cisco APIC.
  - You have configured all the APIC specific configurations such as, VLAN pool, L3OutServicesDom, L3ExtOUt, resource pool. For more information, see *Cisco documentation*.

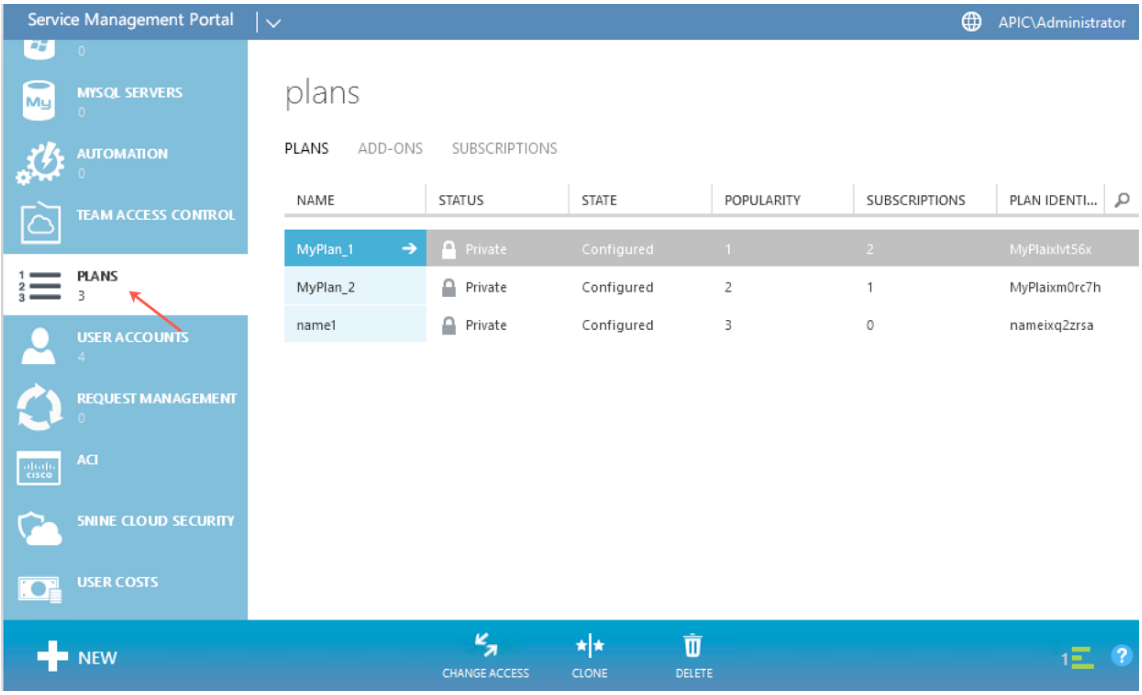
Creating a Citrix ADC Load Balancer in a Plan in the Service Management Portal (Admin Portal)

September 14, 2021

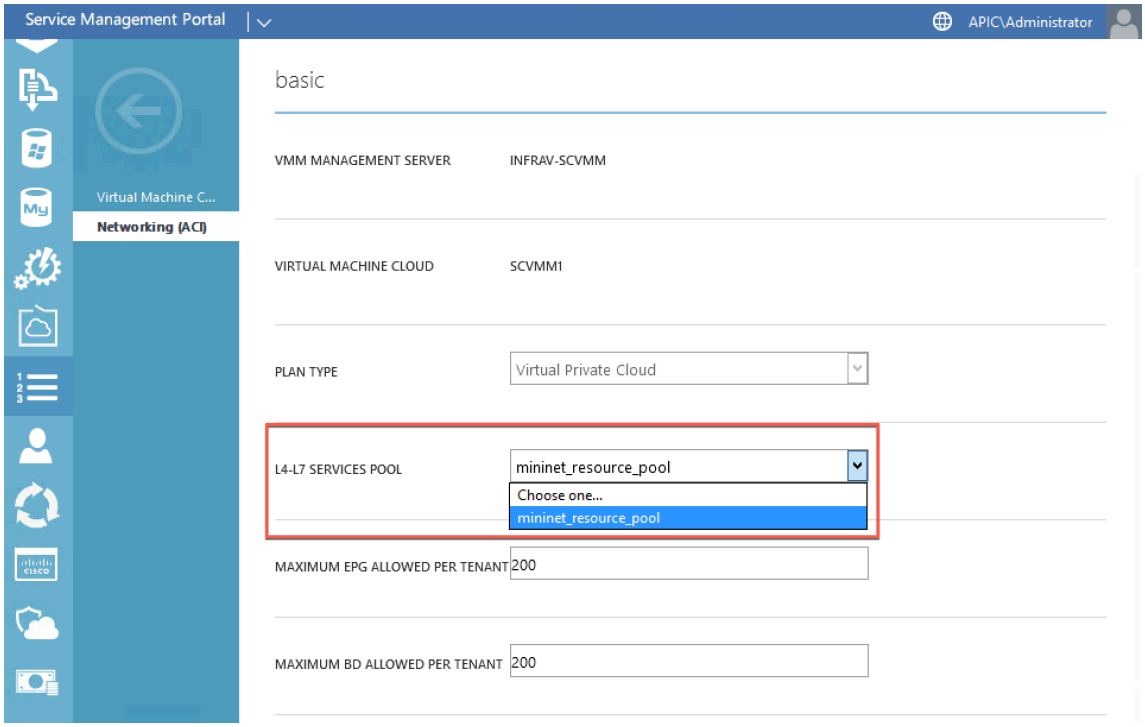
The Service Management Portal in WAP allows an administrator to register Cisco APIC with WAP and also create a hosting plan. As part of the plan, you can specify the VIP range, associate the Citrix ADC load balancer with the plan, and create tenant user accounts.

To create a Citrix ADC Load Balancer in a Plan in the Admin Portal:

1. Log in to the Service Management Portal (Admin Portal).
2. In the Navigation pane, select **PLANS**.



3. In the plans pane, select the plan that you want to add a load balancer.
4. In the selected plan's pane, select **Networking (ACI)**.
5. On the **Networking (ACI)** pane, in the **L4-L7 SERVICE POOL** drop-down list, select the L4-L7 resource pool that you had created in Cisco APIC.



6. Create a tenant user account and associate the user with the plan you have created.

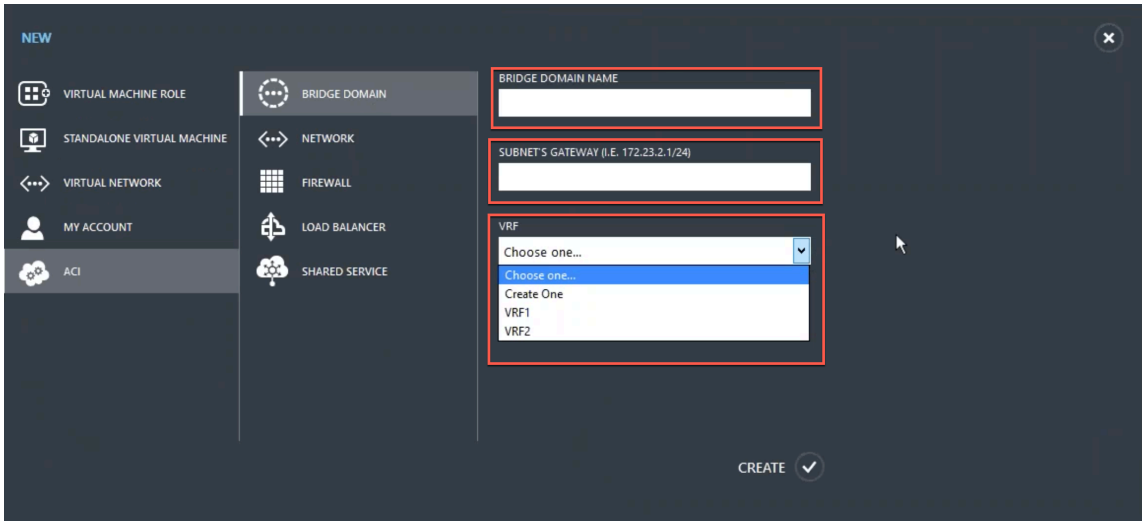
Configuring a Citrix ADC Load Balancer by Using the Service Management Portal (Tenant Portal)

September 14, 2021

In WAP, once the Tenant creates the Bridge Domain (BD), VRF, and a Network, the Tenant can configure a Citrix ADC Load Balancer through the Service Management Portal (Tenant Portal).

To configure Citrix ADC Load Balancer in Service Management Portal (Tenant Portal)

1. Log on to the Service Management Portal (Tenant Portal).
2. Create a bridge domain and VRF, as follows:
   a. In the navigation pane, select **ACI**.
   b. Click **NEW**.
   c. In the **NEW** pane, select **BRIDGE DOMAIN**.



d. In the **BRIDGE DOMAIN** field, enter the bridge domain name (for example, BD01).

e. (Optional) In the **SUBNET'S GATEWAY** field, enter the subnet's gateway (for example, 192.168.1.1/24).

f. In the **VRF** field, select a VRF that is already part of the subscription or select **Create One** to create a VRF.

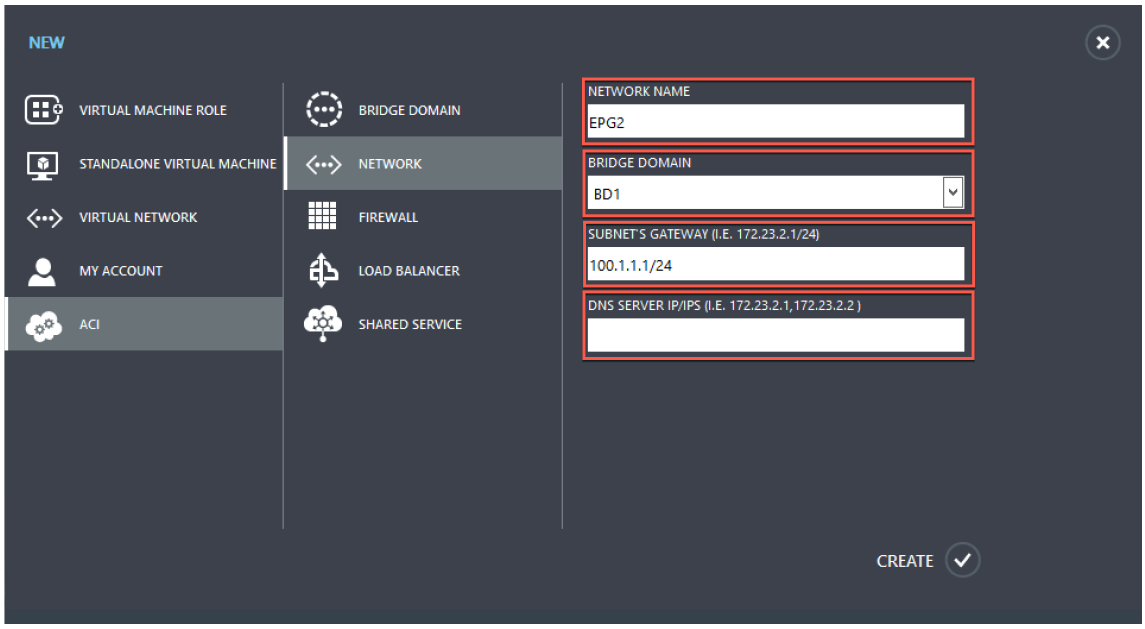
g. Click **CREATE**.

3. Create a network and associate it with the bridge domain that you created. Do the following:

a. In the navigation pane, select **ACI**.

b. Click **NEW**.

c. In the **NEW** pane, select **NETWORK**.



d. In the **NETWORK NAME** field, enter the network name (for example, S01).

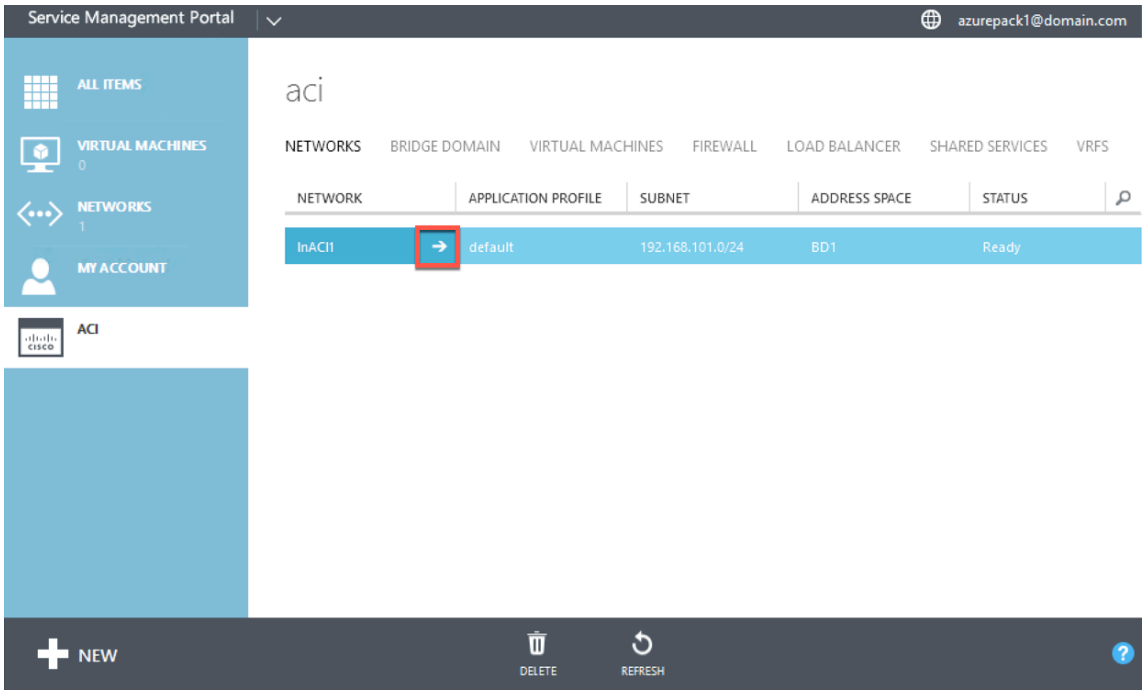
e. In the **BRIDGE DOMAIN** drop-down list, select the bridge domain you have created. (for example, BD01).

f. In the subnet's **GATEWAY** field, enter the subnet's gateway address (for example, 172.23.2.1/24).

g. (Optional) In the **DNS SERVER IP/IPS** field, enter the DNS server details.

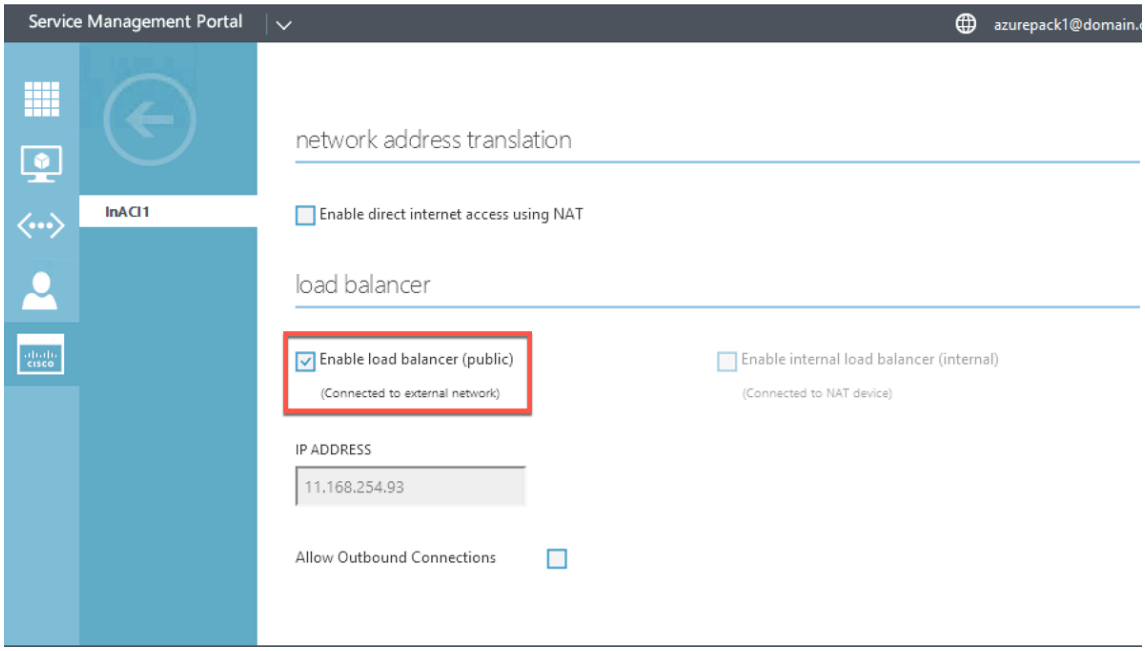
h. Click **CREATE**.

4. In the **ACI** pane, select **NETWORKS**.

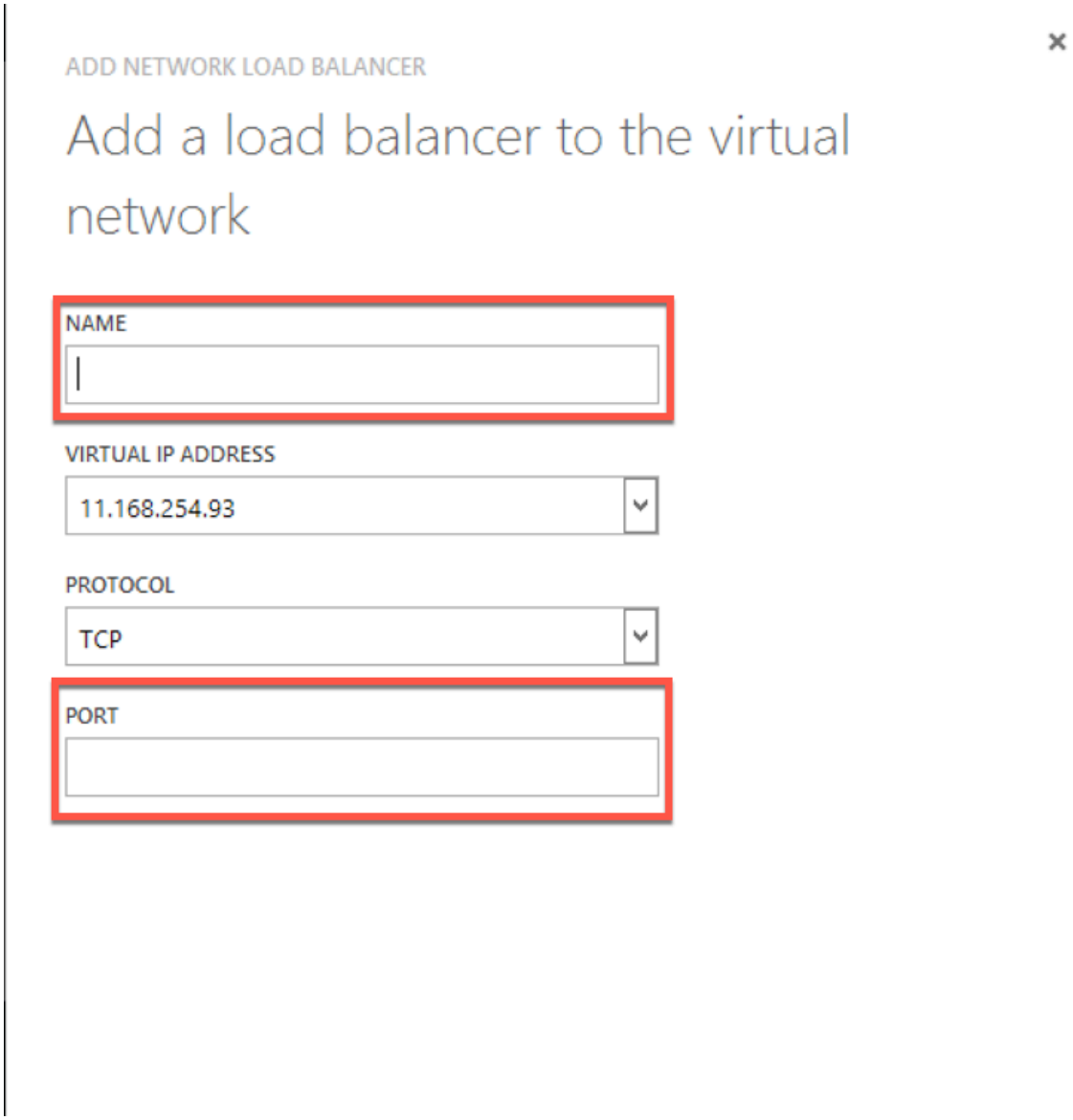


5. Double-click the network that you have created. Then, in the network pane, select **Enable load balancer (public)**. In the **IP ADDRESS** field, a VIP is automatically assigned from the VIP Range that the administrator configured in the Admin Portal. For more information, see Creating a Citrix ADC Load Balancer in a Plan in the Service Management Portal (Admin Portal).

6. Double-click the network that you have created. Then, in the network pane, select **Enable load balancer (public)**. In the **IP ADDRESS** field, a VIP is automatically assigned from the VIP Range that the administrator configured in the Admin Portal. For more information, see Creating a Citrix ADC Load Balancer in a Plan in the Service Management Portal (Admin Portal).



7. In the network pane, select the **Load Balancers** tab, and click **ADD**.



8. In the **ADD NETWORK LOAD BALANCER** pane, do the following:

   a. In the **NAME** field, enter the name for the load balancer.

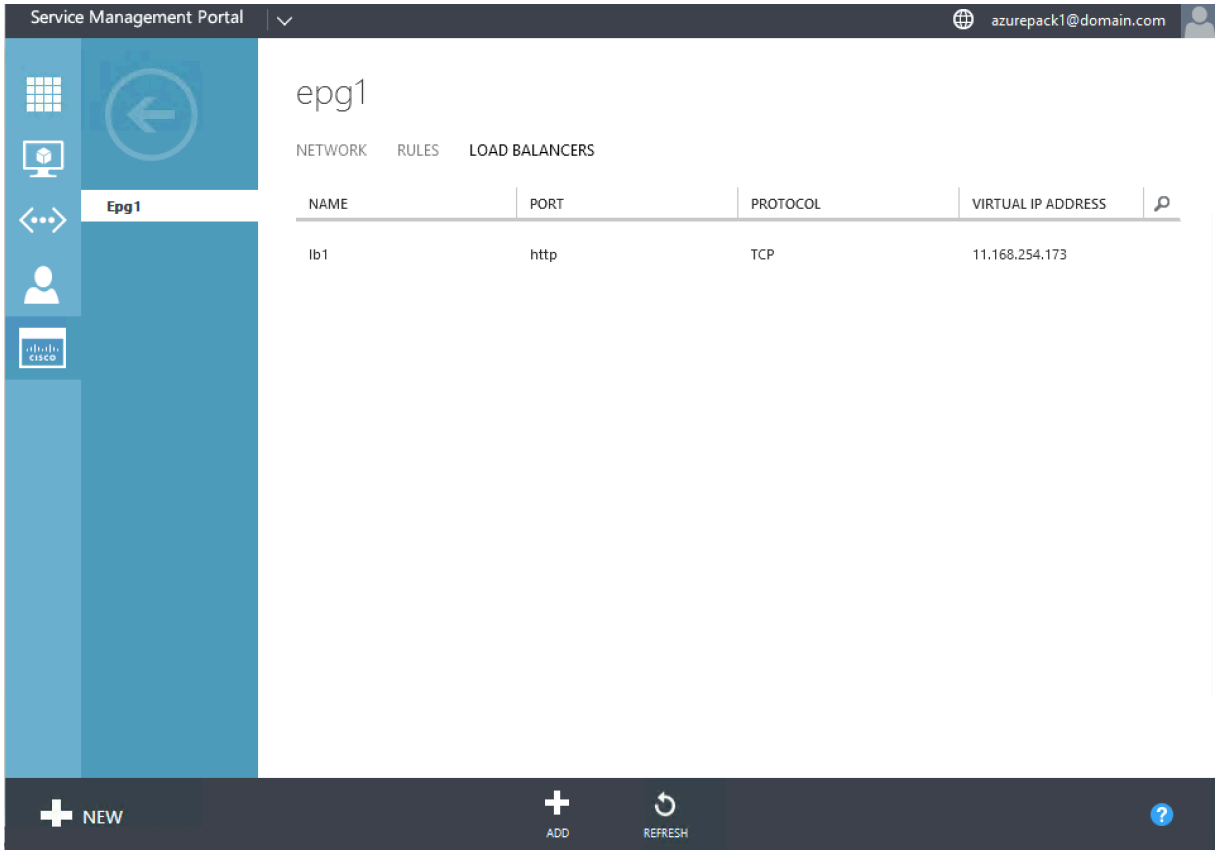
   b. Optionally, in the **VIRTUAL IP ADDRESS** field, assign the load balancer a VIP address from the VIP range that you defined earlier.

   c. Optionally, in the **PROTOCOL** field, select **TCP**.

   d. In the **PORT** field, enter the port number.

9. Click **CREATE**.

   The Citrix ADC Load Balancer is displayed in the **LOAD BALANCERS** tab and the Citrix ADC Load Balancer is data path ready.



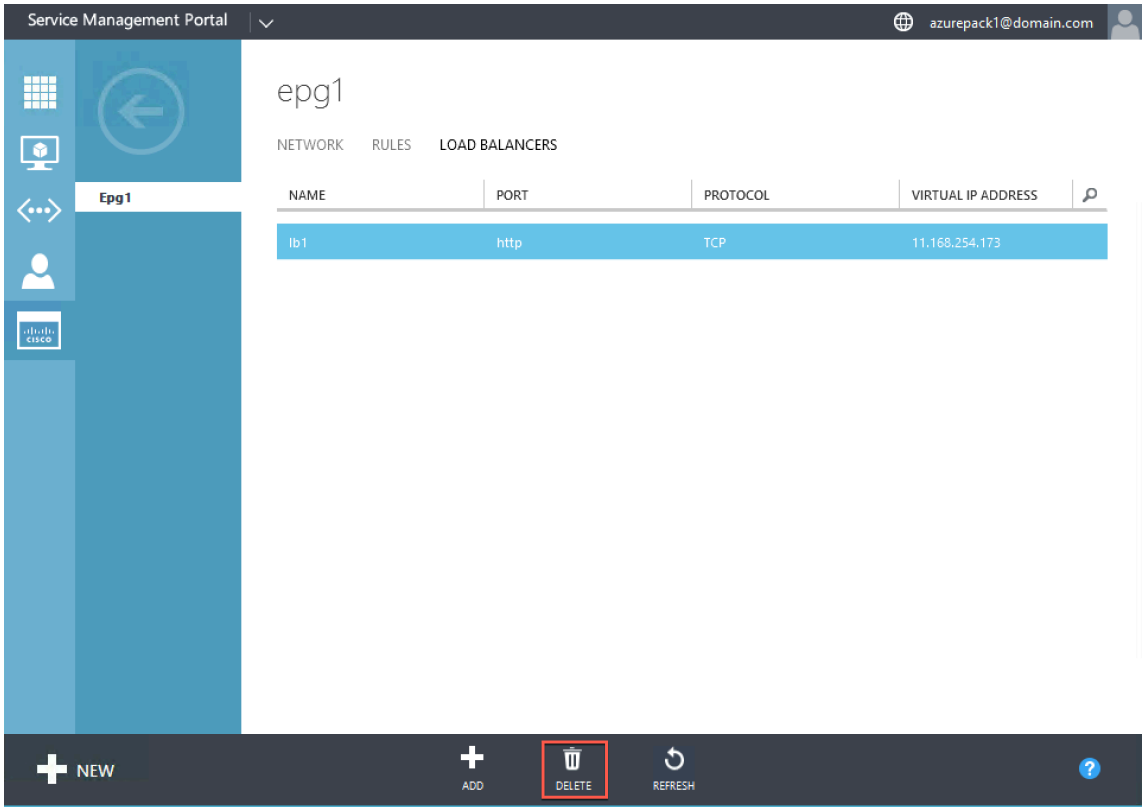
Deleting a Citrix ADC Load Balancer from the Network

September 14, 2021

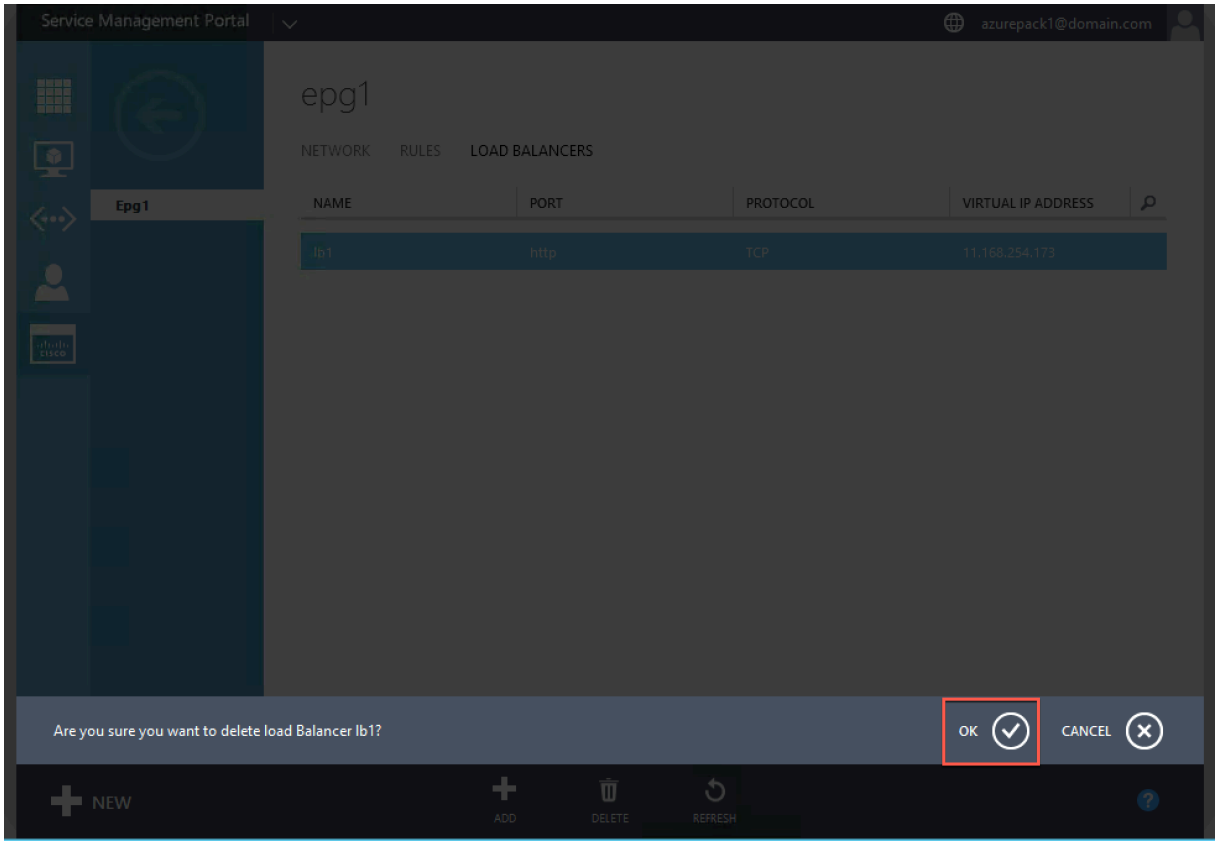
Using the Service Management Portal (Tenant Portal), from the Network, you can delete the Citrix ADC load balancer that you created.

To delete a Citrix ADC load balancer from the Network:

1. Log on to the Service Management Portal (Tenant Portal).
2. In the navigation pane, select **ACI**.
3. In the **ACI** pane, on the **NETWORKS** tab, click the network that you created.
4. In the selected network's pane, select the Citrix ADC load balancer and click **DELETE**.



5. Click **OK** to delete the Citrix ADC load balancer.



Citrix cloud native solution for microservices based on Kubernetes

September 14, 2021

As companies transform to innovate faster and get closer to customers, they are rearchitecting their internal process and breaking down boundaries within their organization. They are removing silos to pull together the right skill sets in the same team. One of the goals is to create and deliver software applications with speed, agility, and efficiency. In this regard, modern application architectures based on microservices are being adopted by a growing number of enterprises.
Using a microservices architecture, you can create applications as sets of loosely coupled services which can be deployed, updated, and scaled independently.

Cloud native is an approach that relies on the microservices architecture for building and deploying applications with the following key attributes:

- Deploys applications as loosely coupled microservices or containers
- Involves a very high degree of automation
- Implements agile DevOps processes and continuous delivery workflows
- Centers around APIs for interaction and collaboration

How does Kubernetes help in the cloud native journey?

To provide the desired levels of agility and stability, cloud native applications require high levels of infrastructure automation, security, networking, and monitoring. You need a container orchestration system that can efficiently manage containers at a large scale. Kubernetes has emerged as the most popular platform for container deployment and orchestration. Kubernetes abstracts the complex task of running, deploying, and managing containers from developers and operators and automatically schedules containers among a cluster of nodes. Kubernetes and the cloud native computing foundation (CNCF) ecosystem helps you to build a platform for cloud native solutions.

Some of the key benefits of using Kubernetes:

- Simplifies application deployment be it on-premises, hybrid, or public cloud infrastructure
- Accelerates application development and deployment
- Increases agility, flexibility, and scalability of applications

What is Citrix cloud native solution?

To maximize the benefits of using Kubernetes in production, you need to integrate Kubernetes with several tools, vendor-sourced, and open-source components. Ensuring production grade reliability and security for their cloud native application is a challenge faced by many organizations.

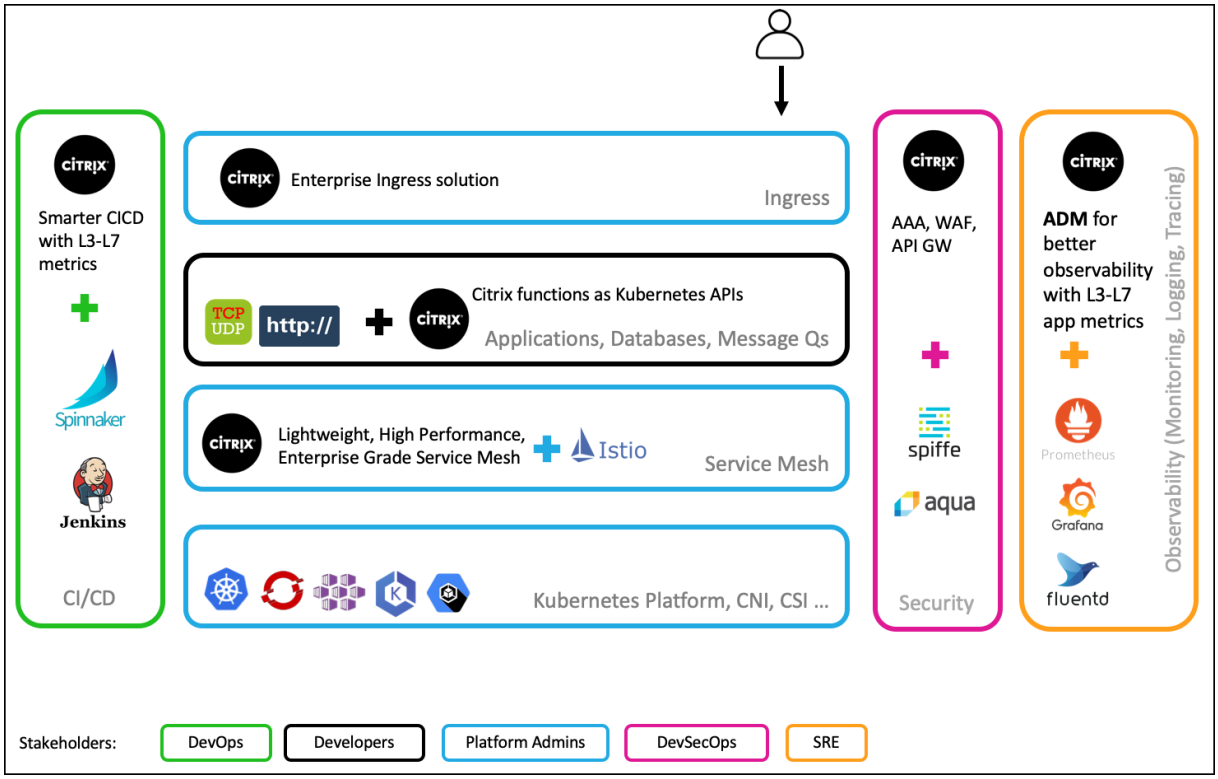
As the provider of industry-leading Citrix ADCs, Citrix offers a Citrix cloud native solution to address the challenges in a Kubernetes production environment.

Citrix cloud native solution leverages the advanced traffic management, observability, and comprehensive security features of Citrix ADCs to ensure enterprise grade reliability and security. It can provide complete visibility to application traffic in your Kubernetes environment, render immediate feedback, and help gain meaningful insights about the application performance.

The following table lists the key requirements of different stakeholders while implementing an Ingress solution.

Stakeholders	Job function	Needs
Platform administrators	Ensure availability of Kubernetes clusters	Simpler ways to manage applications deployed across multiple clusters, operation, and platform life cycle management
DevOps	Accelerate the deployment of applications to production	Integration with CI/CD pipeline, support for deployment techniques like Canary and blue-green for faster deployment
Developers	Develop and test microservices	Ways to bring traffic into the Kubernetes cluster, tracing and debugging, rate limiting for applications, and authentication for applications
SREs	Ensure availability of applications to meet service level agreements	Advanced telemetry for applications and infrastructure
SecOPs	Ensure security compliance	Secure Ingress traffic, API protection, service mesh for secure communication between microservices inside the Kubernetes cluster

The following diagram explains the Citrix cloud native solution and how it addresses the various challenges faced by stakeholders in their cloud native journey.



Citrix cloud native solution provides the following key benefits:

- Provides an advanced Kubernetes Ingress solution that caters to the needs of developers, SREs, devOps, and network or cluster administrators.
- Eliminates the need to rewrite legacy applications based on TCP or UDP traffic while moving them into a Kubernetes environment.
- Secures applications with Citrix ADC policies exposed as Kubernetes APIs.
- Helps to deploy high performing microservices for North-South traffic and East-West traffic.
- Provides an all-in-one view of all microservices using Citrix ADM service graph.
- Enables faster troubleshooting of microservices across different kinds of traffic including TCP, UDP, HTTP, HTTPS, and SSL.
- Secures APIs.
- Automates CI/CD pipeline for Canary deployments.
- Provides out of the box integrations with CNCF open-source tools.

For more information on different components of the Citrix cloud native solution, see the following links:

- Kubernetes Ingress solution
- Service mesh
- Solutions for observability
- API gateway for Kubernetes

Kubernetes Ingress solution

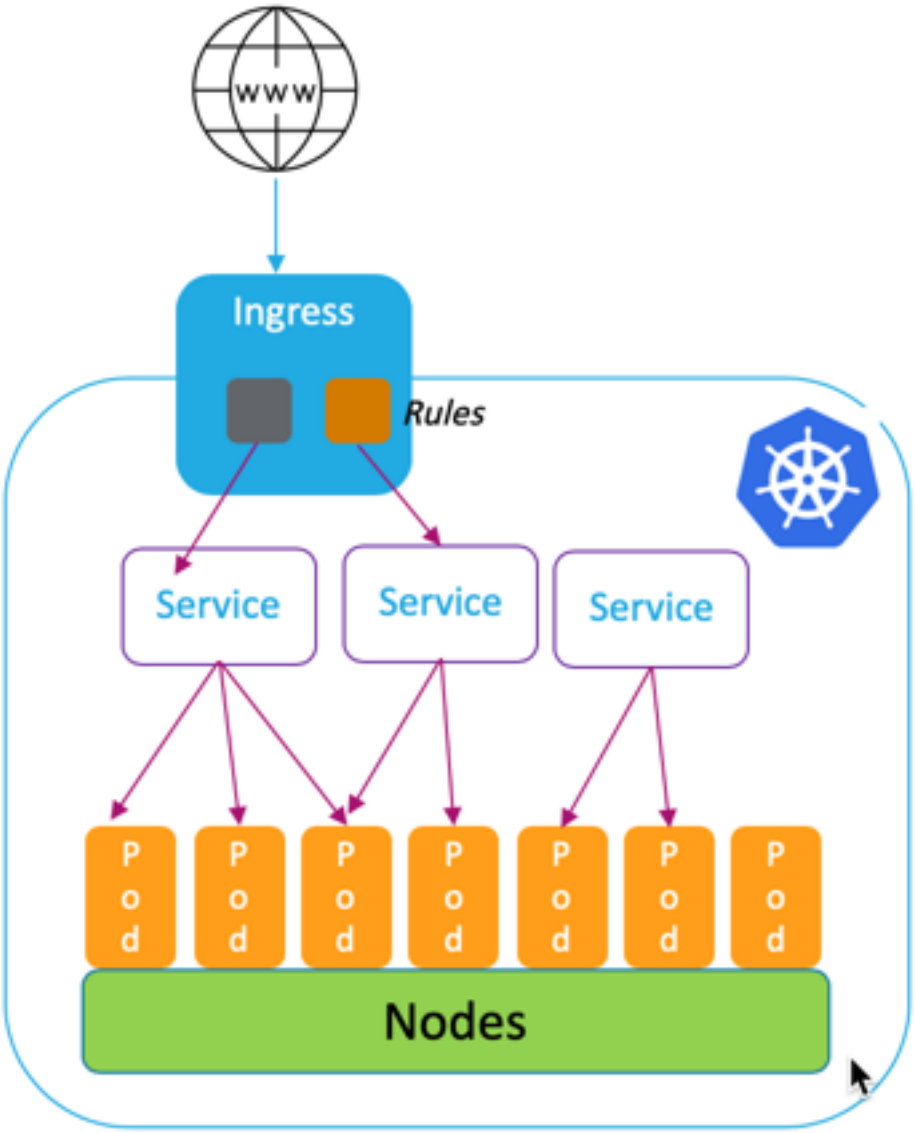
September 14, 2021

This topic provides an overview of the Kubernetes Ingress solution provided by Citrix and explains the benefits.

What is Kubernetes Ingress?

When you are running an application inside a Kubernetes cluster, you need to provide a way for external users to access the applications from outside the Kubernetes cluster. Kubernetes provides an object called Ingress that provides the most effective way to expose multiple services using a stable IP address. A Kubernetes ingress object is always associated with one or more services and acts as a single-entry point for external users to access services running inside the cluster.

The following diagram explains how Kubernetes Ingress works.



Kubernetes Ingress implementation consists of the following components:

- **Ingress resource**. An Ingress resource allows you to define rules for accessing the applications from outside of the cluster.
- **Ingress controller**. An Ingress controller is an application deployed inside the cluster that interprets rules defined in the Ingress. Ingress controller converts the Ingress rules into configuration instructions for a load balancing application integrated with the cluster. The load balancer can be a software application running inside your Kubernetes cluster or a hardware appliance running outside the cluster.
- **Ingress device**. An Ingress device is a load balancing application like Citrix ADC CPX, VPX, or

MPX which performs load balancing according to the configuration instructions provided by the Ingress controller.

What is the Kubernetes Ingress solution from Citrix?

In this solution, Citrix provides an implementation of Kubernetes Ingress controller to manage and route traffic into your Kubernetes cluster using Citrix ADCs (Citrix ADC CPX, VPX, or MPX). The Citrix ingress controller integrates Citrix ADCs with your Kubernetes environment and configures Citrix ADC CPX, VPX, or MPX according to the Ingress rules.

Standard Kubernetes Ingress solutions provide load balancing only at layer 7 (HTTP or HTTPS traffic). Some times, you need to expose many legacy applications which rely on TCP or UDP or applications and need a way to load balance those applications. Citrix Kubernetes Ingress solution provides TCP, TCP-SSL, and UDP traffic support apart from the standard HTTP or HTTPS Ingress. Also, it works seamlessly across multiple clouds or on-premises data centers.

Citrix ADC provides enterprise-grade traffic management policies like rewrite and responder policies for efficiently load balancing traffic at layer 7. However, Kubernetes Ingress lacks such enterprise-grade traffic management policies. With the Kubernetes Ingress solution from Citrix, you can apply rewrite and responder policies for application traffic in a Kubernetes environment using CRDs provided by Citrix.

The Kubernetes Ingress solution from Citrix also supports automated canary deployment for your CI/CD application pipeline. In this solution, Citrix ADC is integrated with the Spinnaker platform and acts as a source for providing accurate metrics for analyzing Canary deployment using Kayenta. After analyzing the metrics, Kayenta generates an aggregate score for the canary and decides to promote or fail the Canary version. You can also regulate traffic distribution to the Canary version using the Citrix ADC policy infrastructure.

The following table summarizes the benefits offered by the Ingress solution from Citrix over Kubernetes Ingress.

Features	Kubernetes Ingress	Ingress Solution from Citrix
HTTP and HTTPs support	Yes	Yes
URL routing	Yes	Yes
TLS	Yes	Yes
Load balancing	Yes	Yes
TCP, TCP-SSL	No	Yes
UDP	No	Yes
HTTP/2	Yes	Yes

Features	Kubernetes Ingress	Ingress Solution from Citrix
Automated canary deployment support with CI/CD tools	No	Yes
Support for applying Citrix ADC rewrite and responder policies	No	Yes
Authentication (Open Authorization (OAuth), mutual TLS (mTLS))	No	Yes
Support for applying Citrix Rate Limiting policies	No	Yes

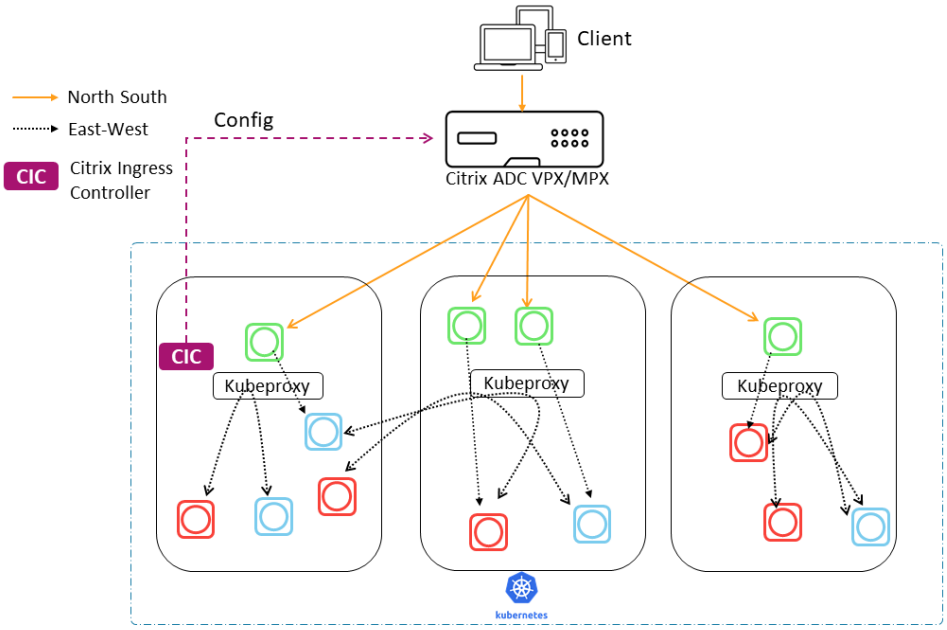
Deployment options for Kubernetes Ingress solution

Kubernetes Ingress solution from Citrix provides you flexible architecture depending on how you want to manage your Citrix ADCs and Kubernetes environment.

Unified Ingress (single-tier)

In a unified Ingress (single-tier) architecture, a Citrix MPX or VPX device deployed outside the Kubernetes cluster is integrated with the Kubernetes environment using the Citrix ingress controller. The Citrix ingress controller is deployed as a pod in the Kubernetes cluster and automates the configuration of Citrix ADCs based on changes to the microservices or the Ingress resources. The Citrix ADC device performs functions like load balancing, TLS termination, and HTTP or TCP protocol optimizations on inbound traffic and then routes the traffic to the correct microservice within a Kubernetes cluster. This architecture suits best in scenarios where the same team manages the Kubernetes platform and other networking infrastructure including application delivery controllers (ADCs).

The following diagram shows a deployment using the unified Ingress architecture.



A unified Ingress solution provides the following key benefits:

- Provides a way to extend the capabilities of your existing Citrix ADC infrastructure to the Kubernetes environment
- Enables you to apply traffic management policies for inbound traffic
- Provides a simplified architecture suitable for network-savvy DevOps teams
- Supports multitenancy

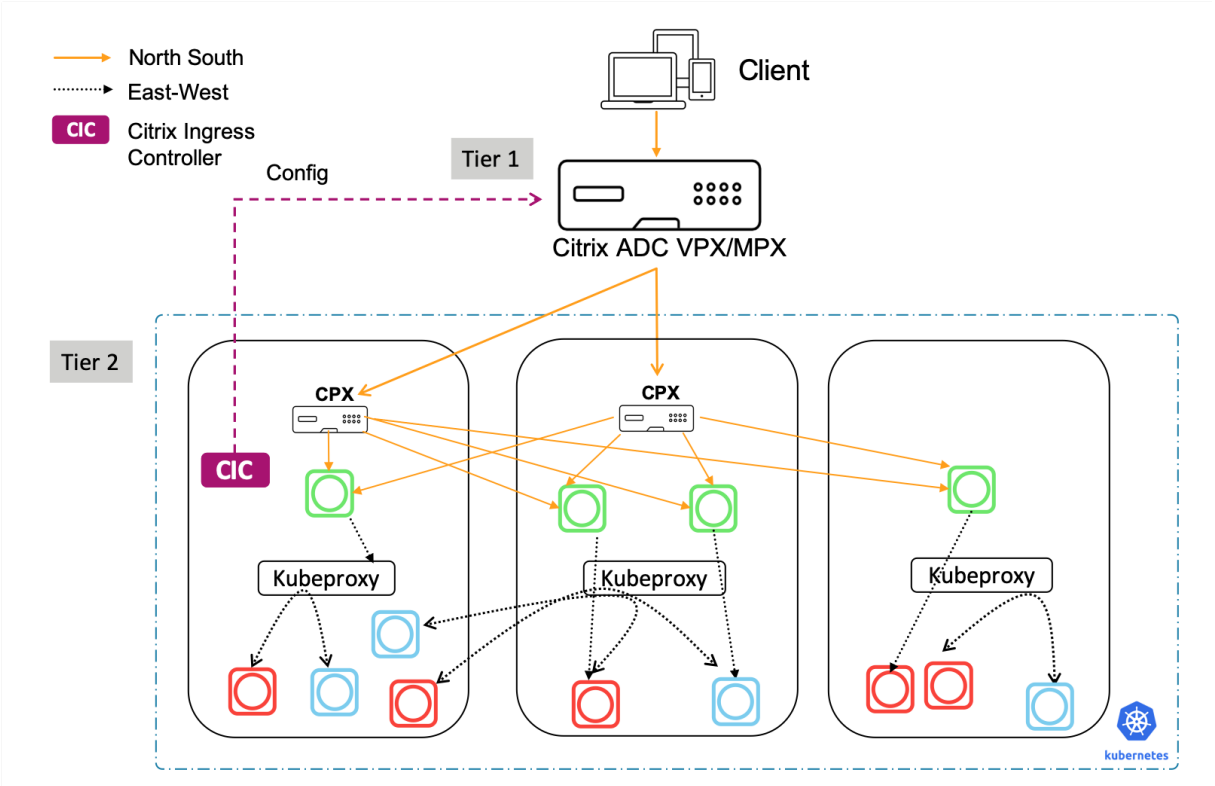
Dual-tier Ingress

In a dual-tier architecture, Citrix ADC (MPX or VPX) deployed outside the Kubernetes cluster acts at tier 1 and load balances North-South traffic to Citrix ADC CPXs running inside the cluster. Citrix ADC CPX acts at tier 2 and performs load balancing for microservices inside the Kubernetes cluster.

In scenarios where separate teams manage the Kubernetes platform and the network infrastructure, the dual-tier architecture is most suitable.

Networking teams use tier 1 Citrix ADC for use cases such as GSLB, TLS termination on the hardware platform, and TCP load balancing. Kubernetes platform teams can use tier 2 Citrix ADC (CPX) for Layer 7 (HTTP/HTTPS) load balancing, mutual TLS, and observability or monitoring of microservices. The tier 2 Citrix ADC (CPX) can have a different software release version than the tier 1 Citrix ADC to accommodate newly available capabilities.

The following diagram shows a deployment with dual-tier architecture.



A dual-tier Ingress provides the following key benefits:

- Ensures high velocity of application development for developers or platform teams
- Enables applying developer driven traffic management policies for microservices inside the Kubernetes cluster
- Enables cloud scale and multitenancy

For more information, see the Citrix ingress controller documentation.

Getting started

To get started with the Kubernetes Ingress solution from Citrix, you can try out the following examples:

- Load balance Ingress traffic with Citrix ADC CPX in Minikube
- Load balance North-South Ingress traffic using Citrix ADC CPX proxy
- Load balance East-West microservice traffic using Citrix ADC CPX proxy
- Deep dive on Kubernetes features with Citrix ADC CPX

Service mesh

September 14, 2021

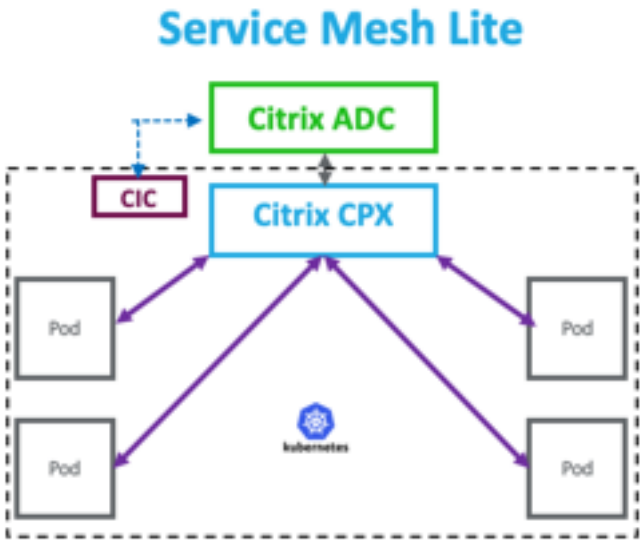
A service mesh is an infrastructure layer for handling service-to-service communication for cloud-native applications using APIs. It provides a way to connect, secure, and monitor your microservices. Citrix provides two solutions to meet your service mesh requirements:

- Service mesh lite
- Service mesh (Citrix ADC integration with Istio)

Service mesh lite

A full-fledged service-mesh implementation is complex and requires a steep learning curve. If you are looking for a simplified implementation of a service mesh with similar benefits, Citrix offers a solution called service mesh lite with lesser complexity. In this solution, a Citrix ADC CPX runs as a centralized load balancer in the Kubernetes cluster and load balances East-West traffic among microservices. Citrix ADC CPX enforces policies for inbound and inter-container traffic.

The following diagram shows a service mesh lite architecture.



For information, see the service mesh lite documentation.

Service mesh (Citrix ADC integration with Istio)

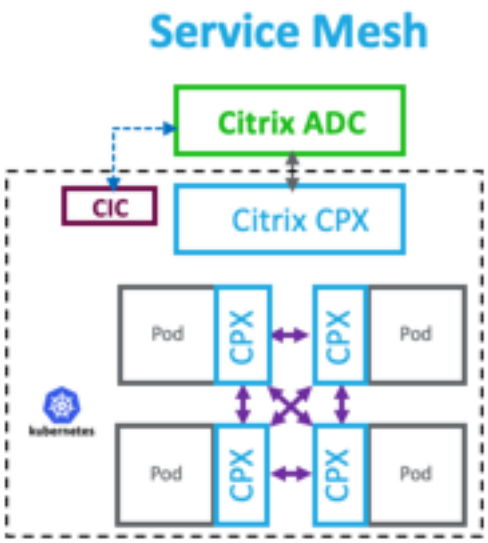
Citrix provides a service mesh solution by integrating Citrix ADC with Istio. Istio, an open source and platform-independent service mesh, is one of the most popular service mesh implementations. By integrating Citrix ADC with Istio, you can take advantage of the Citrix ADC features to secure and optimize the traffic for applications in the service mesh.

Citrix ADC can be integrated with Istio in the following ways:

- Citrix ADC MPX, VPX, or CPX as an Istio Ingress Gateway to the service mesh to expose traffic to the Kubernetes cluster.
- Citrix ADC CPX as a sidecar proxy with application containers in the service mesh to control communication between applications.

You can use either integration independently or you can combine both ways to have a unified data plane solution.

The following diagram shows a service mesh architecture.



Service mesh is ideal for highly secure applications and also offers the following benefits.

- Offers fine-grained (modularized) traffic management per container
- Ensures richer observability, analytics, and security (Mutual TLS) due to sidecar implementation
- Enables automated canary deployment for each container with embedded Citrix ADC CPX
- Supports cloud portability
- Allows offloading of some of the functions performed by applications to the sidecar
- Provides lower sidecar latency
- Provides integrations with open-source tools
- Offers scalability

For more information, see the Citrix ADC integration with Istio documentation.

Solutions for observability

September 14, 2021

In a microservices based architecture, visibility into service to service communications is critical to build an efficient and resilient architecture. Traditional ways for logging and monitoring is not capable of addressing the challenges of a microservices architecture. Observability solutions from Citrix provide you the ability to see what is happening when your services interact with each other and get meaningful insights about your system.

Citrix provides the following solutions to address the observability needs of your microservices architecture:

- Citrix ADM service graph and analytics
- Citrix ADC observability exporter

Citrix ADM service graph and analytics

Citrix Application Delivery Management (ADM) is a centralized management solution that provides enterprise-wide visibility and automation for management jobs that need to be run across multiple instances.

In a microservice architecture, troubleshooting is challenging because a single end-user request may span across multiple microservices.

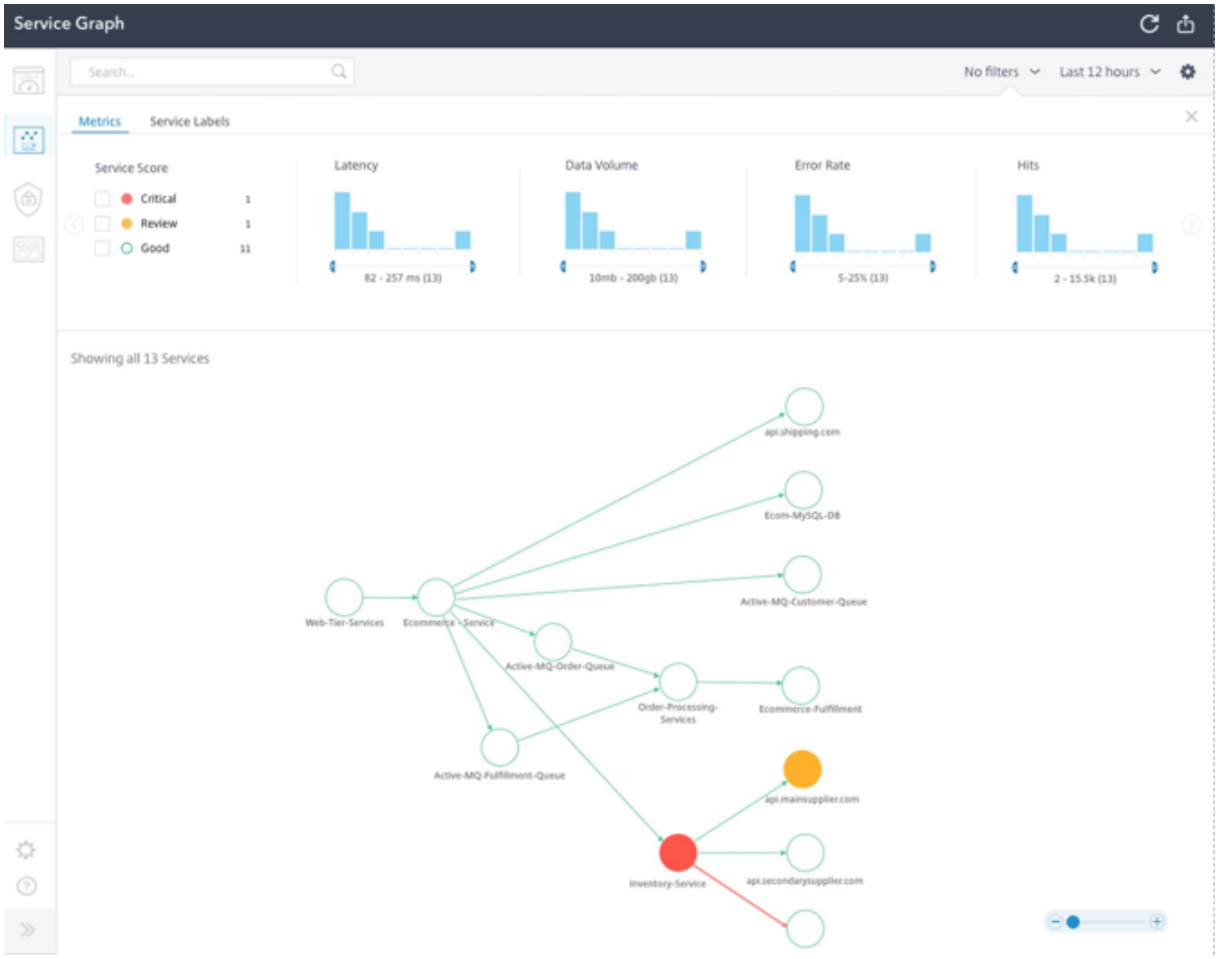
Citrix ADM's service graph and analytics provides visibility into interactions between microservices and helps to identify and fix issues based on various metrics such as latency and HTTP errors.

Citrix ADM also provides advanced analytics based on metrics and transaction logs collected from Citrix ADC.

Citrix ADM solution provides the following benefits:

- Provides a single pane of glass for applications across containers, on-premises, or cloud
- Provides better observability and faster troubleshooting for microservices
- Supports Canary deployment

The following diagram shows a sample service graph for an application which contains multiple microservices.



For more information on how to set up Citrix ADM service graph and analytics, see the Service graph and the Analytics documentation.

Citrix ADC observability exporter

Citrix ADC observability exporter is a container which collects metrics and transactions from Citrix ADCs and transforms them to suitable formats (such as JSON, AVRO) for supported endpoints. You can export the data collected by Citrix ADC observability exporter to the desired endpoint. By analyzing the data, you can get valuable insights at a microservices level for applications proxied by Citrix ADCs.

Distributed tracing support

Distributed tracers allow you to visualize the data flow between your microservices and helps to identify the bottlenecks in your microservices architecture. OpenTracing is a specification and standard set of APIs for designing and implementing distributed tracing.

Citrix observability exporter implements distributed tracing for Citrix ADC and currently supports Zipkin as the distributed tracer.

You can enhance the trace analysis by using Elasticsearch and Kibana with Zipkin. Elasticsearch provides long-term retention of the trace data. Kibana allows you to get much deeper insight into the data by providing a tool to explore, and visualize log messages.

Transaction collection and streaming support

Citrix ADC observability exporter supports collecting transactions and streaming them to endpoints. Currently, Citrix ADC observability exporter supports Elasticsearch and Kafka as transaction endpoints.

For more information, see the Citrix ADC observability exporter documentation.

Enable analytics using annotations in the Citrix ingress controller YAML file

You can enable analytics using the analytics profile which is defined as a smart annotation in Ingress or service of type LoadBalancer configuration. You can define the specific parameters you need to monitor by specifying them in the Ingress or service configuration of the application. For more information about enabling analytics using annotations, see Analytics using annotations.

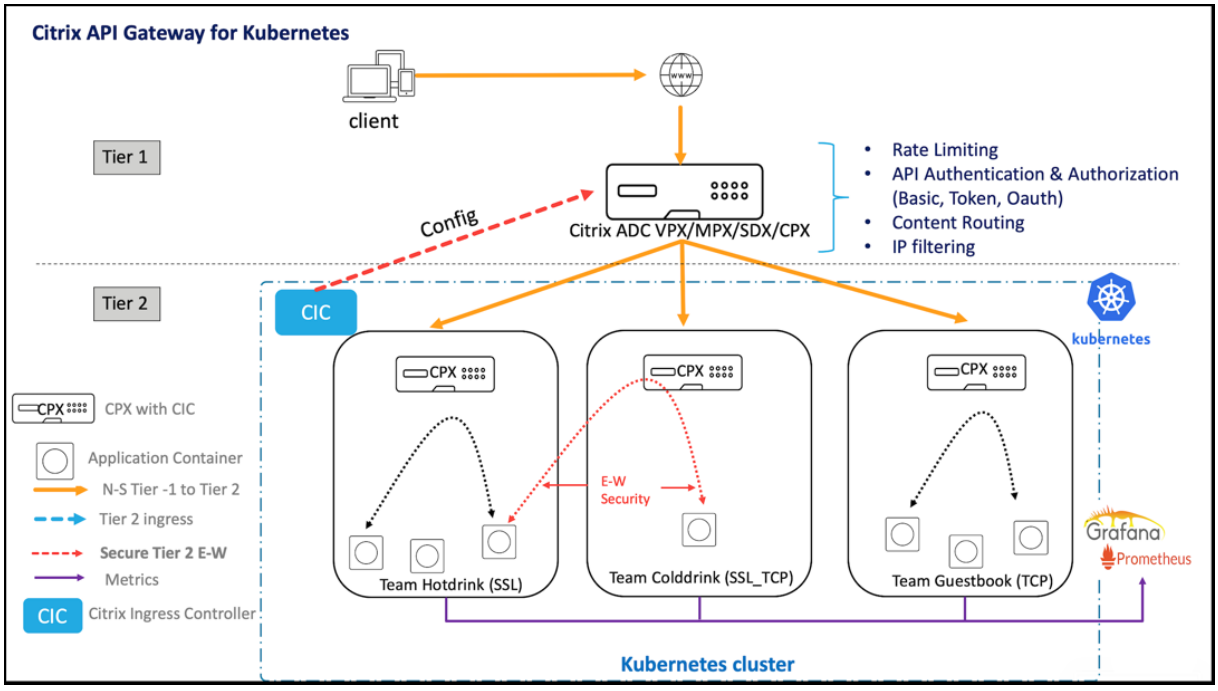
API gateway for Kubernetes

September 14, 2021

An API gateway acts as the single entry point for your APIs and ensures secure and reliable access to multiple APIs and microservices in your system.

Citrix provides an enterprise grade API gateway for North-South API traffic into the Kubernetes cluster. The
API gateway integrates with Kubernetes through the Citrix ingress controller and the Citrix ADC (Citrix ADC MPX, VPX, or CPX) deployed as the Ingress Gateway for on-premises or cloud deployments.

The following diagram shows a dual-tier topology for the API gateway.



Using the API gateway offered by Citrix, you can perform the following functionalities:

- Enforce authentication policies
- Rate limit access to services
- Advanced content routing
- Flexible and comprehensive transformation of HTTP transactions using the rewrite and responder policies
- Enforce web application firewall policies

How does the API gateway work

The API gateway is built on top of the Citrix ingress gateway and uses Kubernetes API extensions such as custom resource definitions (CRDs). Using CRDs, you can automatically configure the Citrix ADC and API gateway in the same instance.

Citrix provides the following CRDs for the API gateway:

- Auth CRD
- Rate limit CRD
- Content routing CRD
- Rewrite and responder CRD
- WAF CRD

Key benefits of using the API gateway

Following are the key benefits of the API gateway offered by Citrix:

- Uses the advanced traffic management and comprehensive security features of Citrix ADC.
- Optimizes your deployments by consolidating multiple network functions into a single component of the Citrix ingress gateway.
- Reduces the operational complexity and cost involved in deploying multiple components.
- Ensures better performance for your application traffic by reducing multiple hops of TCP or TLS decryption while using separate components.
- Simplifies deploy and integrate in your Kubernetes environments either by directly using YAMLs or helm charts.

Deploying the API gateway

For more information on how to configure the API gateway features using CRDs, see the Citrix ingress controller documentation:

- Authentication
- Rate limiting
- Advanced content routing
- Rewrite and responder policies
- Web application firewall policies

Deploy a Citrix ADC VPX instance

September 14, 2021

> **Note**
>
> Citrix ADM service connect is enabled by default, after you install or upgrade Citrix ADC or Citrix Gateway to release 13.0 build 61.xx and above. For more information see, Data governance and Citrix ADM service connect.

The Citrix ADC VPX product is a virtual appliance that can be hosted on a wide variety of virtualization and cloud platforms:

- Citrix Hypervisor
- VMware ESX
- Microsoft Hyper-V
- Linux KVM
- Amazon Web Services
- Microsoft Azure

- Google Cloud Platform

For more information, see the Citrix ADC VPX data sheet.

For more information about provisioning a Citrix ADC VPX instance on an SDX appliance, see Provisioning Citrix ADC instances.

Citrix Application Delivery Management for Citrix ADC VPX

Citrix Application Delivery Management software is a centralized management solution that simplifies operations by providing administrators with enterprise-wide visibility and automating management jobs that need to be run across multiple instances.

You can manage and monitor Citrix ADC VPX instances in addition to other Citrix application networking products such as Citrix Gateway, Citrix ADC SDX, Citrix ADC CPX, and Citrix SD-WAN. You can use the Application Delivery Management software to manage, monitor, and troubleshoot the entire global application delivery infrastructure from a single, unified console.

For more information, see Citrix Application Delivery Management documentation.

Support matrix and usage guidelines

October 7, 2021

This document lists the different hypervisors, and features supported on a Citrix ADC VPX instance, their usage guidelines, and known limitations.

Table 1. VPX instance on Citrix Hypervisor

Citrix Hypervisor version	SysID	VPX models
8.2 supported 13.0 64.x onwards, 8.0, 7.6, 7.1	450000	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G

Table 2. VPX instance on VMware ESXi server

ESXi version	ESXi release date in MM/DD/YYYY format	ESXi build number	Citrix ADC VPX version	SysID	VPX models
ESXi 7.0 update 2a	12/17/2020	17867351	13.0-82.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 update 1d	12/17/2020	17551050	13.0-82.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 update 1c	12/17/2020	17325551	13.0-82.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi version	ESXi release date in MM/DD/YYYY format	ESXi build number	Citrix ADC VPX version	SysID	VPX models
ESXi 7.0 update 1b	10/06/2020	16850804	13.0-76.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0b	06/23/2020	16324942	13.0-71.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 7.0 GA	04/02/2020	15843807	13.0-71.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi version	ESXi release date in MM/DD/YYYY format	ESXi build number	Citrix ADC VPX version	SysID	VPX models
ESXi 6.7 P04	11/19/2020	17167734	13.0-67.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 P03	08/20/2020	16713306	13.0-67.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 P02	04/28/2020	16075168	13.0-67.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi version	ESXi release date in MM/DD/YYYY format	ESXi build number	Citrix ADC VPX version	SysID	VPX models
ESXi 6.7 P01	12/05/2019	15160138	13.0-67.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 Update 3	08/20/2019	14320388	13.0-58.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.7 U2	04/11/2019	13006603	13.0-47.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi version	ESXi release date in MM/DD/YYYY format	ESXi build number	Citrix ADC VPX version	SysID	VPX models
ESXi 6.5 GA	11/15/2016	4564106	13.0-47.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.5 U1g	3/20/2018	7967591	13.0 47.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G
ESXi 6.0 Update 3	2/24/2017	5050593	12.0-51.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

ESXi version	ESXi release date in MM/DD/YYYY format	ESXi build number	Citrix ADC VPX version	SysID	VPX models
ESXi 6.0 Express Patch 11	10/5/2017	6765062	12.0-56.x onwards	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

Table 3. VPX on Microsoft Hyper-V

Hyper-V version	SysID	VPX models
2012, 2012 R2, 2016, 2019	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000

Table 4. VPX instance on generic KVM

Generic KVM version	SysID	VPX models
RHEL 7.4, RHEL 7.5 (from Citrix ADC version 12.1 50.x onwards), RHEL 7.6, RHEL 8.2, Ubuntu 16.04, Ubuntu 18.04, RHV 4.2	450070	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G. VPX 25G, VPX 40G, VPX 100G

Points to note:

Consider the following points while using KVM hypervisors.

- The VPX instance is qualified for hypervisor release versions mentioned in table 1–4, and not for patch releases within a version. However, the VPX instance is expected to work seamlessly with patch releases of a supported version. If it does not, log a support case for troubleshooting and

debugging.

- Use the `ip link` commands to configure RHEL 8.2 network bridges.
- Before using RHEL 7.6, complete the following steps on the KVM host:
  1. Edit /etc/default/grub and append `"kvm_intel.preemption_timer=0"` to `GRUB_CMDLINE_LINUX` variable.
  2. Regenerate grub.cfg with the command `"## grub2-mkconfig -o /boot/grub2/grub.cfg"`.
  3. Restart the host machine.
- Before using Ubuntu 18.04, complete the following steps on the KVM host:
  1. Edit /etc/default/grub and append `"kvm_intel.preemption_timer=0"` to `GRUB_CMDLINE_LINUX` variable.
  2. Regenerate grub.cfg with the command `"## grub-mkconfig -o /boot/grub/grub.cfg "`.
  3. Restart the host machine.

Table 5. VPX instance on AWS

AWS version	SysID	VPX models
N/A	450040	VPX 10, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX BYOL(VPX 8000, VPX 10G, VPX 15G, and VPX 25G are available only with BYOL with EC2 instance types (C5, M5, and C5n)

> Note:
> The VPX 25G offering doesn't give the desired 25G throughput in AWS but can give higher SSL transactions rate compared to VPX 15G offering.

Table 6. VPX instance on Azure

Azure version	SysID	VPX models
N/A	450020	VPX 10, VPX 200, VPX 1000, VPX 3000, VPX BYOL

Table 7. VPX feature matrix

	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM			VPX on AWS	VPX on Azure
Features	PV	SR-IOV	PV	SR-IOV	Emulated	PCI Passthrough	PV	PV	SR-IOV	PCI Passthrough		
Multi-PE Support	√	√	√	√	√	√	√	√	√	√	√	√
Clustering Support	√	*√	√	*√	√	√	√	√	*√	√	X	X
VLAN Tagging	√	√	√	√	√	√	√ ((Only on 2012R2))	√	√	√	X	X
Detecting Link Events	**X	***√	**X	***√	**X	***√	**X	**X	***√	***√	**X	**X
Interface Parameter Configuration	X	X	X	X	X	√	X	X	X	√	X	X
Static LA	**√	***√	**√	X	**√	***√	**√	**√	***√	***√	X	X
LACP	X	***√	**√	X	**√	***√	X	**√	***√	***√	X	X
Static CLAG	X	X	X	X	X	X	X	X	X	X	X	X
LACP CLAG	X	X	**√	X	**√	***√	X	**√	***√	***√	X	X
Hot-Plug	X	X	X	X	X	X	X	X	X	X	√	X

1. Clustering support is available on SRIOV for client and server facing interfaces and not for the backplane.
2. Interface DOWN events are not recorded in Citrix ADC VPX instances.
3. For Static LA, traffic might still be sent on the interface whose physical status is DOWN.
4. For LACP, the peer device knows the interface DOWN event based on the LACP timeout mechanism.
    - Short timeout: 3 seconds
    - Long timeout: 90 seconds
5. For LACP, do not share interfaces across VMs.
6. For Dynamic routing, convergence time depends on the Routing Protocol since link events are not detected.
7. Monitored static Route functionality fails if you do not bind monitors to static routes because the Route state depends on the VLAN status. The VLAN status depends on the link status.
8. Partial failure detection does not happen in high availability if there's link failure. High availability-split brain condition might happen if there's link failure.
    - When any link event (disable/enable, reset) is generated from a VPX instance, the physical status of the link does not change. For static LA, any traffic initiated by the peer gets dropped on the instance.

- For the VLAN tagging feature to work, do the following:

On the VMware ESX, set the port group's VLAN ID to 1–4095 on the vSwitch of the VMware ESX server. For more information about setting a VLAN ID on the vSwitch of the VMware ESX server, see VMware ESX Server 3 802.1Q VLAN Solutions.

Table 8. Supported browsers

Operating system	Browser and versions
Windows 7	Internet Explorer- 8, 9, 10, and 11; Mozilla Firefox 3.6.25 and above; Google Chrome- 15 and above
Windows 64 bit	Internet Explorer - 8, 9; Google Chrome - 15 and above
MAC	Mozilla Firefox - 12 and above; Safari - 5.1.3; Google Chrome - 15 and above

Usage guidelines

Follow these usage guidelines:

See the **VMware ESXi CPU Considerations** section in the document Performance Best Practices for VMware vSphere 6.5. Here's an extract:

- It is not recommended that virtual machines with high CPU/Memory demand sit on a Host/Cluster that is overcommitted.
- In most environments, ESXi allows significant levels of CPU overcommitment without impacting virtual machine performance. On a host, you can run more vCPUs than the total number of physical processor cores in that host.
- If an ESXi host becomes CPU saturated, that is, the virtual machines and other loads on the host demand all the CPU resources the host has, latency-sensitive workloads might not perform well. In this case you might want to reduce the CPU load, for example by powering off some virtual machines or migrating them to a different host (or allowing DRS to migrate them automatically).
- Citrix recommends the latest hardware compatibility version to avail latest feature sets of the ESXi hypervisor for the virtual machine. For more information about the hardware and ESXi version compatibility, see VMware documentation.
- The Citrix ADC VPX is a latency-sensitive, high-performance virtual appliance. To deliver its expected performance, the appliance requires vCPU reservation, memory reservation, vCPU pinning on the host. Also, hyper threading must be disabled on the host. If the host does not meet

these requirements, issues such as high-availability failover, CPU spike within the VPX instance, sluggishness in accessing the VPX CLI, pit boss daemon crash, packet drops, and low throughput occur.

A hypervisor is considered over-provisioned if one of the following two conditions is met:

- The total number of virtual cores (vCPU) provisioned on the host is greater than the total number of physical cores (pCPUs).
- The total number of provisioned VMs consume more vCPUs than the total number of pCPUs.

  If an instance is over-provisioned, the hypervisor might not guarantee the resources reserved (such as CPU, memory, and others) for the instance due to hypervisor scheduling over-heads, bugs, or limitations with the hypervisor. This behavior can cause lack of CPU resource for Citrix ADC and might lead to the issues mentioned in the first point under **Usage guidelines**. As administrators, you're recommended to reduce the tenancy on the host so that the total number of vCPUs provisioned on the host is lesser or equal to the total number of pCPUs.

  Example

  For ESX hypervisor, if the `%RDY%` parameter of a VPX vCPU is greater than 0 in the `esxtop` command output, the ESX host is said to be having scheduling overheads, which can cause latency related issues for the VPX instance.

  In such a situation, reduce the tenancy on the host so that `%RDY%` returns to 0 always. Alternatively, contact the hypervisor vendor to triage the reason for not honoring the resource reservation done.

- Hot adding is supported only for PV and SRIOV interfaces with Citrix ADC on AWS. VPX instances with ENA interfaces do not support hot-plug, and the behavior of the instances can be unpredictable if hot-plugging is attempted.
- Hot removing either through the AWS Web console or AWS CLI interface is not supported with PV, SRIOV, and ENA interfaces for Citrix ADC. The behavior of the instances can be unpredictable if hot-removal is attempted.

Commands to control the packet engine CPU usage

You can use two commands (`set ns vpxparam` and `show ns vpxparam`) to control the packet engine (non-management) CPU usage behavior of VPX instances in hypervisor and cloud environments:

- `set ns vpxparam [-cpuyield (YES | NO | DEFAULT)] [-masterclockcpu1 (YES | NO)]`

  Allow each VM to use CPU resources that have been allocated to another VM but are not being used.

`Set ns vpxparam` parameters:

-cpuyield: Release or do not release of allocated but unused CPU resources.

- **YES**: Allow allocated but unused CPU resources to be used by another VM.
- **NO**: Reserve all CPU resources for the VM to which they have been allocated. This option shows higher percentage in hypervisor and cloud environments for VPX CPU usage.
- **DEFAULT**: No.

> **Note**
>
> On all the Citrix ADC VPX platforms, the vCPU usage on the host system is 100 percent. Type the `set ns vpxparam -cpuyield YES` command to override this usage.

If you want to set the cluster nodes to "yield", you must perform the following extra configurations on CCO:

- If a cluster is formed, all the nodes come up with "yield=DEFAULT".
- If a cluster is formed using the nodes that are already set to "yield=YES", then the nodes are added to cluster using the "DEFAULT" yield.

> Note:
>
> If you want to set the cluster nodes to "yield=YES", you can perform suitable configurations only after forming the cluster but not before the cluster is formed.

-masterclockcpu1: You can move the main clock source from CPU0 (management CPU) to CPU1. This parameter has the following options:

- **YES**: Allow VM to move the main clock source from CPU0 to CPU1.
- **NO**: VM uses CPU0 for the main clock source. By default, CPU0 is the main clock source.

- `show ns vpxparam`

  Display the current `vpxparam` settings.

Other References

- For Citrix Ready products, visit Citrix Ready Marketplace.
- For Citrix Ready product support, see the FAQ page.
- For VMware ESX hardware versions, see Upgrading VMware Tools.

Optimize Citrix ADC VPX performance on VMware ESX, Linux KVM, and Citrix Hypervisors

September 29, 2021

The Citrix ADC VPX performance greatly varies depending on the hypervisor, allocated system resources, and the host configurations. To achieve the desired performance, first follow the recommendations in the VPX data sheet, and then further optimize it using the best practices provided in this document.

Citrix ADC VPX instance on VMware ESX hypervisors

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of Citrix ADC VPX instance on VMware ESX hypervisors.

- Recommended configuration on ESX hosts
- Citrix ADC VPX with E1000 network interfaces
- Citrix ADC VPX with VMXNET3 network interfaces
- Citrix ADC VPX with SR-IOV and PCI passthrough network interfaces

Recommended configuration on ESX hosts

To achieve high performance for VPX with E1000, VMXNET3, SR-IOV, and PCI passthrough network interfaces, follow these recommendations:

- The total number of virtual CPUs (vCPUs) provisioned on the ESX host must be less than or equal to the total number of physical CPUs (pCPUs) on the ESX host.
- Non-uniform Memory Access (NUMA) affinity and CPU affinity must be set for the ESX host to achieve good results.

  – To find the NUMA affinity of a Vmnic, log in to the host locally or remotely, and type:

```
1  #vsish -e get /net/pNics/vmnic7/properties | grep NUMA
2  Device NUMA Node: 0
3  <!--NeedCopy-->
```

  - To set NUMA and vCPU affinity for a VM, see VMware documentation.

Citrix ADC VPX with E1000 network interfaces

Perform the following settings on the VMware ESX host:

- On the VMware ESX host, create two vNICs from one pNIC vSwitch. Multiple vNICs create multiple Rx threads in the ESX host. This increases the Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase vNIC transmit (Tx) throughput, use a separate Tx thread in the ESX host per vNIC. Use the following ESX command:
  - For ESX version 5.5:

```
esxcli system settings advanced set -o /Net/NetTxWorldlet -i
<!--NeedCopy-->
```

  - For ESX version 6.0 onwards:

```
esxcli system settings advanced set -o /Net/NetVMTxType -i 1
<!--NeedCopy-->
```

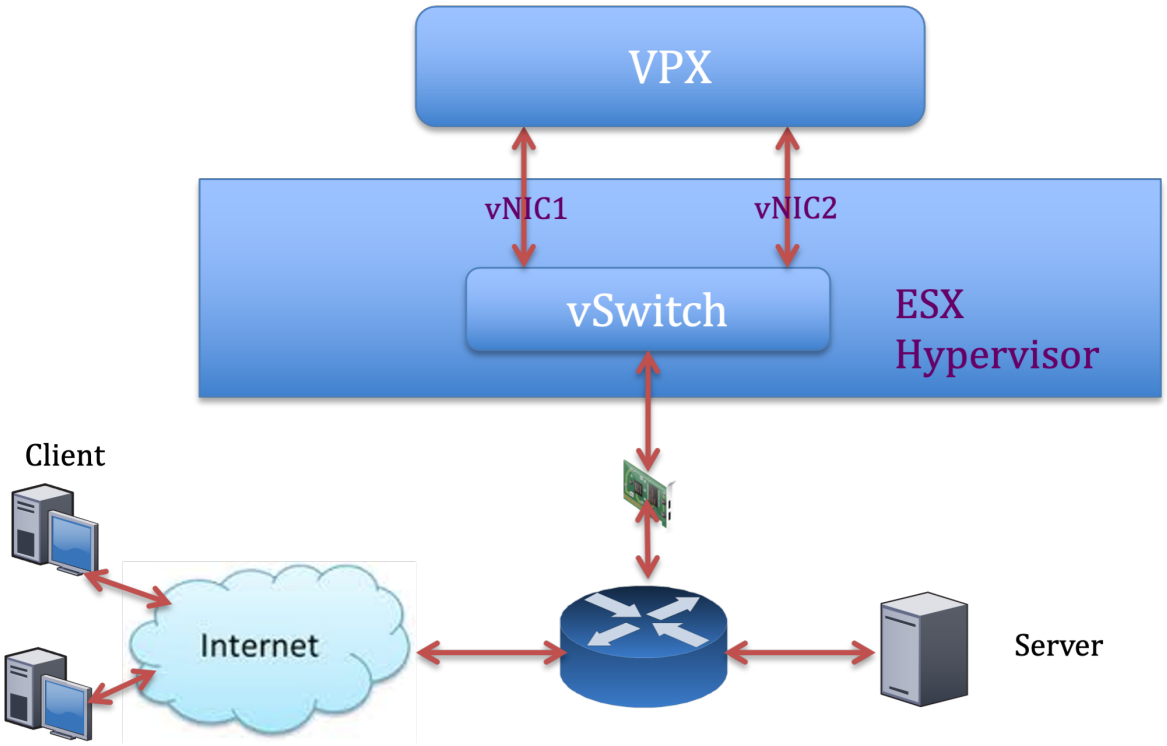
- To further increase the vNIC Tx throughput, use a separate Tx completion thread and Rx threads per device (NIC) queue. Use the following ESX command:

```
esxcli system settings advanced set -o /Net/
   NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```

> Note:
>
> Make sure that you reboot the VMware ESX host to apply the updated settings.

Two vNICs per pNIC deployment

The following is a sample topology and configuration commands for the **Two vNICs per pNIC** model of deployment that delivers better network performance.



Citrix ADC VPX sample configuration:

To achieve the deployment shown in the preceding sample topology, perform the following configuration on the Citrix ADC VPX instance:

- On the client side, bind the SNIP (1.1.1.2) to network interface 1/1 and enable the VLAN tag mode.

```
1  bind vlan 2 -ifnum 1/1  - tagged
2  bind vlan 2 -IPAddress 1.1.1.2 255.255.255.0
3  <!--NeedCopy-->
```

- On the server side, bind the SNIP (2.2.2.2) to network interface 1/1 and enable the VLAN tag mode.

```
1  bind vlan 3 -ifnum 1/2  - tagged
2  bind vlan 3 -IPAddress 2.2.2.2 255.255.255.0
3  <!--NeedCopy-->
```

- Add an HTTP virtual server (1.1.1.100) and bind it to a service (2.2.2.100).

```
1  add lb vserver v1 HTTP 1.1.1.100 80 -persistenceType NONE -
       Listenpolicy None -cltTimeout 180
2  add service s1 2.2.2.100 HTTP 80 -gslb NONE -maxClient 0 -maxReq
       0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
       180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
```



```
3  bind lb vserver v1 s1
4  <!--NeedCopy-->
```

> Note:
>
> Make sure that you include the following two entries in the route table:
>
> - 1.1.1.0/24 subnet with gateway pointing to SNIP 1.1.1.2
> - 2.2.2.0/24 subnet with gateway pointing to SNIP 2.2.2.2

Citrix ADC VPX with VMXNET3 network interfaces

To achieve high performance for VPX with VMXNET3 network interfaces, do the following settings on the VMware ESX host:

- Create two vNICs from one pNIC vSwitch. Multiple vNICs create multiple Rx threads in the ESX host. This increases the Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase vNIC transmit (Tx) throughput, use a separate Tx thread in the ESX host per vNIC. Use the following ESX commands:
  - For ESX version 5.5:

```
1  esxcli system settings advanced set  -o /Net/NetTxWorldlet  -i
2  <!--NeedCopy-->
```

  - For ESX version 6.0 onwards:

```
1   esxcli system settings advanced set -o /Net/NetVMTxType  -i 1
2   <!--NeedCopy-->
```

On the VMware ESX host, perform the following configuration:

- On the VMware ESX host, create two vNICs from 1 pNIC vSwitch. Multiple vNICs create multiple Tx and Rx threads in the ESX host. This increases the Tx and Rx throughput of the pNIC interface.
- Enable VLANs on the vSwitch port group level for each vNIC that you have created.
- To increase Tx throughput of a vNIC, use a separate Tx completion thread and Rx threads per device (NIC) queue. Use the following command:

```
1  esxcli system settings advanced set -o /Net/
      NetNetqRxQueueFeatPairEnable -i 0
2  <!--NeedCopy-->
```

- Configure a VM to use one transmit thread per vNIC, by adding the following setting to the VM's configuration:


```
ethernetX.ctxPerDev = "1"
<!--NeedCopy-->
```

For more information, see Best Practices for Performance Tuning of Telco and NFV Workloads in vSphere

> Note:
>
> Make sure that you reboot the VMware ESX host to apply the updated settings.

You can configure VMXNET3 as a **Two vNICs per pNIC** deployment. For more information, see Two vNICs per pNIC deployment.

Citrix ADC VPX with SR-IOV and PCI passthrough network interfaces

To achieve high performance for VPX with SR-IOV and PCI passthrough network interfaces, see Recommended configuration on ESX hosts.

Citrix ADC VPX instance on Linux-KVM platform

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of Citrix ADC VPX instance on Linux-KVM platform.

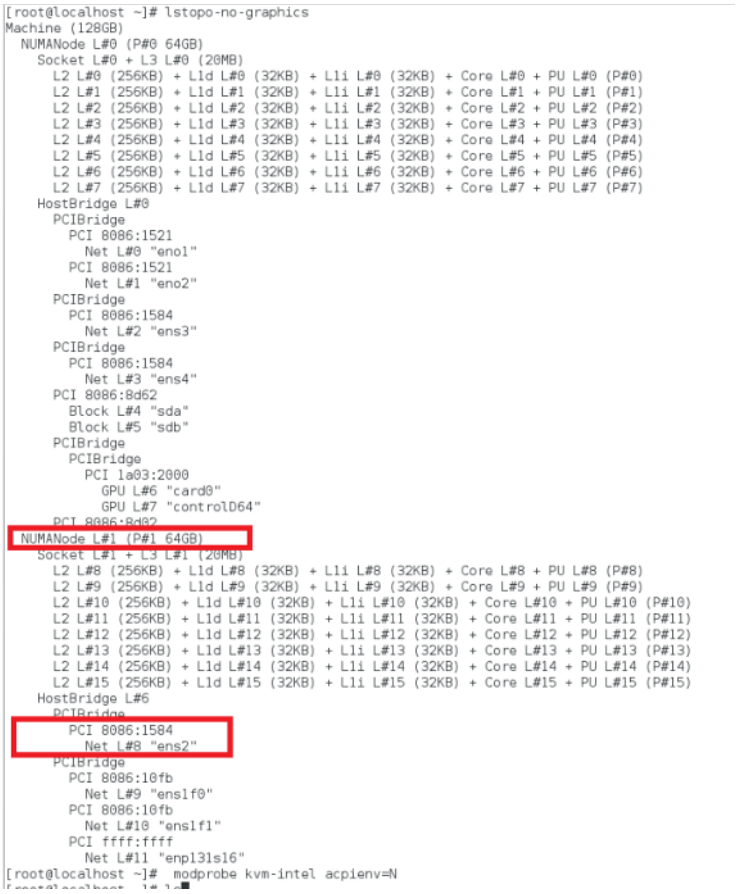
- Performance settings for KVM
- Citrix ADC VPX with PV network interfaces
- Citrix ADC VPX with SR-IOV and Fortville PCIe passthrough network interfaces

Performance settings for KVM

Perform the following settings on the KVM host:

Find the NUMA domain of the NIC using the `lstopo` command:

Make sure that memory for the VPX and the CPU is pinned to the same location.
In the following output, the 10G NIC “ens2” is tied to NUMA domain #1.



Allocate the VPX memory from the NUMA domain.

The `numactl` command indicates the NUMA domain from which the memory is allocated. In the following output, around 10 GB RAM is allocated from NUMA node #0.

```
[root@localhost ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 65429 MB
node 0 free: 55854 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 65536 MB
node 1 free: 52388 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
[root@localhost ~]#
```

To change the NUMA node mapping, follow these steps.

1. Edit the .xml of the VPX on the host.

```
1 /etc/libvirt/qemu/<VPX_name>.xml
2 <!--NeedCopy-->
```


2. Add the following tag:

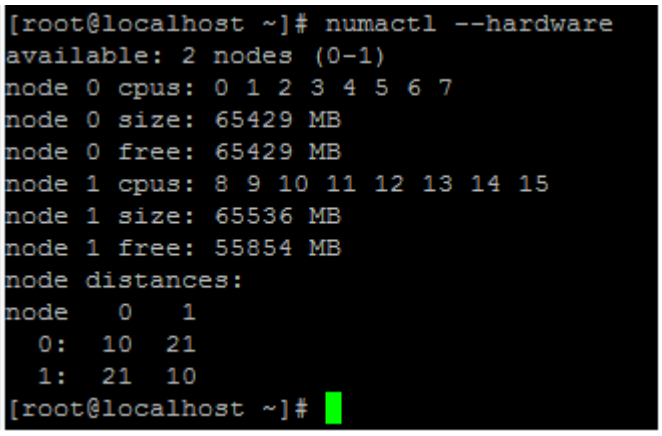
```
<numatune>
<memory mode="strict" nodeset="1"/>    This is the NUMA domain
   name
</numatune>
<!--NeedCopy-->
```

3. Shut down the VPX.
4. Run the following command:

```
virsh define /etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

   This command updates the configuration information for the VM with the NUMA node mappings.

5. Power on the VPX. Then check the `numactl -hardware` command output on the host to see the updated memory allocations for the VPX.



Pin vCPUs of VPX to physical cores.

- To view the vCPU to pCPU mappings of a VPX, type the following command

```
virsh vcpupin <VPX name>
<!--NeedCopy-->
```



```
root@localhost qemu]#  virsh vcpupin NS-VPX-DVR
CPU: CPU Affinity
----------------------------------
  0: 8
  1: 9
  2: 10
  3: 11
```

The vCPUs 0–4 are mapped to physical cores 8–11.

- To view the current pCPU usage, type the following command:

```
mpstat -P ALL 5
<!--NeedCopy-->
```

```
[root@localhost qemu]# mpstat -P ALL 5
Linux 3.10.0-123.el7.x86_64 (localhost.localdomain)     05/17/2016      _x86_64_        (16 CPU)

02:26:20 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
02:26:25 PM  all    0.24    0.00    1.67    0.00    0.00    0.00    0.00   17.32    0.00   80.78
02:26:25 PM    0    0.20    0.00    1.00    0.00    0.00    0.00    0.00    0.00    0.00   98.80
02:26:25 PM    1    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    2    0.20    0.00    0.40    0.00    0.00    0.00    0.00    0.00    0.00   99.40
02:26:25 PM    3    0.00    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.80
02:26:25 PM    4    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    5    0.60    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.20
02:26:25 PM    6    0.40    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    7    1.62    0.00    1.42    0.00    0.00    0.00    0.00    0.00    0.00   96.96
02:26:25 PM    8    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM    9    0.00    0.00    7.60    0.00    0.00    0.00    0.00   92.40    0.00    0.00
02:26:25 PM   10    0.20    0.00    7.00    0.00    0.00    0.00    0.00   92.80    0.00    0.00
02:26:25 PM   11    0.00    0.00    8.60    0.00    0.00    0.00    0.00   91.40    0.00    0.00
02:26:25 PM   12    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   13    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   14    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   15    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
```

In this output, 8 is management CPU, and 9–11 are packet engines.

- To change the vCPU to pCPU pinning, there are two options.
  - Change it at runtime after the VPX boots up using the following command:

```
virsh vcpupin <VPX name> <vCPU id>  <pCPU number>
virsh vcpupin NetScaler-VPX-XML 0 8
virsh vcpupin NetScaler-VPX-XML 1 9
virsh vcpupin NetScaler-VPX-XML 2 10
virsh vcpupin NetScaler-VPX-XML 3 11
<!--NeedCopy-->
```

  - To make static changes to the VPX, edit the `.xml` file as before with the following tags:
    1. Edit the .xml file of the VPX on the host

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```


2. Add the following tag:

```
<vcpu placement='static' cpuset='8-11'>4</vcpu>
    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>
<!--NeedCopy-->
```

3. Shut down the VPX.
4. Update the configuration information for the VM with the NUMA node mappings using the following command:

```
virsh define /etc/libvirt/qemu/ <VPX_name>.xml
<!--NeedCopy-->
```

5. Power on the VPX. Then check the `virsh vcpupin <VPX name>` command output on the host to see the updated CPU pinning.

Eliminate host interrupt overhead.

- Detect VM_EXITS using the `kvm_stat` command.

  At the hypervisor level, host interrupts are mapped to the same pCPUs on which the vCPUs of the VPX are pinned. This might cause vCPUs on the VPX to get kicked out periodically.

  To find the VM exits done by VMs running the host, use the `kvm_stat` command.

```
[root@localhost ~]# kvm_stat -1  | grep EXTERNAL
kvm_exit(EXTERNAL_INTERRUPT)  1728349 27738
[root@localhost ~]#
<!--NeedCopy-->
```

  A higher value in the order of 1+M indicates an issue.

  If a single VM is present, the expected value is 30–100 K. Anything more than that can indicate that there are one or more host interrupt vectors mapped to the same pCPU.

- Detect host interrupts and migrate host interrupts.

  When you run the `concatenate` command for the “/proc/interrupts” file, it displays all the host interrupt mappings. If one or more active IRQs map to the same pCPU, its corresponding counter increments.

  Move any interrupts that overlap with your Citrix ADC VPX’s pCPUs to unused pCPUs:


```
echo 0000000f > /proc/irq/55/smp_affinity
0000000f - - > it is a bitmap, LSBs indicates that IRQ 55 can
   only be scheduled on pCPUs 0 - 3
<!--NeedCopy-->
```

- Disable IRQ balance.

  Disable IRQ balance daemon, so that no rescheduling happens on the fly.

```
service irqbalance stop
service irqbalance show - To check the status
service irqbalance start - Enable if needed
<!--NeedCopy-->
```

  Make sure you run the `kvm_stat` command to ensure that there are not many counters.

Citrix ADC VPX with PV network interfaces

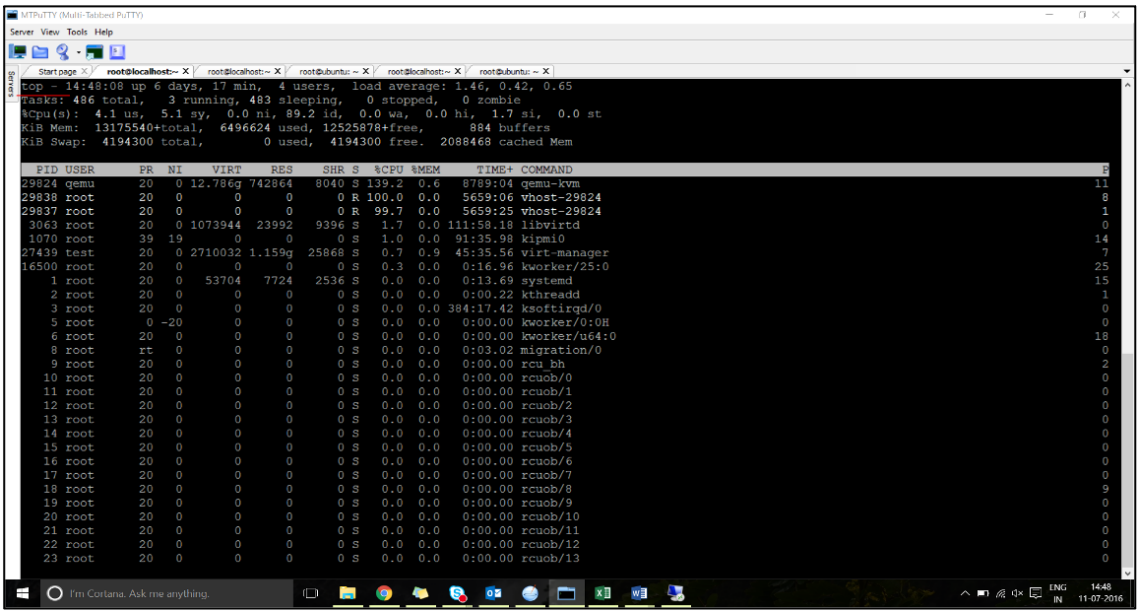
You can configure para-virtualization (PV), SR-IOV, and PCIe passthrough network interfaces as a **Two vNICs per pNIC** deployment. For more information, see Two vNICs per pNIC deployment.

For optimal performance of PV (virtio) interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot/NIC is tied to.
- The Memory and vCPU for the VPX must be pinned to the same NUMA domain.
- Vhost thread must be bound to the CPUs in the same NUMA domain.

Bind the virtual host threads to the corresponding CPUs:

1. Once the traffic is started, run the `top` command on the host.



2. Identify the virtual host process (named as vhost-<pid-of-qemu>) affinity.
3. Bind the vHost processes to the physical cores in the NUMA domain identified earlier using the following command:

```
taskset -pc <core-id> <process-id>
<!--NeedCopy-->
```

Example:

```
taskset -pc 12 29838
<!--NeedCopy-->
```

4. The processor cores corresponding to the NUMA domain can be identified with the following command:

```
[root@localhost ~]# virsh capabilities | grep cpu
<cpu>
    </cpu>
        <cpus num='8'>
            <cpu id='0' socket_id='0' core_id='0' siblings='0'/>
            <cpu id='1' socket_id='0' core_id='1' siblings='1'/>
            <cpu id='2' socket_id='0' core_id='2' siblings='2'/>
            <cpu id='3' socket_id='0' core_id='3' siblings='3'/>
            <cpu id='4' socket_id='0' core_id='4' siblings='4'/>
            <cpu id='5' socket_id='0' core_id='5' siblings='5'/>
            <cpu id='6' socket_id='0' core_id='6' siblings='6'/>
            <cpu id='7' socket_id='0' core_id='7' siblings='7'/>
        </cpus>

        <cpus num='8'>
        <cpu id='8' socket_id='1' core_id='0' siblings='8'/>
        <cpu id='9' socket_id='1' core_id='1' siblings='9'/>
        <cpu id='10' socket_id='1' core_id='2' siblings='10'/>
        <cpu id='11' socket_id='1' core_id='3' siblings='11'/>
        <cpu id='12' socket_id='1' core_id='4' siblings='12'/>
        <cpu id='13' socket_id='1' core_id='5' siblings='13'/>
        <cpu id='14' socket_id='1' core_id='6' siblings='14'/>
        <cpu id='15' socket_id='1' core_id='7' siblings='15'/>
        </cpus>

    <cpuselection/>
    <cpuselection/>

<!--NeedCopy-->
```


Bind the QEMU process to the corresponding physical core:

1. Identify the physical cores on which the QEMU process is running. For more information, see the preceding output.
2. Bind the QEMU process to the same physical cores to which you bind the vCPUs, using the following command:

```
taskset -pc 8-11 29824
<!--NeedCopy-->
```

Citrix ADC VPX with SR-IOV and Fortville PCIe passthrough network interfaces

For optimal performance of the SR-IOV and Fortville PCIe passthrough network interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot/NIC is tied to.
- The Memory and vCPU for the VPX must be pinned to the same NUMA domain.

Sample VPX XML file for vCPU and memory pinning for Linux KVM:

```
    <domain type='kvm'>
        <name>NetScaler-VPX</name>
        <uuid>138f7782-1cd3-484b-8b6d-7604f35b14f4</uuid>
        <memory unit='KiB'>8097152</memory>
        <currentMemory unit='KiB'>8097152</currentMemory>
        <vcpu placement='static'>4</vcpu>

    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>

    <numatune>
    <memory mode='strict' nodeset='1'/>
    </numatune>

    </domain>
<!--NeedCopy-->
```


Citrix ADC VPX instance on Citrix Hypervisors

This section contains details of configurable options and settings, and other suggestions that help you achieve optimal performance of Citrix ADC VPX instance on Citrix Hypervisors.

- Performance settings for Citrix Hypervisors
- Citrix ADC VPX with SR-IOV network interfaces
- Citrix ADC VPX with para-virtualized interfaces

Performance settings for Citrix Hypervisors

Find the NUMA domain of the NIC using the “xl” command:

```
xl info -n
<!--NeedCopy-->
```

Pin vCPUs of VPX to physical cores.

```
xl vcpu-pin <Netsclaer VM Name>  <vCPU id>  <physical CPU id>
<!--NeedCopy-->
```

Check binding of vCPUs.

```
xl vcpu-list
<!--NeedCopy-->
```

Allocate more than 8 vCPUs to Citrix ADC VMs.

For configuring more than 8 vCPUs, run the following commands from the Citrix Hypervisor console:

```
xe vm-param-set uuid=your_vms_uuid VCPUs-max=16
xe vm-param-set uuid=your_vms_uuid VCPUs-at-startup=16
<!--NeedCopy-->
```

Citrix ADC VPX with SR-IOV network interfaces

For optimal performance of the SR-IOV network interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot or NIC is tied to.
- Pin the Memory and vCPU for the VPX to the same NUMA domain.
- Bind the Domain-0 vCPU to the remaining CPU.

Citrix ADC VPX with para-virtualized interfaces

For optimal performance, two vNICs per pNIC and one vNIC per pNIC configurations are advised, as in other PV environments.

To achieve optimal performance of para-virtualized (netfront) interfaces, follow these steps:

- Identify the NUMA domain to which the PCIe slot or NIC is tied to.
- Pin the memory and vCPU for the VPX to the same NUMA domain.
- Bind the Domain-0 vCPU to the remaining CPU of the same NUMA domain.
- Pin host Rx/Tx threads of vNIC to Domain-0 vCPUs.

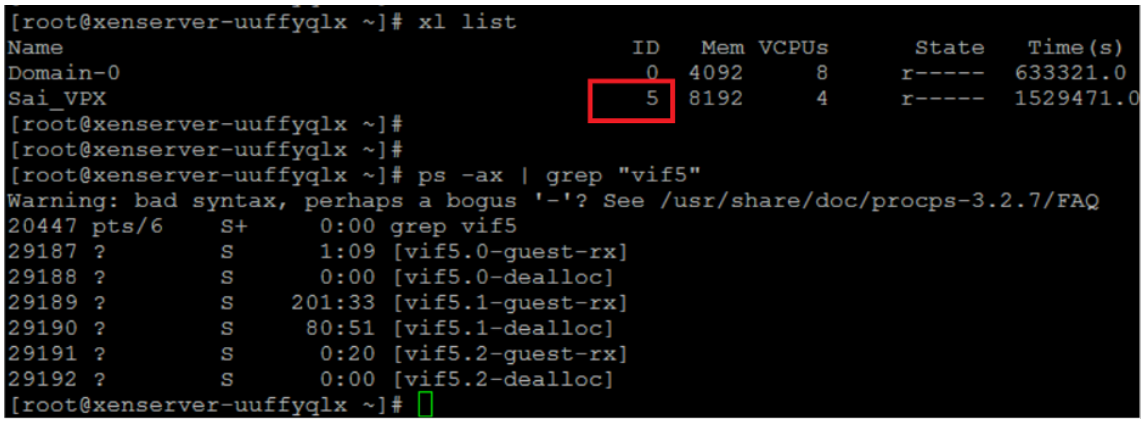
Pin host threads to Domain-0 vCPUs:

1. Find Xen-ID of the VPX by using the `xl list` command on the Citrix Hypervisor host shell.
2. Identify host threads by using the following command:

```
ps -ax | grep vif <Xen-ID>
<!--NeedCopy-->
```

   In the following example, these values indicate:

   - **vif5.0** - The threads for first interface allocated to VPX in XenCenter (management interface).
   - **vif5.1** - The threads for second interface assigned to VPX and so on.



3. Pin the threads to Domain-0 vCPUs using the following command:

```
taskset -pc <core-id> <process-id>
<!--NeedCopy-->
```

   Example:

```
taskset -pc 1 29189
<!--NeedCopy-->
```


Apply Citrix ADC VPX configurations at the first boot of the Citrix ADC appliance in cloud

September 17, 2021

You can apply the Citrix ADC VPX configurations during the first boot of the Citrix ADC appliance in a cloud environment. This stage is addressed as the **preboot** stage in this document. Therefore in certain cases like ADC pooled licensing, a specific VPX instance is brought up in much lesser time. This feature is available in Microsoft Azure, Google Cloud platform, and AWS clouds.

What is user data

When you provision a VPX instance in a cloud environment, you have the option of passing user data to the instance. The user data allows you to perform common automated configuration tasks, customize the startup behaviors of instances, and run scripts after the instance starts. At the first boot, the Citrix ADC VPX instance performs the following tasks:

- Reads the user data.
- Interprets the configuration provided in user data.
- Applies the newly added configuration as it boots up.

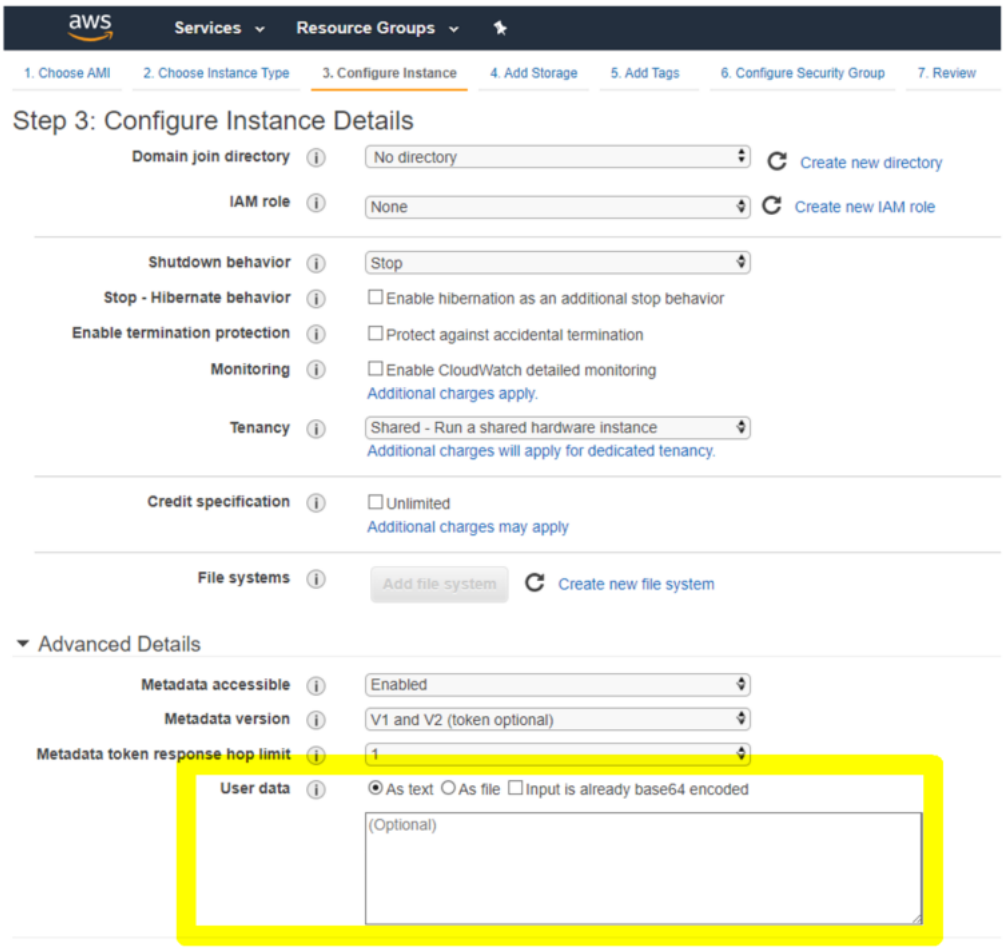
How to provide preboot user data in cloud instance

You can provide preboot user data to the cloud instance in XML format. Different clouds have different interfaces for providing user data.

Provide preboot user data using the AWS console

When you provision a Citrix ADC VPX instance using the AWS console, navigate to **Configure Instance Details > Advanced Details**, and provide the preboot user data configuration in the **User data** field.

For detailed instructions on each of the steps, see Deploy a Citrix ADC VPX instance on AWS by using the AWS web console.
For more information, see AWS documentation on Launching an instance.



Provide preboot user data using AWS CLI

Type the following command in the AWS CLI:

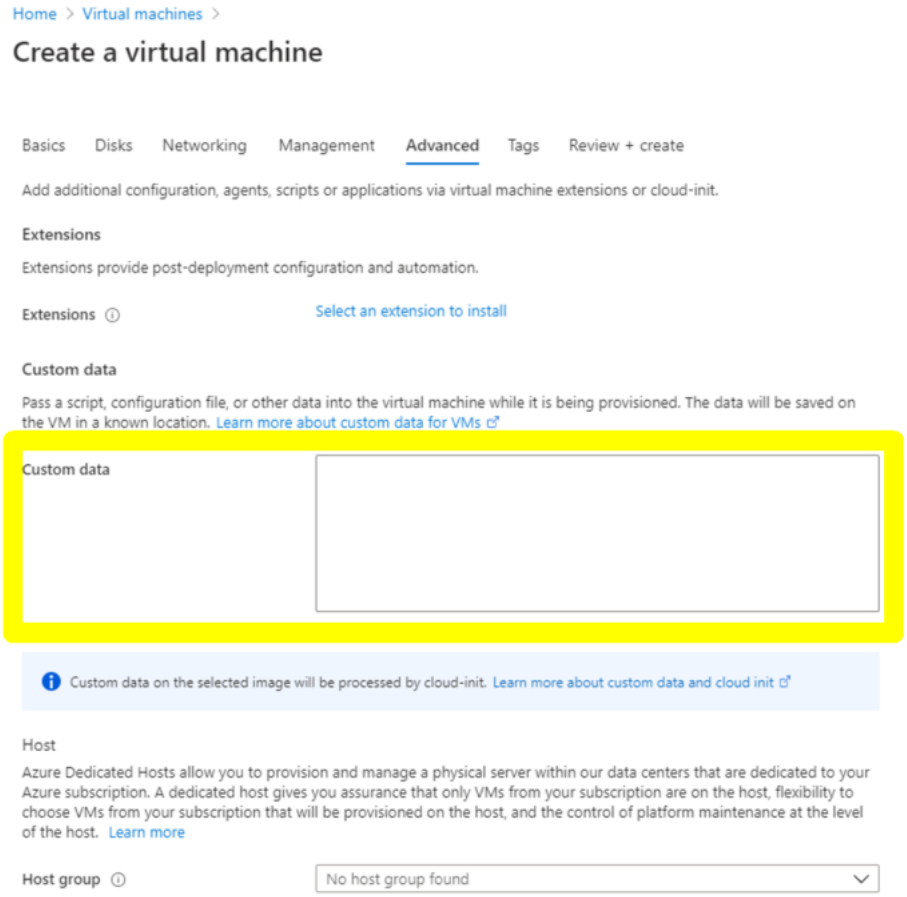
```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --user-data file://my_script.txt
<!--NeedCopy-->
```

For more information, see AWS documentation on Running instances.

For more information, see AWS documentation on Using instance user data

Provide preboot user data using the Azure console

When you provision a Citrix ADC VPX instance using Azure console, navigate to **Create a virtual machine > Advanced** tab. In the **Custom data** field, provide preboot user data configuration.



Provide preboot user data using the Azure CLI

Type the following command in the Azure CLI:

```
az vm create \
  --resource-group myResourceGroup \
  --name MyVm \
  --image debian \
  --custom-data MyCloudInitScript.txt \
<!--NeedCopy-->
```

Example:

```
az vm create --resource-group MyResourceGroup -name MyVm --image debian
    --custom-data MyCloudInitScript.txt
<!--NeedCopy-->
```

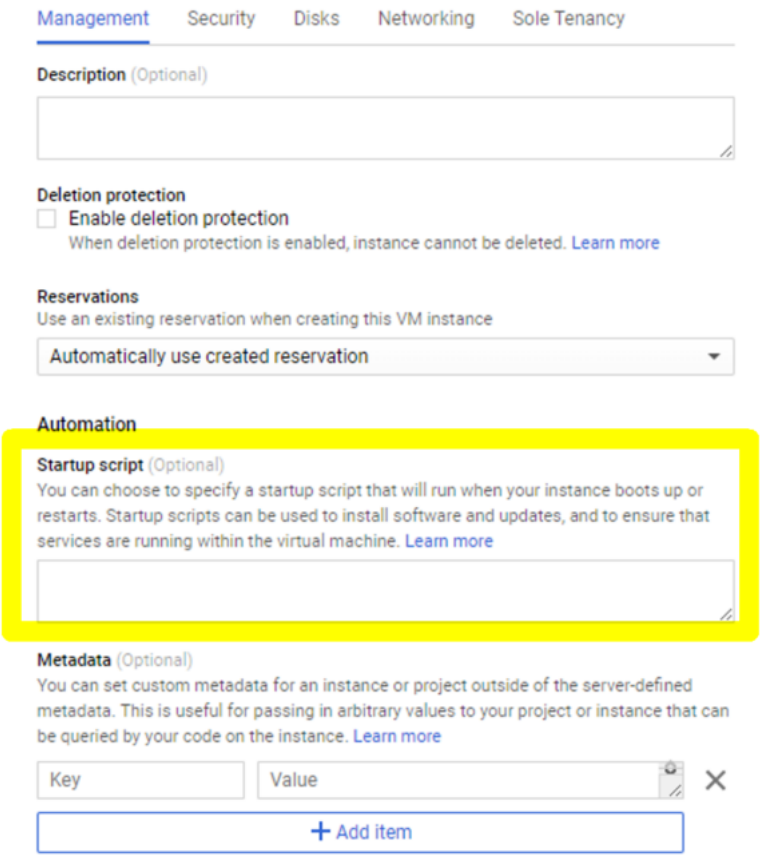

You can pass your custom data or preboot configuration as a file to "–custom-data" parameter. In this example, the file name is **MyCloudInitScript.txt**.

For more information, see Azure CLI documentation.

Provide preboot user data using the GCP console

When you provision a Citrix ADC VPX instance using GCP console, fill in the properties of instance. Expand **Management, security, disks, networking, sole tenancy**. Navigate to the **Management** tab. In the **Automation** section, provide preboot user data configuration in the **Startup Script** field.

For detailed information on creating the VPX instance using GCP, see Deploy a Citrix ADC VPX instance on Google Cloud Platform.



Provide preboot user data using the gcloud CLI

Type the following command in the GCP CLI:


```
gcloud compute instances create INSTANCE_NAMES --metadata-from-file=
   startup-script=LOCAL_FILE_PATH
<!--NeedCopy-->
```

metadata-from-file - Reads the value or user data from a file stored at the .

For more information, see gcloud CLI documentation

Preboot user data format

The preboot user data must be provided to the cloud instance in XML format. The Citrix ADC preboot user data that you provide through the cloud infrastructure during boot can comprise the following four sections:

- Citrix ADC configuration represented with the <NS-CONFIG> tag.
- Custom bootstrapping the Citrix ADC represented with the <NS-BOOTSTRAP> tag.
- Storing user-scripts in Citrix ADC represented with the <NS-SCRIPTS> tag.
- Pooled licensing configuration represented with the <NS-LICENSE-CONFIG> tag.

You can provide the preceding four sections in any order within the ADC preboot configuration. Ensure to strictly follow the formatting shown in the following sections while providing the preboot user data.

> Note:
>
> The entire preboot user data configuration must be enclosed in the <NS-PRE-BOOT-CONFIG> tag as shown in the following examples.

Example 1:

```
<NS-PRE-BOOT-CONFIG>
     <NS-CONFIG>          </NS-CONFIG>
     <NS-BOOTSTRAP>       </NS-BOOTSTRAP>
     <NS-SCRIPTS>         </NS-SCRIPTS>
     <NS-LICENSE-CONFIG>  </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

Example 2:

```
<NS-PRE-BOOT-CONFIG>
    <NS-LICENSE-CONFIG> </NS-LICENSE-CONFIG>
    <NS-SCRIPTS>        </NS-SCRIPTS>
    <NS-BOOTSTRAP>      </NS-BOOTSTRAP>
    <NS-CONFIG>         </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
```



```
<!--NeedCopy-->
```

Use the <NS-CONFIG> tag to provide the specific Citrix ADC VPX configurations that needs to be applied to the VPX instance at the preboot stage.

> NOTE:
>
> The <NS-CONFIG> section must have valid ADC CLI commands. The CLIs are not verified for the syntactic errors or format.

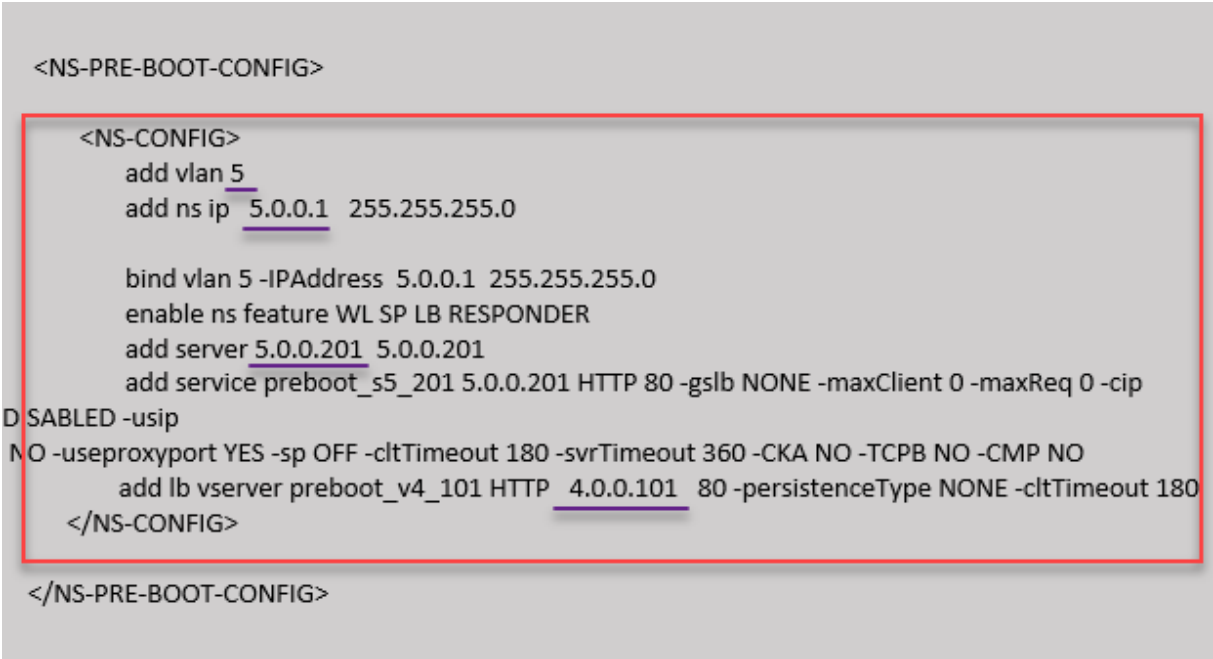
Citrix ADC configurations

Use the <NS-CONFIG> tag to provide the specific Citrix ADC VPX configurations that needs to be applied to the VPX instance at the preboot stage.

> NOTE:
>
> The <NS-CONFIG> section must have valid ADC CLI commands. The CLIs are not verified for the syntactic errors or format.

Example:

In the following example, the <NS-CONFIG> section has the details of the configurations. A VLAN of ID '5' is configured and bound to the SNIP (5.0.0.1). A load balancing virtual server (4.0.0.101) is also configured.



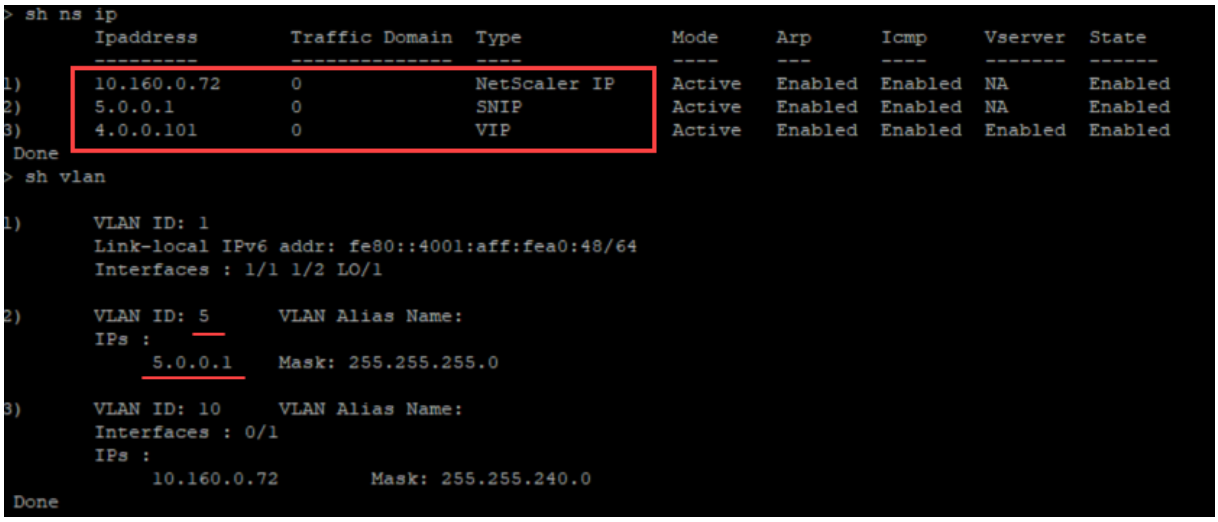
You can copy the configuration shown in the preceding screenshot from here:

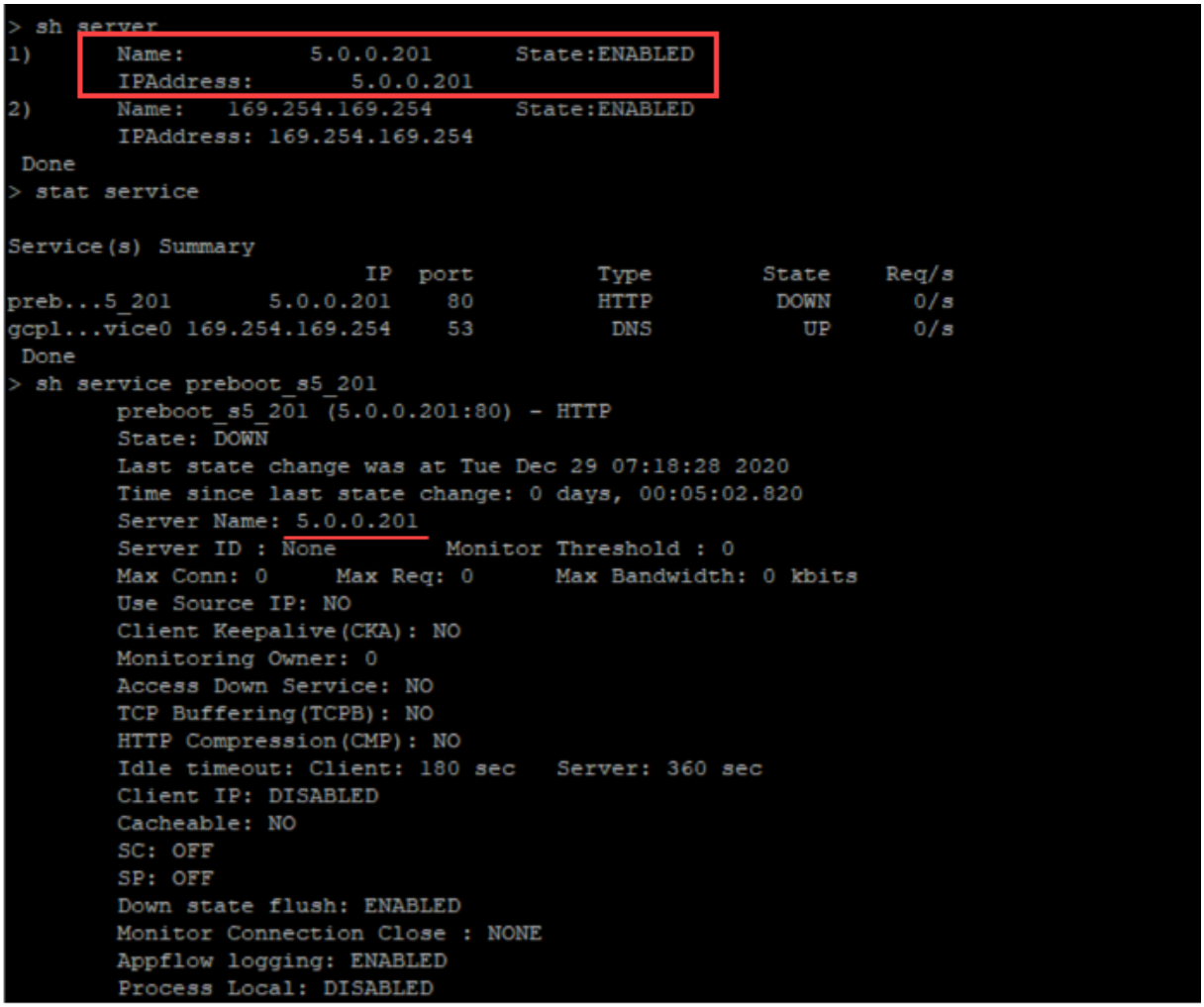
```
<NS-PRE-BOOT-CONFIG>
```



```
      <NS-CONFIG>
          add vlan 5
          add ns ip 5.0.0.1 255.255.255.0
          bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
          enable ns feature WL SP LB RESPONDER
          add server 5.0.0.201 5.0.0.201
          add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
             maxClient 0 -maxReq 0 -cip DISABLED -usip
   NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -
      TCPB NO -CMP NO
          add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
             persistenceType NONE -cltTimeout 180
      </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

The Citrix ADC VPX instance comes up with the configuration applied in the <NS-CONFIG> section as shown in the following illustrations.





User scripts

Use the <NS-SCRIPTS> tag to provide any script that must be stored and ran in Citrix ADC VPX instance.

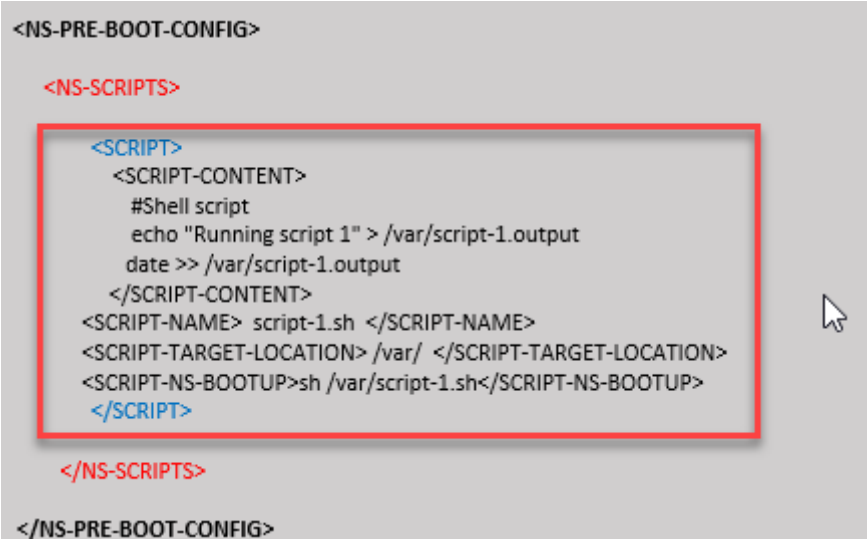
You can include many scripts within the <NS-SCRIPTS> tag. Each script must be included within the <SCRIPT> tag.
Each <SCRIPT> section corresponds to one script and contains all the details of the script using the following sub tags.

- **<SCRIPT-NAME>:** Indicates the name of the script file that must be stored.
- **<SCRIPT-CONTENT>:** Indicates the content of the file that must be stored.
- **<SCRIPT-TARGET-LOCATION>:** Indicates the designated target location where this file must be stored. If the target location is not provided, by default, the file, or script is saved in the "/nsconfig" directory.
- **<SCRIPT-NS-BOOTUP>:** Specify the commands that you use to run the script.

- If you use the <SCRIPT-NS-BOOTUP> section, the commands provided in the section are stored in "/nsconfig/nsafter.sh", and the commands are run after the packet engine boots up as part of "nsafter.sh" execution.
- If you do not use the <SCRIPT-NS-BOOTUP> section, the script file is stored in the target location that you specify.

Example 1:

In this example, the <NS-SCRIPTS> tag contains details of only one script: script-1.sh. The "script-1.sh" script is saved at the "/var" directory. The script is populated with the specified contents, and is run with the "sh /var/script-1.sh" command after packet engine boots up.



You can copy the configuration shown in the preceding screenshot from here:

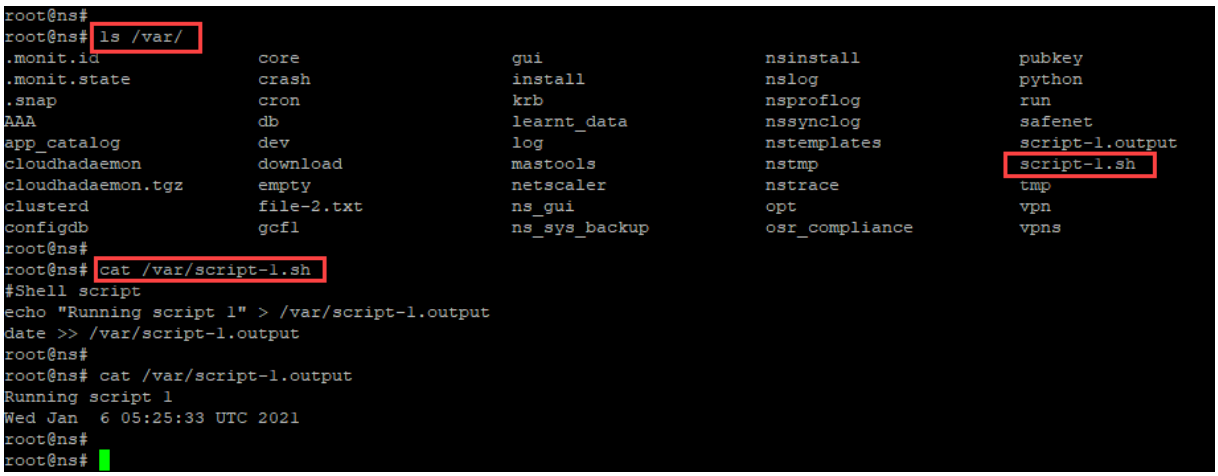
```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
    <SCRIPT>
            <SCRIPT-CONTENT>
                #Shell script
                echo "Running script 1" > /var/script-1.output
                date >> /var/script-1.output
            </SCRIPT-CONTENT>

                <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
                <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION
                    >
                <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP
                    >
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```



```
<!--NeedCopy-->
```

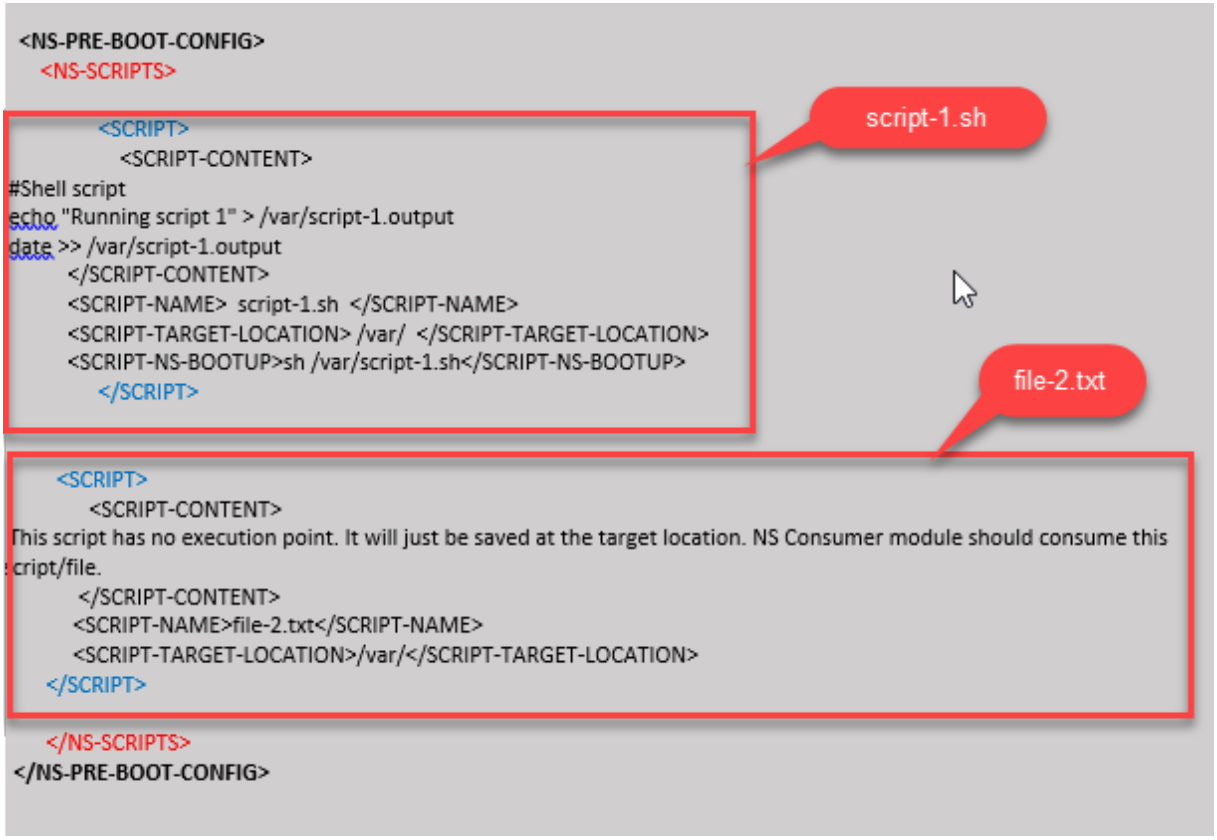
In the following snapshot, you can verify that “script-1.sh” script is saved in the “/var/” directory. The “Script-1.sh” script is run, and the output file is created appropriately.



Example 2:

In the following example, the <NS-SCRIPTS> tag contains details of two scripts.

- The first script is saved as “script-1.sh” at the “/var” directory. The script is populated with the specified contents, and is run with command “sh /var/script-1.sh” after packet engine boots up.
- The second script is saved as “file-2.txt” at the “/var” directory. This file is populated with the specified contents. But it is not run because the bootup execution command <SCRIPT-NS-BOOTUP> is not provided.



You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
        <SCRIPT>
            <SCRIPT-CONTENT>
               #Shell script
               echo "Running script 1" > /var/script-1.output
               date >> /var/script-1.output
            </SCRIPT-CONTENT>

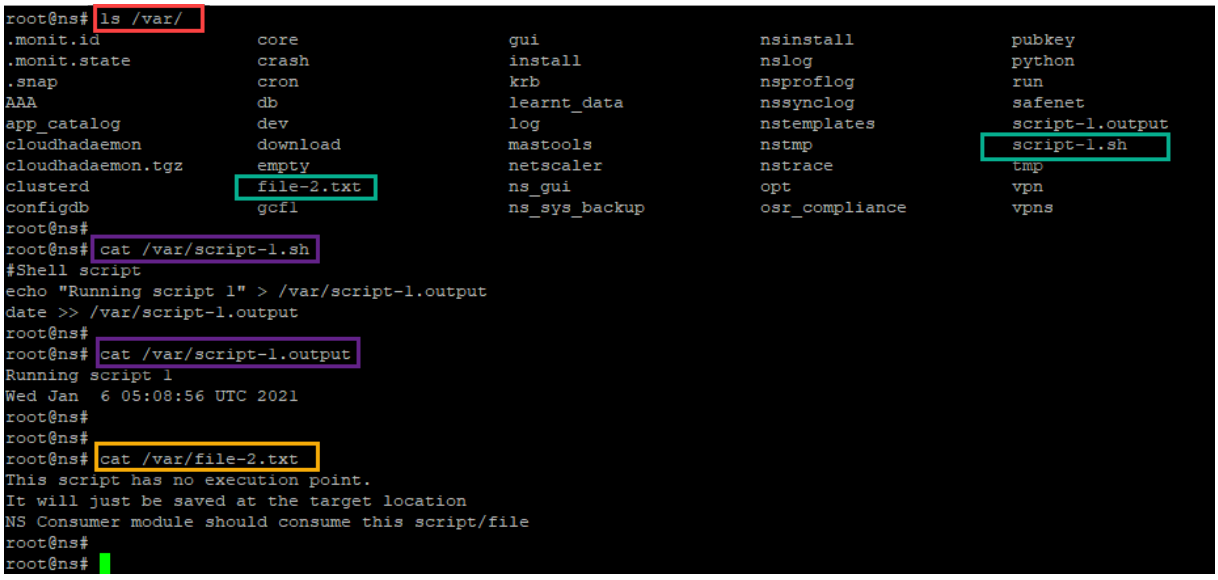
            <SCRIPT-NAME>  script-1.sh  </SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION> /var/  </SCRIPT-TARGET-LOCATION>
            <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
            </SCRIPT>

        <SCRIPT>
            <SCRIPT-CONTENT>
                This script has no execution point.
                It will just be saved at the target location
                NS Consumer module should consume this script/file
            </SCRIPT-CONTENT>
```



```
            <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

In the following snapshot, you can verify that script-1.sh and file-2.txt are created in the "/var/" directory. The Script-1.sh is run, and the output file is created appropriately.



Licensing

Use the <NS-LICENSE-CONFIG> tag to apply Citrix ADC pooled licensing while booting up the VPX instance. Use the <LICENSE-COMMANDS> tag within <NS-LICENSE-CONFIG> section to provide the pooled license commands. These commands must be syntactically valid.

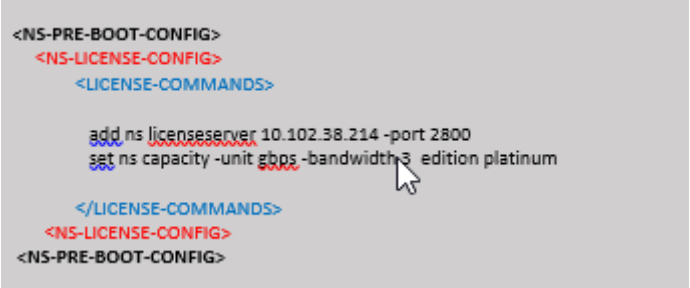
You can specify the pooled licensing details such as, license type, capacity, and license server in the <LICENSE-COMMANDS> section using the standard pooled licensing commands. For more information, see Configure Citrix ADC pooled capacity licensing.

After applying the <NS-LICENSE-CONFIG>, the VPX comes up with the requested edition upon boot, and VPX tries to check out the configured licenses from the license server.

- If the license checkout is successful, the configured bandwidth is applied to VPX.
- If the license checkout fails, the license is not retrieved from license server within 10–12 minutes approximately. As a result, the system reboots and enters an unlicensed state.

Example:

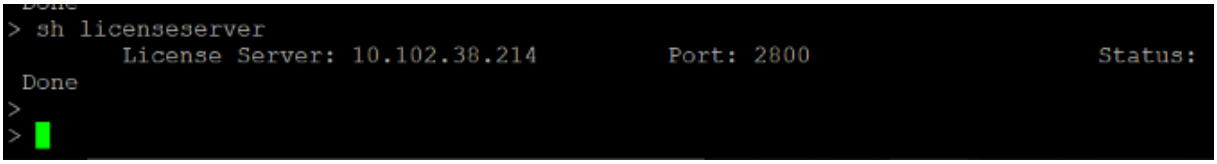
In the following example, after applying the <NS-LICENSE-CONFIG>, the VPX comes up with the Premium edition upon boot, and VPX tries to check out the configured licenses from the license server (10.102.38.214).



You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
   <NS-LICENSE-CONFIG>
        <LICENSE-COMMANDS>
            add ns licenseserver 10.102.38.214 -port 2800
            set ns capacity -unit gbps -bandwidth 3  edition platinum
        </LICENSE-COMMANDS>
    </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

As shown in the following illustration, you can run the "show license server" command, and verify that the license server (10.102.38.214) is added to the VPX.



Bootstrapping

Use the <NS-BOOTSTRAP> tag to provide the custom bootstrapping information. You can use the <SKIP-DEFAULT-BOOTSTRAP> and <NEW-BOOTSTRAP-SEQUENCE> tags within the <NS-BOOTSTRAP> section. This section informs Citrix ADC appliance whether to avoid the default bootstrap or not. If the default bootstrapping is avoided, this section provides you an option to provide a new bootstrapping sequence.

Default bootstrap configuration

The default bootstrap configuration in Citrix ADC appliance follows these interface assignments:

- **Eth0** - Management interface with a certain NSIP address.
- **Eth1** - Client-facing interface with a certain VIP address.
- **Eth2** - Server-facing interface with a certain SNIP address.

Customize bootstrap configuration

You can skip the default bootstrap sequence and provide a new bootstrap sequence for the Citrix ADC VPX instance. Use the <NS-BOOTSTRAP> tag to provide the custom bootstrapping information. For example, you can change the default bootstrapping, where the Management interface (NSIP), Client-facing interface (VIP), and server-facing interface (SNIP) are always provided in certain order.

The following table indicates the bootstrapping behavior with the different values that are allowed for <SKIP-DEFAULT-BOOTSTRAP> and <NEW-BOOTSTRAP-SEQUENCE> tags.

SKIP-DEFAULT-BOOTSTRAP	NEW-BOOTSTRAP-SEQUENCE	Bootstrap behavior
YES	YES	The default bootstrapping behavior is skipped, and a new custom bootstrap sequence provided in the <NS-BOOTSTRAP> section is run.
YES	NO	The default bootstrapping behavior is skipped. The bootstrap commands provided in the <NS-CONFIG> section is run.

You can customize the bootstrap configuration by the following three methods:

- Provide only the interface details
- Provide the interface details along with IP addresses and subnet mask
- Provide bootstrap related commands in the <NS-CONFIG> section

Method 1: Custom bootstrap by specifying only the interface details

You specify the management, client-facing and server-facing interfaces but not their IP addresses and subnet masks. The IP addresses and subnet masks are populated by querying the cloud infrastructure.

Custom bootstrap example for AWS

You provide the custom bootstrap sequence as shown in the following example. For more information, see How to provide preboot user data in cloud instance. Eth1 interface is assigned as the management interface (NSIP), Eth0 interface as the client interface (VIP), and Eth2 interface as the server interface (SNIP). The <NS-BOOTSTRAP> section contains only the interface details and not the details of IP addresses and subnet masks.

```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
      <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
      <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

      <MGMT-INTERFACE-CONFIG>
          <INTERFACE-NUM> eth1 </INTERFACE-NUM>
      </MGMT-INTERFACE-CONFIG>

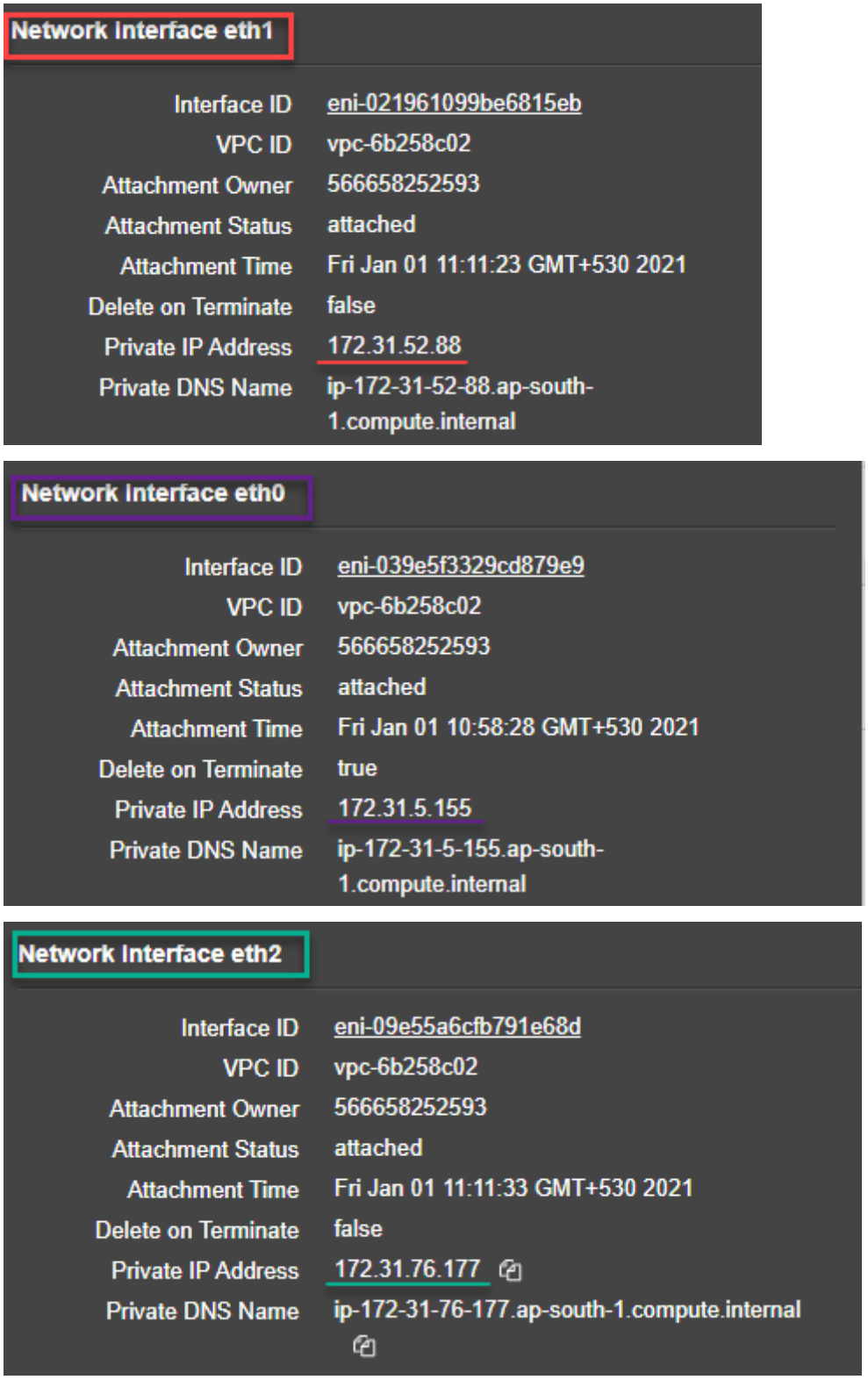
      <CLIENT-INTERFACE-CONFIG>
          <INTERFACE-NUM> eth0 </INTERFACE-NUM>
      </CLIENT-INTERFACE-CONFIG>

      <SERVER-INTERFACE-CONFIG>
          <INTERFACE-NUM> eth2 </INTERFACE-NUM>
      </SERVER-INTERFACE-CONFIG>

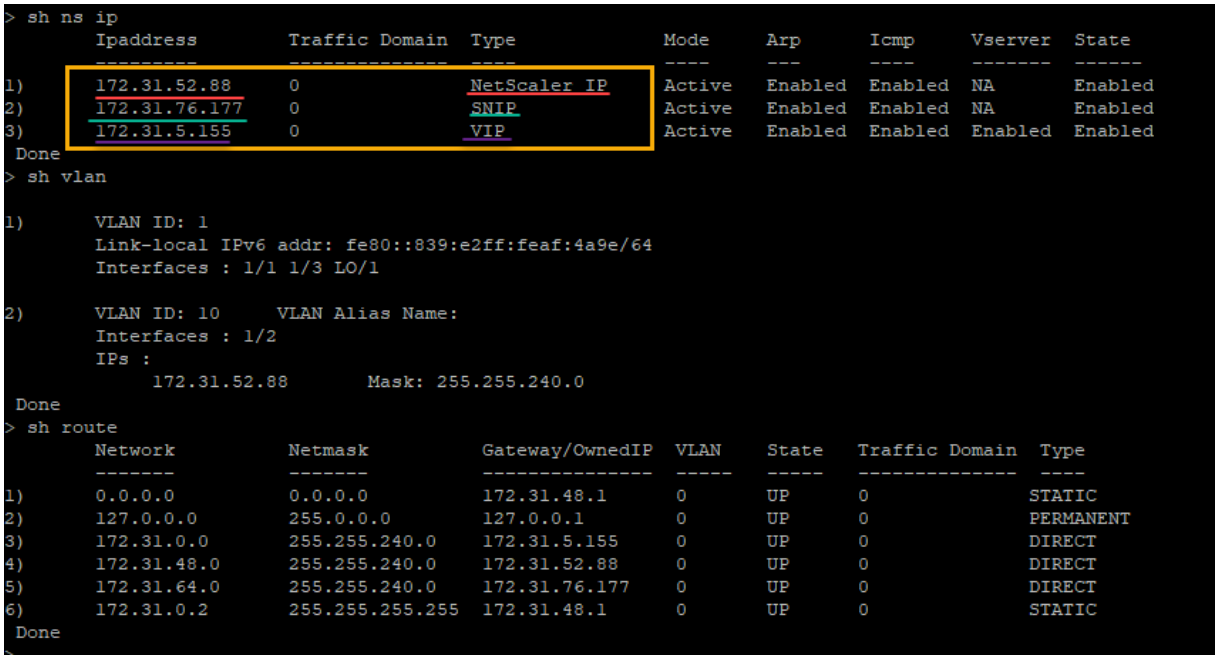
    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

After the VM instance is created, in the AWS portal, you can verify the network interface properties as follows:

1. Navigate to the **AWS Portal > EC2 instances**, and select the instance that you have created by providing the custom bootstrap information.
2. In the **Description** tab, you can verify the properties of each network interface as shown in the following illustrations.



You can run the `show nsip` command in **ADC CLI**, and verify the network interfaces applied to the ADC VPX instance during the first boot of the ADC appliance.



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
```

Custom bootstrap example for Azure

You provide the custom bootstrap sequence as shown in the following example. For more information, see How to provide preboot user data in cloud instance. Eth2 interface is assigned as the management interface (NSIP), Eth1 interface as the client interface (VIP), and Eth0 interface as the server interface (SNIP). The <NS-BOOTSTRAP> section contains only the interface details and not the details of IP addresses and subnet masks.


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

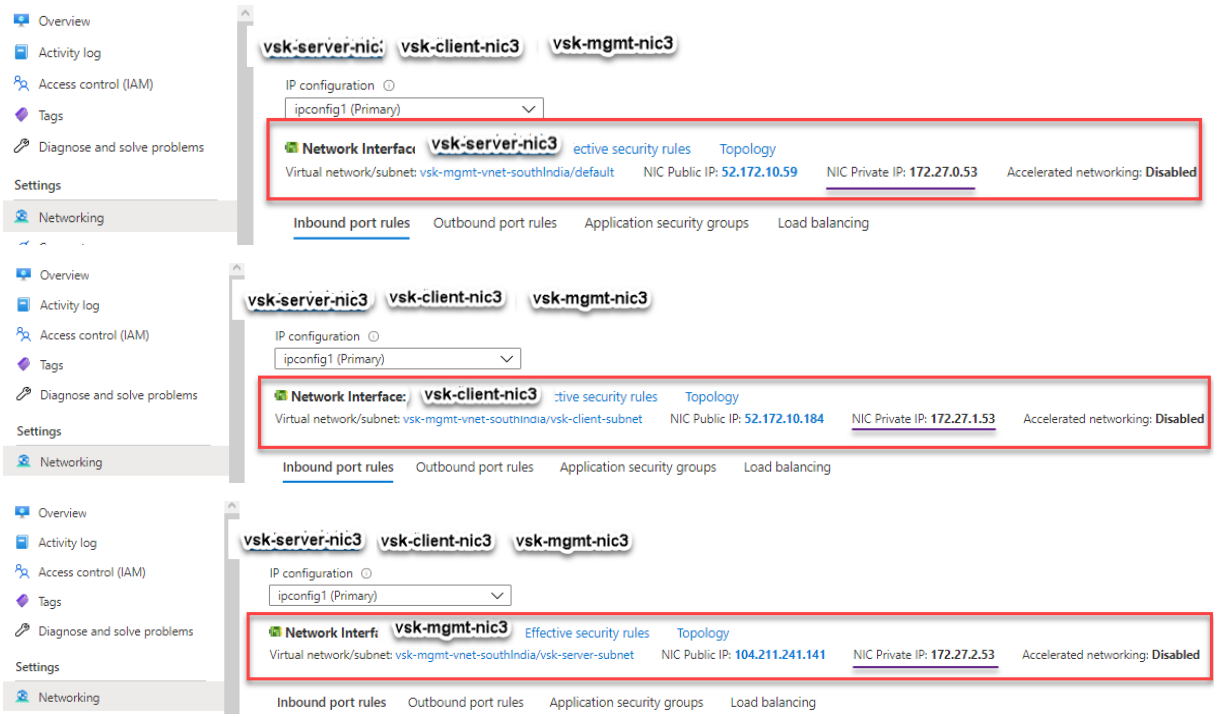
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can see that the Citrix ADC VPX instance is created with three network interfaces. Navigate to the **Azure portal > VM instance > Networking**, and verify the networking properties of the three NICs as shown in the following illustrations.



You can run the “show nsip” command in the ADC CLI, and verify that the new bootstrap sequence

specified in the <NS-BOOTSTRAP> section is applied. You can run the "show route" command to verify the subnet mask.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
>
```

Custom bootstrap examples for GCP

You provide the custom bootstrap sequence as shown in the following example. For more information, see How to provide preboot user data in cloud instance. Eth1 interface is assigned as the management interface (NSIP), Eth0 interface as the client interface (VIP), and Eth2 interface as the server interface (SNIP). The <NS-BOOTSTRAP> section contains only the interface details and not the details of IP addresses and subnet masks.


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

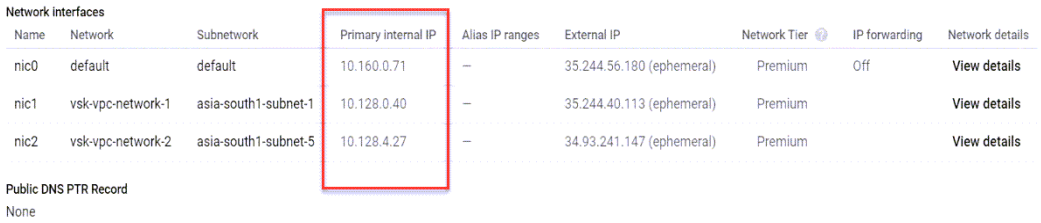
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

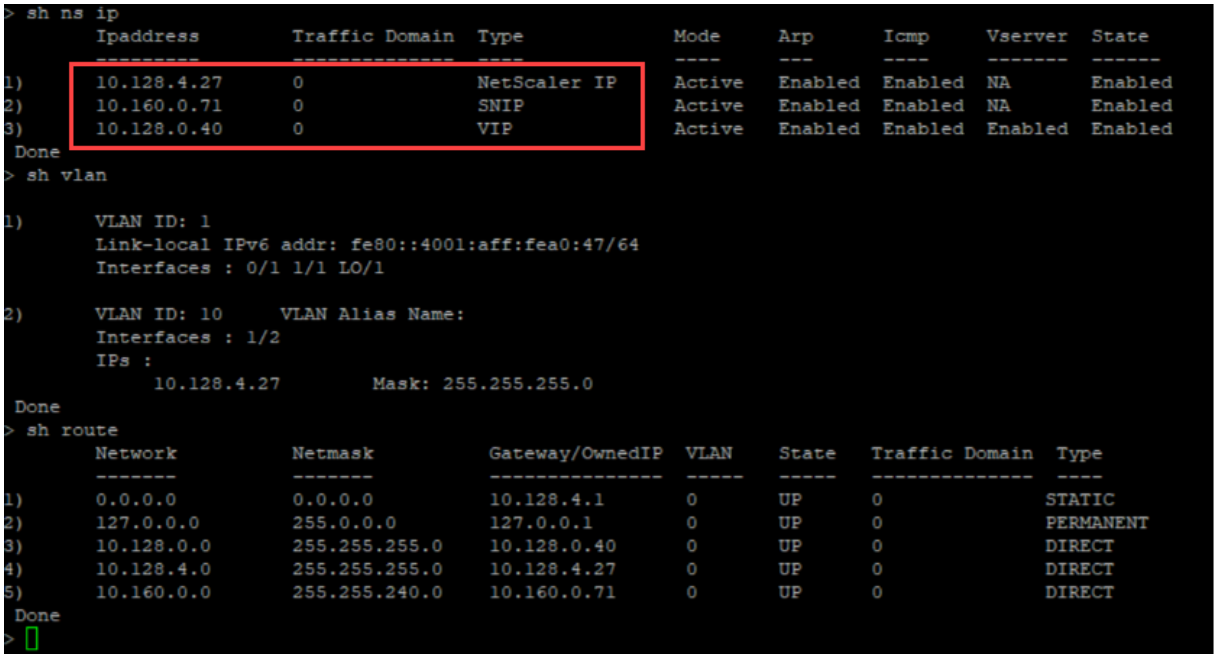
    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

After the VM instance is created in the GCP portal, you can verify the network interface properties as follows:

1. Select the instance that you have created by providing the custom bootstrap information.
2. Navigate to the Network interface properties and verify the NIC details as follows:



You can run the `show nsip` command in **ADC CLI**, and verify the network interfaces applied to the ADC VPX instance during the first boot of the ADC appliance.



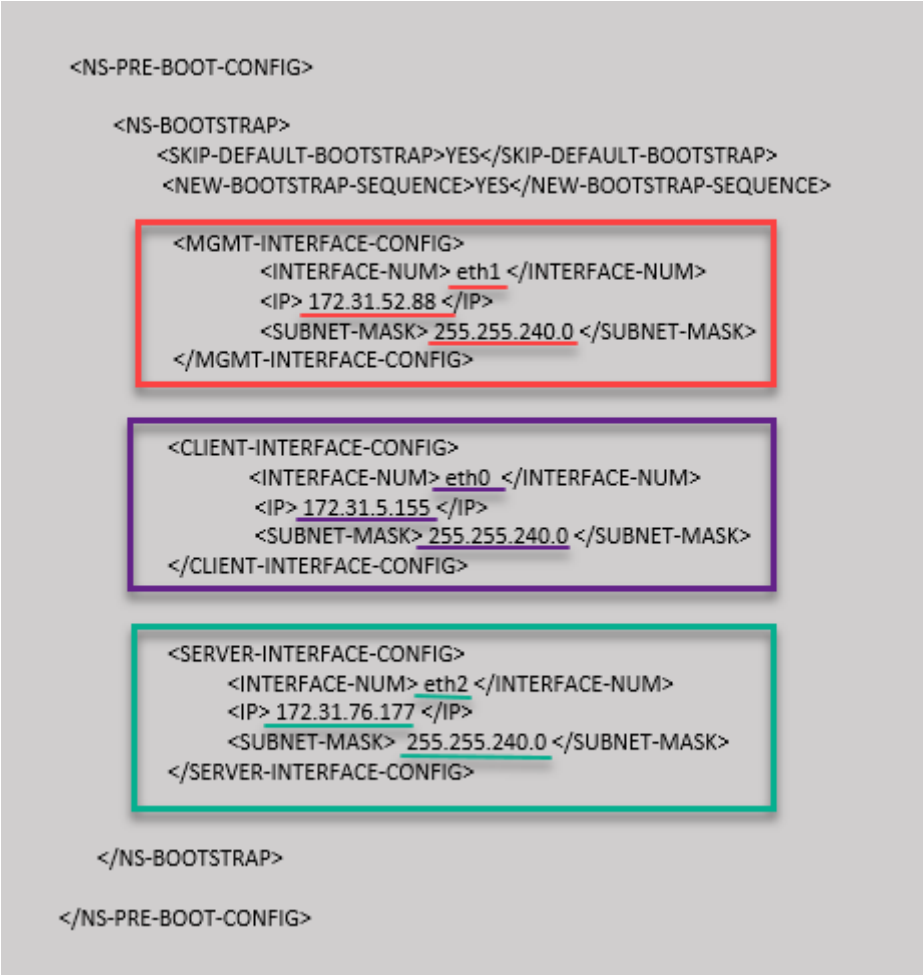
Method 2: Custom bootstrap by specifying the interfaces, IP addresses, and subnet masks

You specify the management, client-facing and server-facing interfaces along with their IP addresses and subnet mask.

Custom bootstrap examples for AWS

In the following example, you skip the default bootstrap and run a new bootstrap sequence for the Citrix ADC appliance. For the new bootstrap sequence, you specify the following details:

- **Management interface:** Interface - Eth1, NSIP - 172.31.52.88, and subnet mask - 255.255.240.0
- **Client facing interface:** Interface - Eth0, VIP - 172.31.5.155, and subnet mask - 255.255.240.0.
- **Server facing interface:** Interface - Eth2, SNIP - 172.31.76.177, and subnet mask - 255.255.240.0.



You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the <NS-BOOTSTRAP> section is applied. You can run the “show route” command to verify the subnet mask.


```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Custom bootstrap example for Azure

In the following example, a new bootstrap sequence for ADC is mentioned and default bootstrap is skipped. You provide the interface details along with the IP addresses and subnet masks as follows:

- Management interface (eth2), NSIP (172.27.2.53), and subnet mask (255.255.255.0)
- Client facing interface (eth1), VIP (172.27.1.53), and subnet mask (255.255.255.0)
- Server facing interface (eth0), SNIP (172.27.0.53), and subnet mask (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 172.27.2.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

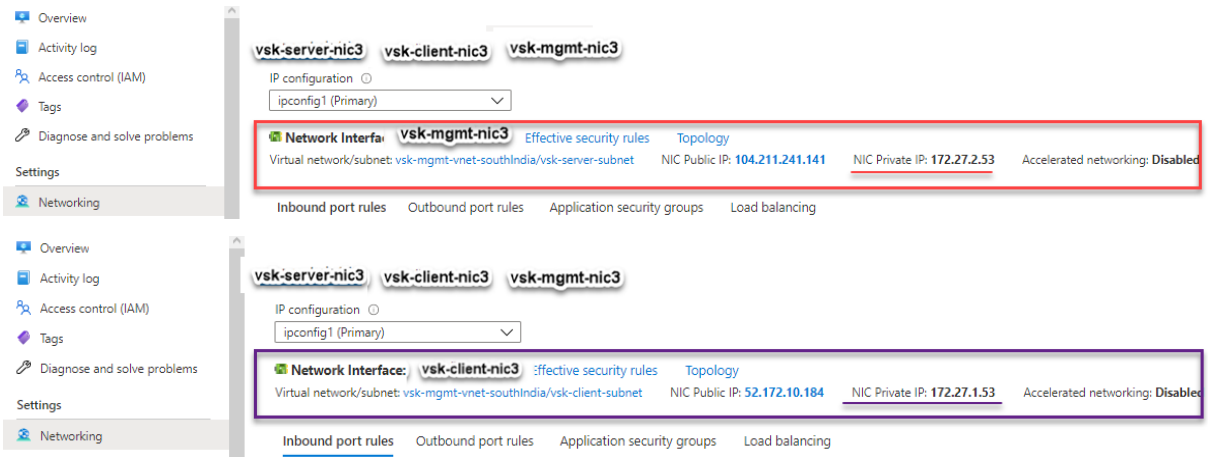
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 172.27.1.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
            </CLIENT-INTERFACE-CONFIG>

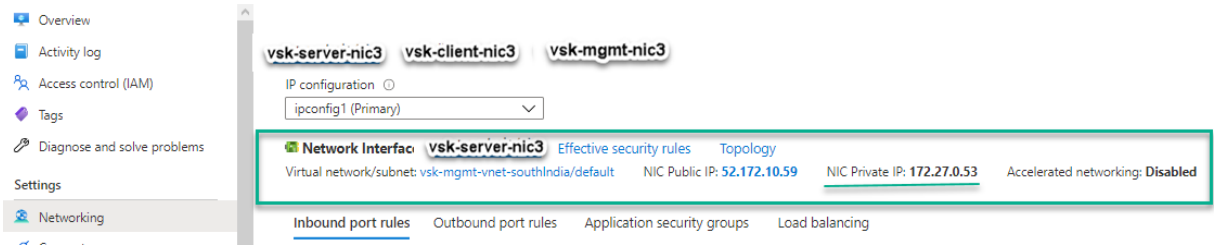
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 172.27.0.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

You can see that the Citrix ADC VPX instance is created with three network interfaces. Navigate to the **Azure portal > VM instance > Networking**, and verify the networking properties of the three NICs as shown in the following illustrations.





You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the <NS-BOOTSTRAP> section is applied. You can run the "show route" command to verify the subnet mask.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

Custom bootstrap example for GCP

In the following example, a new bootstrap sequence for ADC is mentioned and default bootstrap is skipped. You provide the interface details along with the IP addresses and subnet masks as follows:

- Management interface (eth2), NSIP (10.128.4.31), and subnet mask (255.255.255.0)
- Client facing interface (eth1), VIP (10.128.0.43), and subnet mask (255.255.255.0)
- Server facing interface (eth0), SNIP (10.160.0.75), and subnet mask (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>
        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 10.128.4.31 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 10.128.0.43 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </CLIENT-INTERFACE-CONFIG>
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 10.160.0.75 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

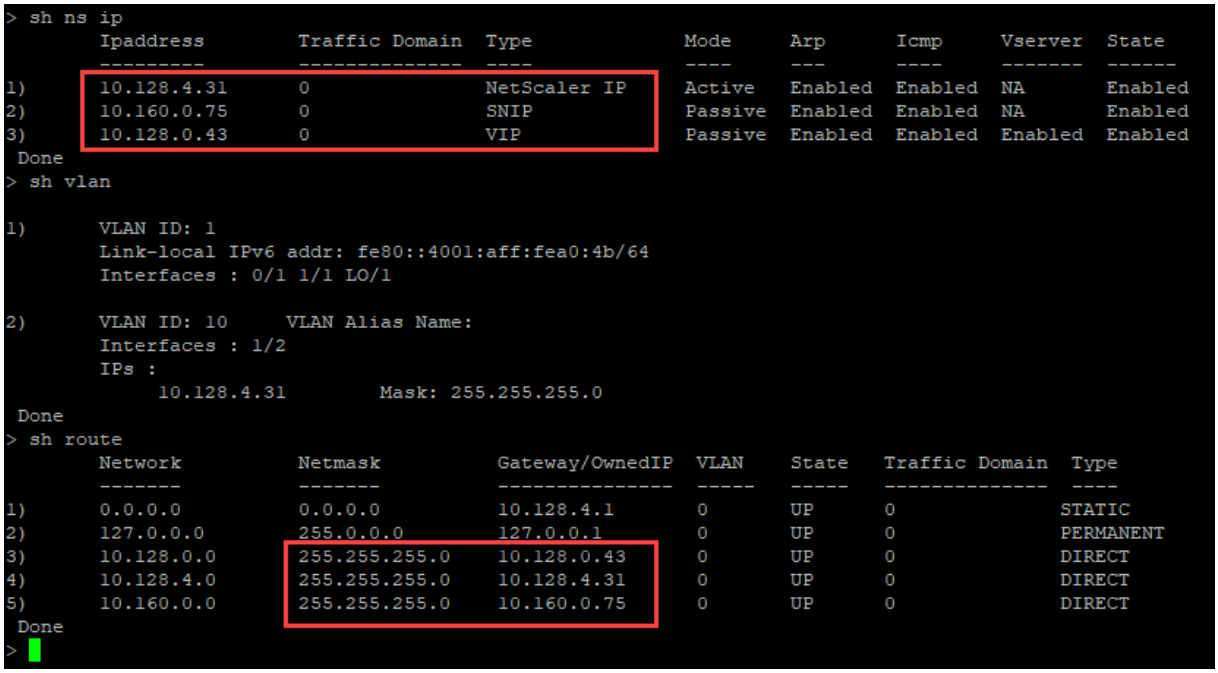
After the VM instance is created in the GCP portal with the custom bootstrap, you can verify the network interface properties as follows:

1. Select the instance that you have created by providing the custom bootstrap information.
2. Navigate to the Network interface properties and verify the NIC details as follows.

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	vsk-defnw-st-ip1 (10.160.0.75)	–	34.93.216.90 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	vsk-vpc-nw1-st-ip1 (10.128.0.43)	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	vsk-nw2-st-ip-1 (10.128.4.31)	–	34.93.202.214 (ephemeral)	Premium		View details

You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence specified in the <NS-BOOTSTRAP> section is applied. You can run the “show route” command to verify the subnet mask.



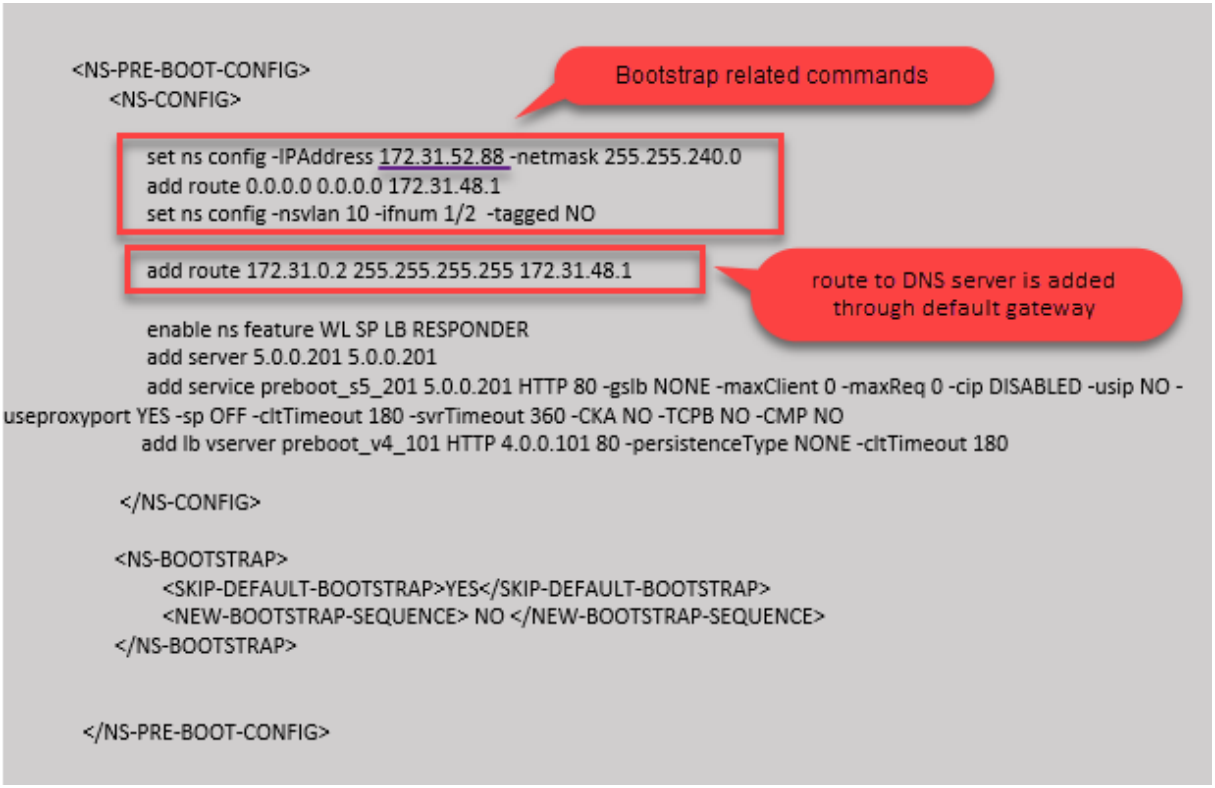
Method 3: Custom bootstrap by providing bootstrap related commands in the <NS-CONFIG> section

You can provide the bootstrap related commands in the <NS-CONFIG> section. In the <NS-BOOTSTRAP> section, you must specify the <NEW-BOOTSTRAP-SEQUENCE> as “No” to run the bootstrapping commands in the <NS-CONFIG> section. You must also provide the commands to assign NSIP, default route, and NSVLAN. In addition, provide the commands relevant for the cloud that you use.

Before providing a custom bootstrap, ensure that your cloud infrastructure supports a particular interface configuration.

Custom bootstrap example for AWS

In this example, bootstrap related commands are provided in the <NS-CONFIG> section. The <NS-BOOTSTRAP> section indicates that the default bootstrapping is skipped, and the custom bootstrap information provided in the <NS-CONFIG> section is run. You must also provide the commands to create NSIP, add default route, and add NSVLAN.



You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>

        set ns config -IPAddress 172.31.52.88 -netmask 255.255.240.0
        add route 0.0.0.0 0.0.0.0 172.31.48.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add route 172.31.0.2 255.255.255.255 172.31.48.1

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO - useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

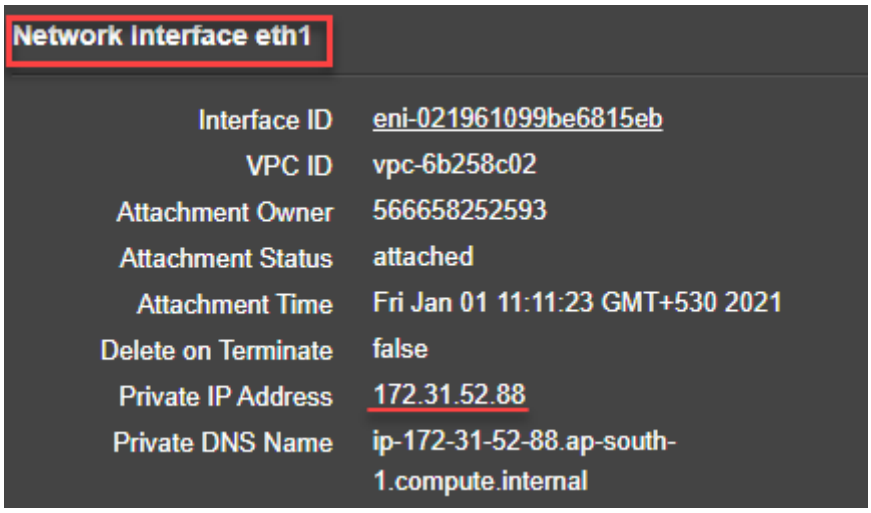
    <NS-BOOTSTRAP>
     <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
     <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
```

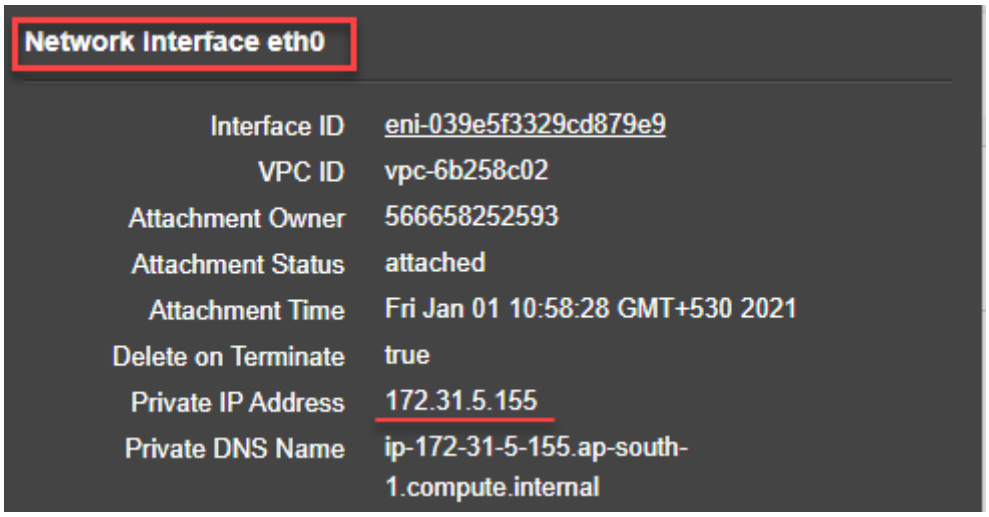


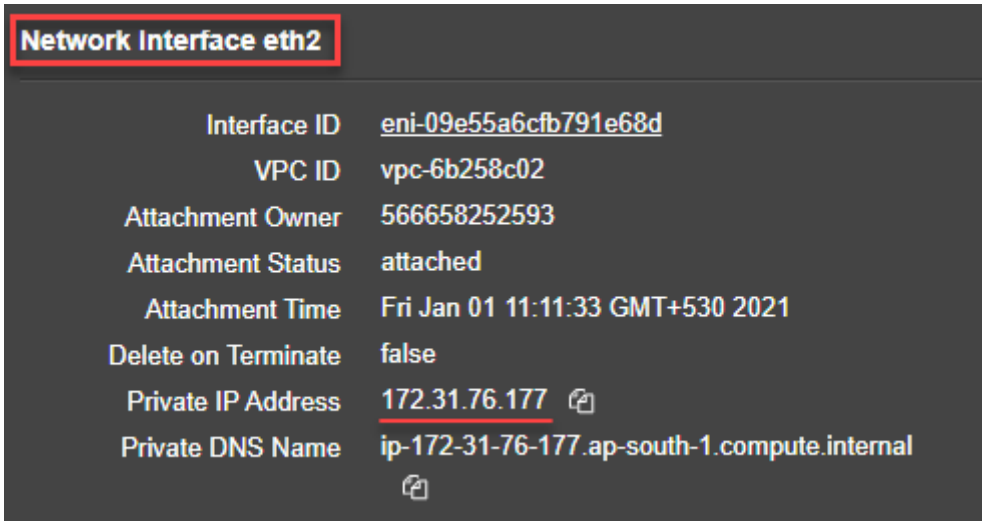
```
19      </NS-BOOTSTRAP>
20
21
22  </NS-PRE-BOOT-CONFIG>
23  <!--NeedCopy-->
```

After the VM instance is created, in the AWS portal, you can verify the network interface properties as follows:

1. Navigate to the **AWS Portal > EC2 instances**, and select the instance that you have created by providing the custom bootstrap information.
2. In the **Description** tab, you can verify the properties of each network interface as shown in the following illustrations.







You can run the `show nsip` command in **ADC CLI**, and verify the network interfaces applied to the ADC VPX instance during the first boot of the ADC appliance.

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

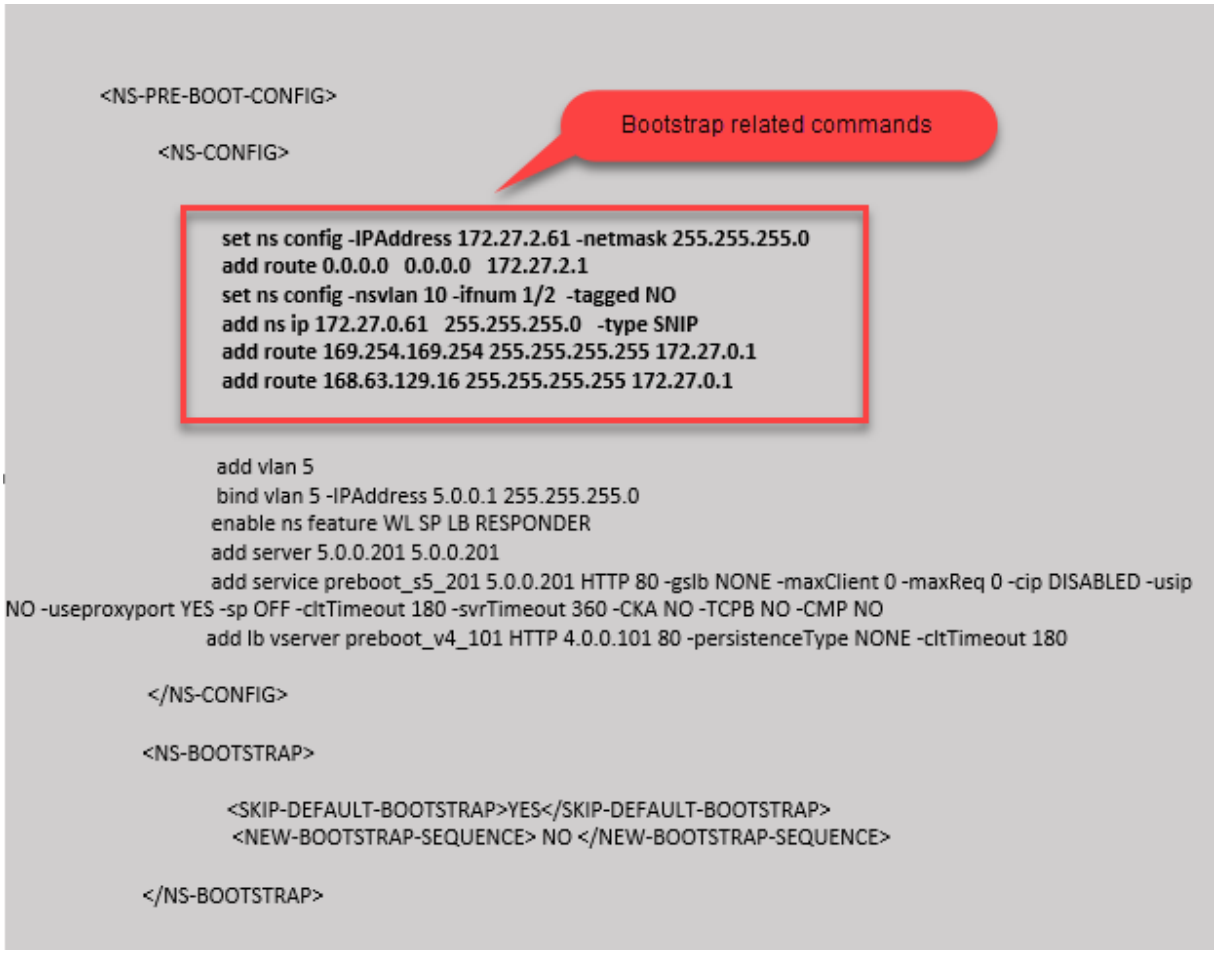
2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
4)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Custom bootstrap example for Azure

In this example, bootstrap related commands are provided in the <NS-CONFIG> section. The <NS-BOOTSTRAP> section indicates that the default bootstrapping is skipped, and the custom bootstrap information provided in the <NS-CONFIG> section is run.

> Note:
>
> For Azure cloud, Instance Metadata Server (IMDS) and DNS servers are accessible only through primary interface (Eth0). Therefore, if Eth0 interface is not used as management interface (NSIP),

> Eth0 interface must at least be configured as SNIP for IMDS or DNS access to work. The route to IMDS endpoint (169.254.169.254) and DNS endpoint (168.63.129.16) through Eth0's gateway must also be added.




```
<NS-PRE-BOOT-CONFIG>

   <NS-CONFIG>

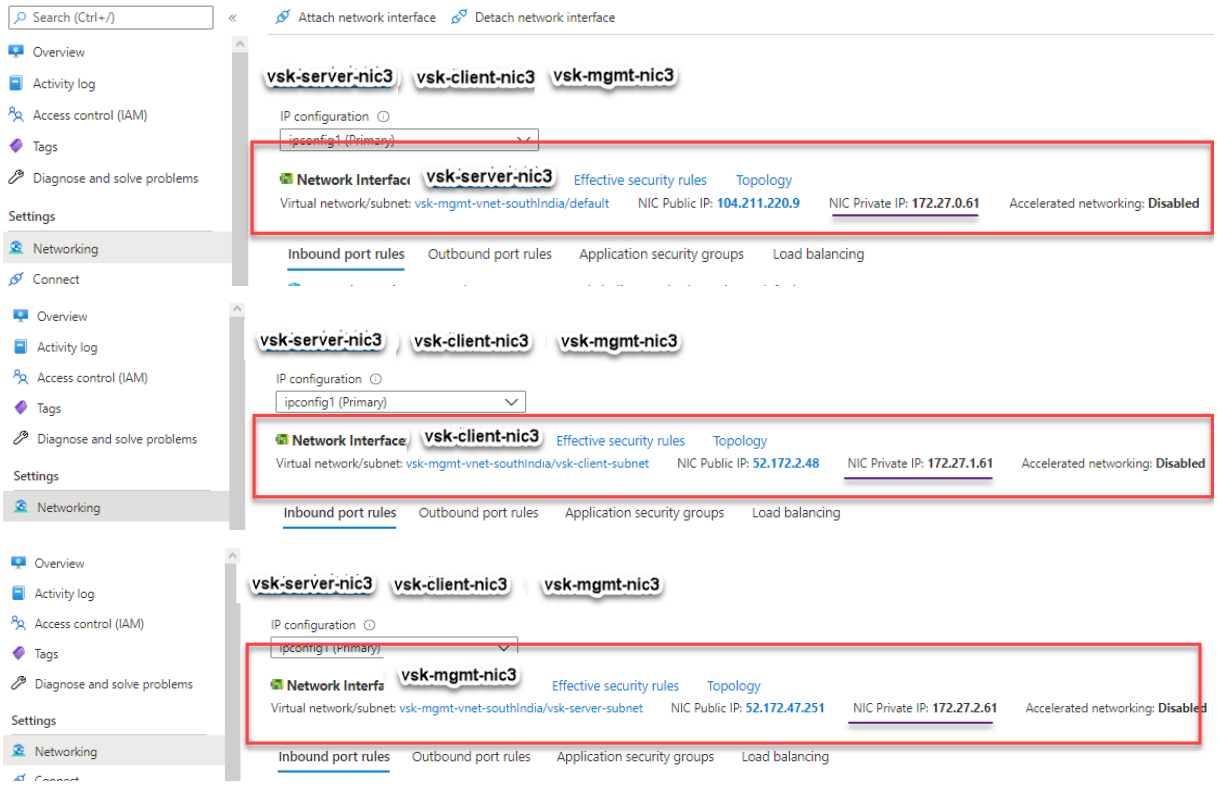
        set ns config -IPAddress 172.27.2.61 -netmask 255.255.255.0
        add route 0.0.0.0   0.0.0.0   172.27.2.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add ns ip 172.27.0.61   255.255.255.0   -type SNIP
        add route 169.254.169.254 255.255.255.255 172.27.0.1
        add route 168.63.129.16 255.255.255.255 172.27.0.1

        add vlan 5
        bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
```



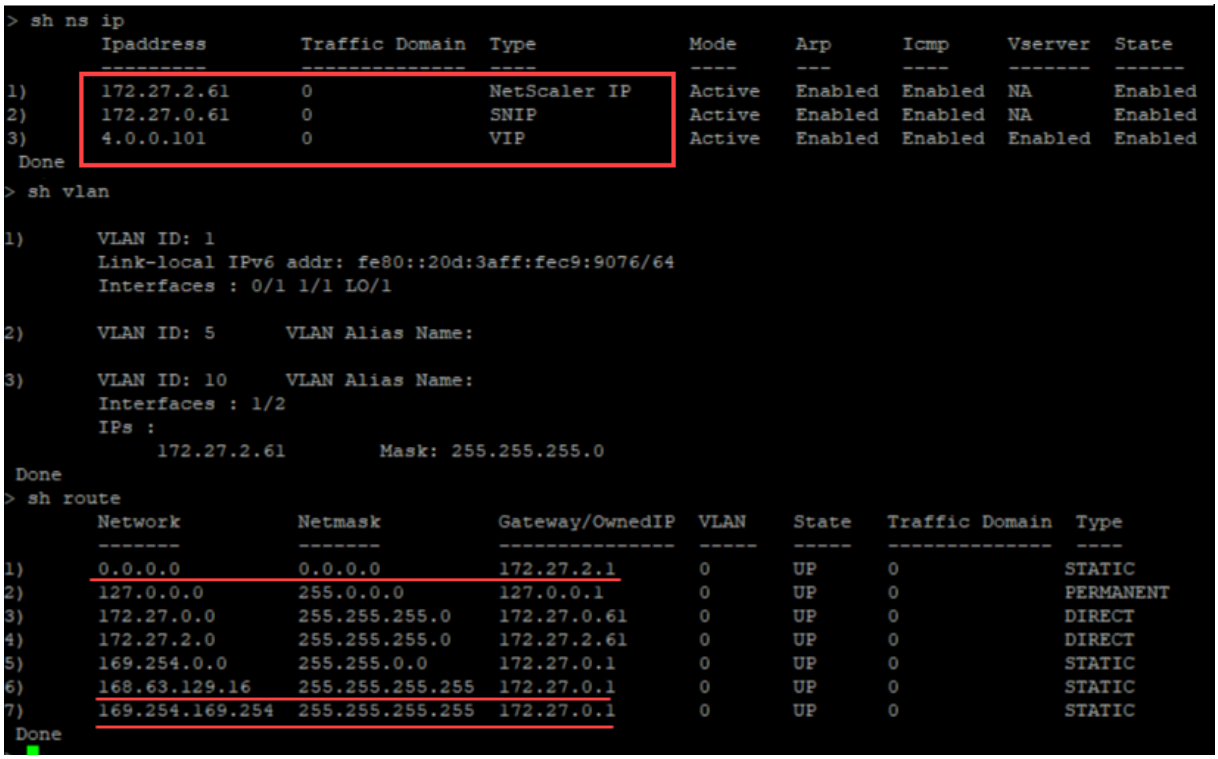
```
16          add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
                maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
                YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
                 -CMP NO
17          add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
                persistenceType NONE -cltTimeout 180
18
19      </NS-CONFIG>
20
21      <NS-BOOTSTRAP>
22
23      <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
24      <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
25
26      </NS-BOOTSTRAP>
27
28  </NS-PRE-BOOT-CONFIG>
29  <!--NeedCopy-->
```

You can see that the Citrix ADC VPX instance is created with three network interfaces. Navigate to the **Azure portal > VM instance > Networking**, and verify the networking properties of the three NICs as shown in the following illustrations.



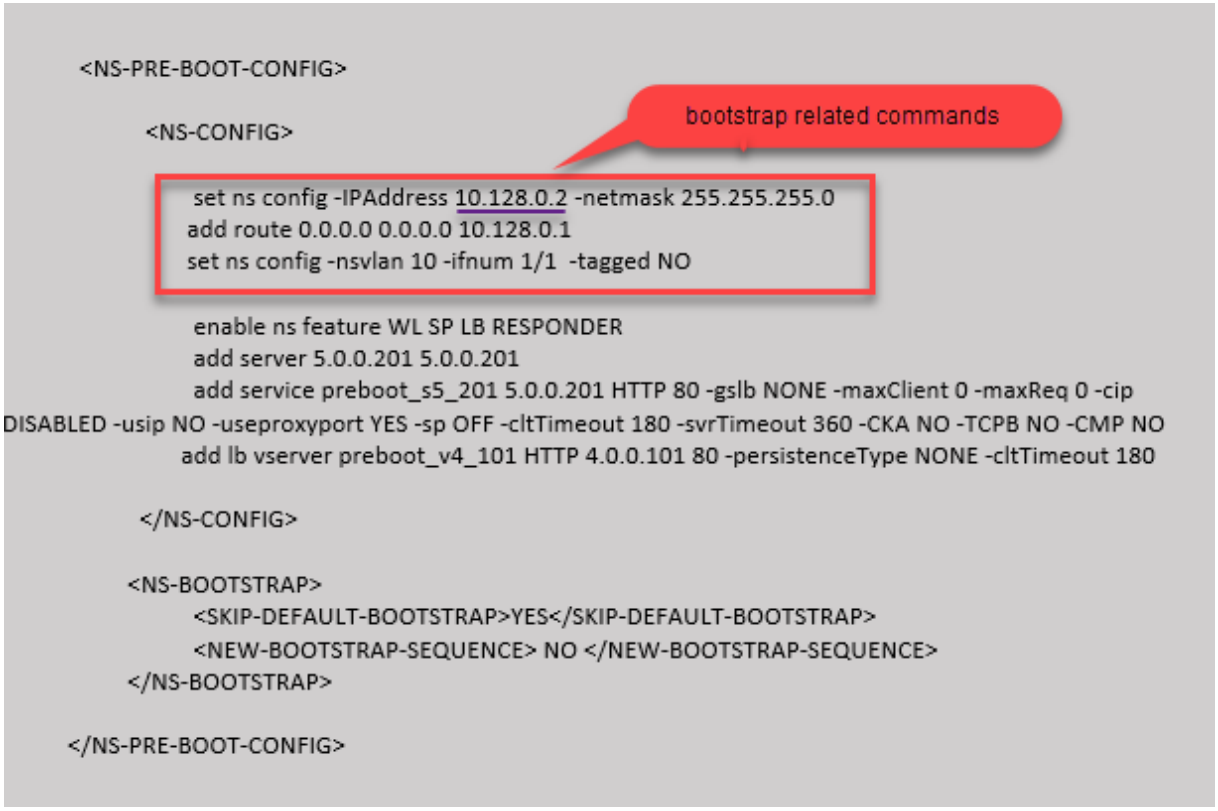
You can run the `show nsip` command in the ADC CLI, and verify that the new bootstrap sequence

specified in the <NS-BOOTSTRAP> section is applied. You can run the “show route” command to verify the subnet mask.



Custom bootstrap example for GCP

In this example, bootstrap related commands are provided in the <NS-CONFIG> section. The <NS-BOOTSTRAP> section indicates that the default bootstrapping is skipped, and the custom bootstrap information provided in the <NS-CONFIG> section is applied.



You can copy the configuration shown in the preceding screenshot from here:

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

        set ns config -IPAddress 10.128.0.2 -netmask 255.255.255.0
        add route 0.0.0.0 0.0.0.0 10.128.0.1
        set ns config -nsvlan 10 -ifnum 1/1  -tagged NO

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
```

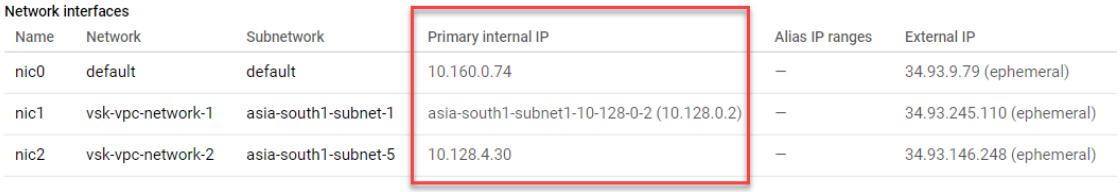


```
        <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

After the VM instance is created in the GCP portal with the custom bootstrap, you can verify the network interface properties as follows:

1. Select the instance that you have created by providing the custom bootstrap information.
2. Navigate to the Network interface properties and verify the NIC details as shown in the illustration.



You can run the `show nsip` command in **ADC CLI**, and verify that the configurations provided in the preceding <NS-CONFIG> section are applied at the first boot of the ADC appliance.

```
![Show NSIP output](/en-us/citrix-adc/media/gcp-show-nsip-method3.png)
```

Impact of attaching and detaching NICs in AWS and Azure

AWS and Azure provide the option to attach a network interface to an instance, and detach a network interface from an instance. Attaching or detaching interfaces might alter interface positions. Hence, Citrix recommends you to refrain from detaching interfaces from the ADC VPX instance. If you detach or attach an interface when custom bootstrapping is configured, Citrix ADC VPX instance reassigns the primary IP of the newly available interface in the management interface's position as NSIP. If no further interfaces are available after the one you detached, then the first interface is made the management interface for the ADC VPX instance.

For example, a Citrix ADC VPX instance is brought up with 3 interfaces: Eth0 (SNIP), Eth1 (NSIP), and Eth2 (VIP). If you detach Eth1 interface from the instance, which is a management interface, ADC configures the next available interface (Eth2) as the management interface. Thereby, the ADC VPX instance is still accessed through the primary IP of Eth2 interface. If Eth2 is also not available, then the remaining interface (Eth0) is made the management interface. Therefore, the access to ADC VPX instance continues to exist.

Let's consider a different assignment of interfaces as follows: Eth0 (SNIP), Eth1 (VIP), and Eth2 (NSIP). If you detach Eth2 (NSIP), because no new interface is available after Eth2, the first interface (Eth0) is made the management interface.

Install a Citrix ADC VPX instance on a bare metal server

September 14, 2021

A bare metal is a fully dedicated physical server that delivers physical isolation, fully integrated into the cloud environment. It is also known as a single-tenant server. Single tenancy allows you to avoid the noisy neighbor effect. With bare metal, you do not witness the noisy neighbor effect because you are the sole user.

A bare metal server installed with a hypervisor provides you a management suite to create virtual machines on the server. The hypervisor does not run applications natively. Its purpose is to virtualize your workloads into separate virtual machines to gain the flexibility and reliability of virtualization.

Prerequisites for installing Citrix ADC VPX instance on bare metal servers

A bare metal server must be obtained from a cloud vendor that meets all the system requirements for the respective hypervisor.

Install the Citrix ADC VPX instance on bare metal servers

To install Citrix ADC VPX instances on a bare metal server, you must first obtain a bare metal server with adequate system resources from a cloud vendor. On that bare metal server, any of the supported hypervisors such as Linux KVM, VMware ESX, Citrix Hypervisor, or Microsoft Hyper-V must be installed and configured before deploying the ADC VPX instance.

For more information on the list of different hypervisors and features supported on a Citrix ADC VPX instance, see Support matrix and usage guidelines.

For more information on installing Citrix ADC VPX instances on different hypervisors, see the respective documentation.

- **Citrix Hypervisor:** See Install a Citrix ADC VPX instance on Citrix Hypervisor.
- **VMware ESX:** See Install a Citrix ADC VPX instance on VMware ESX.
- **Microsoft Hyper-V:** See Install a Citrix ADC VPX instance on Microsoft Hyper-V server.
- **Linux KVM platform:** See Install a Citrix ADC VPX instance on Linux-KVM platform.

Install a Citrix ADC VPX instance on Citrix Hypervisor

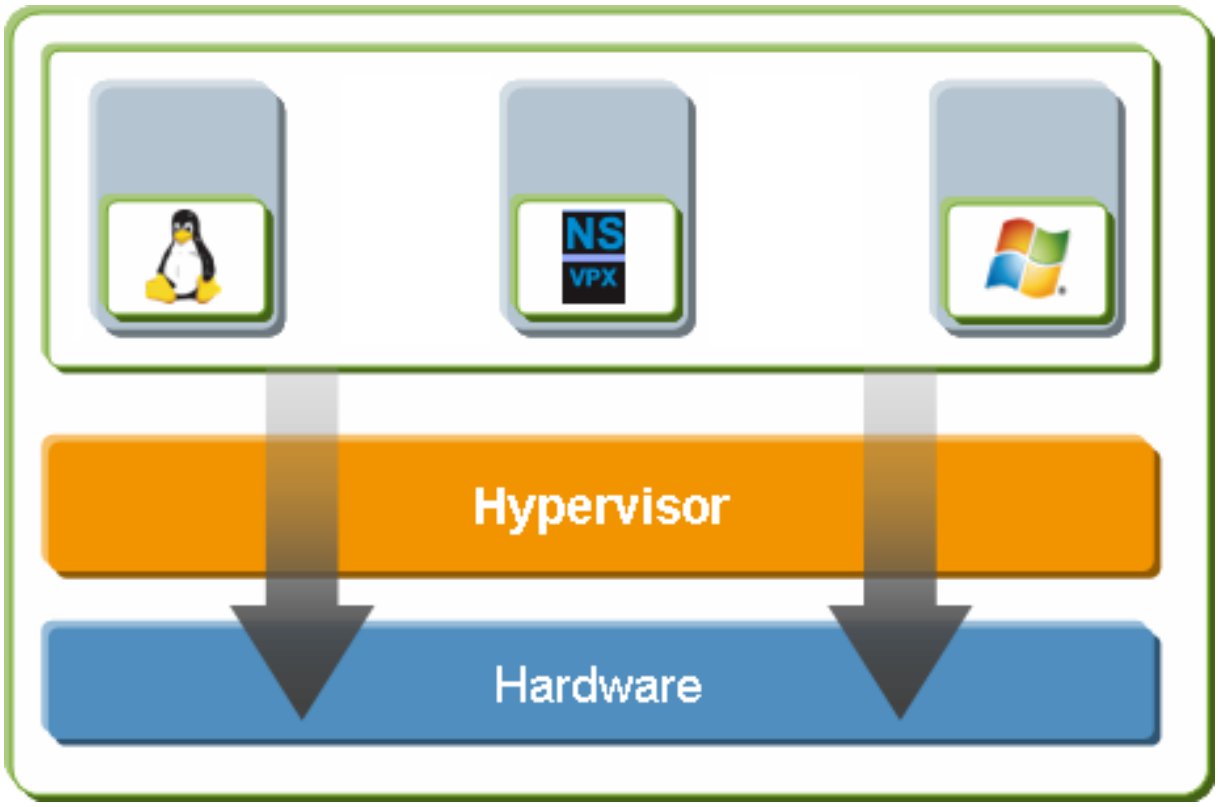
September 14, 2021

To install VPX instances on the Citrix Hypervisor, you must first install the Hypervisor on a machine with adequate system resources. To perform the Citrix ADC VPX instance installation, you use Citrix XenCenter, which must be installed on a remote machine that can connect to the Hypervisor host through the network.

For more information about Hypervisor, see Citrix Hypervisor documentation.

The following figure shows the bare-metal solution architecture of Citrix ADC VPX instance on Hypervisor.

Figure. A Citrix ADC VPX instance on Citrix Hypervisor



Prerequisites for installing a Citrix ADC VPX instance on Hypervisor

Before you begin installing a virtual appliance, do the following:

- Install Hypervisor version 6.0 or later on hardware that meets the minimum requirements.
- Install XenCenter on a management workstation that meets the minimum system requirements.
- Obtain virtual appliance license files. For more information about virtual appliance licenses, see the *Citrix ADC VPX Licensing Guide* at http://support.citrix.com/article/ctx122426.

Hypervisor hardware requirements

The following table describes the minimum hardware requirements for a Hypervisor platform running a Citrix ADC VPX instance.

Table 1. Minimum system requirements for Hypervisor running a nCore VPX instance

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT) enabled. AMD processor is not supported. To run the Citrix ADC VPX instance, hardware support for virtualization must be enabled on the Hypervisor host. Make sure that the BIOS option for virtualization support is not disabled. For more details, see BIOS documentation.
RAM	3 GB
Disk space	Locally attached storage (PATA, SATA, SCSI) with 40 GB of disk space. Note: Hypervisor installation creates a 4 GB partition for the Hypervisor host control domain. The remaining space is available for Citrix ADC VPX instance and other virtual machines.
NIC	One 1-Gbps NIC; recommended: two 1-Gbps NICs

For information about installing Hypervisor, see the Hypervisor documentation at http://support.citrix.com/product/xens/.

The following table lists the virtual computing resources that Hypervisor must provide for each nCore VPX virtual appliance.

Table 2. Minimum virtual computing resources required for running a nCore VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	2

> Note: For production use of
> Citrix ADC VPX instance, Citrix recommends that CPU priority (in virtual machine properties) be set to the highest level, to improve scheduling behavior and network latency.

XenCenter system requirements

XenCenter is a Windows client application. It cannot run on the same machine as the Hypervisor host. For more information about minimum system requirements and installing XenCenter, see the following Hypervisor documents:

- System requirements
- Install

Install Citrix ADC VPX instances on Hypervisor by using XenCenter

After you have installed and configured Hypervisor and XenCenter, you can use XenCenter to install virtual appliances on Hypervisor. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running Hypervisor.

After you have used XenCenter to install the initial Citrix ADC VPX instance (.xva image) on Hypervisor, you can use Command Center to provision Citrix ADC VPX instance. For more information, see the Command Center documentation.

To install Citrix ADC VPX instances on Hypervisor by using XenCenter, follow these steps:

1. Start XenCenter on your workstation.
2. On the Server menu, click Add.
3. In the Add New Server dialog box, in the host name text box, type the IP address or DNS name of the Hypervisor that you want to connect to.
4. In the User Name and Password text boxes, type the administrator credentials, and then click Connect. The Hypervisor name appears in the navigation pane with a green circle, which indicates that the Hypervisor is connected.
5. In the navigation pane, click the name of the Hypervisor on which you want to install the Citrix ADC VPX instance.
6. On the VM menu, click Import.
7. In the Import dialog box, in the Import file name, browse to the location at which you saved the Citrix ADC VPX instance .xva image file. Make sure that the Exported VM option is selected, and then click Next.
8. Select the Hypervisor on which you want to install the virtual appliance, and then click Next.

9. Select the local storage repository in which to store the virtual appliance, and then click Import to begin the import process.
10. You can add, modify, or delete the virtual network interfaces as required. When finished, click Next.
11. Click Finish to complete the import process.

    Note: To view the status of the import process, click the **Log** tab.
12. If you want to install another virtual appliance, repeat steps 5 through 11.

> **Note**
>
> After the initial configuration of the VPX instance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Configure VPX instances to use single root I/O virtualization (SR-IOV) network interfaces

September 14, 2021

After you have installed and configured a Citrix ADC VPX instance on XenServer, you can configure the virtual appliance to use SR-IOV network interfaces.

Limitations

XenServer does not support the following features on SRIOV interfaces:

- L2 mode switching
- Clustering
- Admin partitioning [Shared VLAN mode]
- High Availability [Active - Active mode]
- Jumbo frames
- IPv6 protocol in Cluster environment

Prerequisites

On the XenServer host, ensure that you:

- Add the Intel 82599 NIC (NIC) to the host.
- Block list the `ixgbevf` driver by adding the following entry to the **/etc/modprobe.d/blacklist.conf** file:

blacklist ixgbevf

- Enable SR-IOV Virtual Functions (VFs) by adding the following entry to the **/etc/modprobe.d/ixgbe** file:

  options ixgbe max_vfs=<*number_of_VFs*>

  where <*number_VFs*> is the number of SR-IOV VFs that you want to create.

- Verify that SR-IOV is enabled in BIOS.

IXGBE driver version 3.22.3 is recommended.

Assign SR-IOV VFs to the VPX instance by using the XenServer host

To assign SR-IOV network interfaces to Citrix ADC VPX instance, follow these steps:

1. On the XenServer host, use the following command to assign the SR-IOV VFs to the Citrix ADC VPX instance:

   xe host-call-plugin plugin=iovirt host-uuid=<*Xen host UUID*> **fn=assign_free_vf args:uuid**=<*Netscalar VM UUID*> **args:ethdev**=<*interface name*> **args:mac=**<*Mac addr*>

   Where:

   - <*Xen host UUID*> is the UUID of the XenServer host.
   - <*NetScaler VM UUID*> is the UUID of the Citrix ADC VPX instance.
   - <*interface name*> is the interface for the SR-IOV VFs.
   - <*MAC address* > is the MAC address of the SR-IOV VF.

   > **Note**
   >
   > Specify the MAC address that you want use in the args:Mac= parameter, if not specified, the `iovirt` script randomly generates and assigns a MAC address. Also, if you want to use the SR-IOV VFs in Link Aggregation mode, make sure that you specify the MAC address as 00:00:00:00:00:00.

2. Boot the Citrix ADC VPX instance.

Unassign SR-IOV VFs to the VPX instance by using the XenServer host

If you have assigned an incorrect SR-IOV VFs or if you want to modify an assigned SR-IOV VFs, you need to unassign and reassign the SR-IOV VFs to the Citrix ADC VPX instance.

To unassign SR-IOV network interface assigned to a Citrix ADC VPX instance, follow these steps:

1. On the XenServer host, use the following command to assign the SR-IOV VFs to the Citrix ADC VPX instance and reboot the Citrix ADC VPX instance:

xe host-call-plugin plugin=iovirt **host-uuid**=<*Xen_host_UUID*> **fn**=unassign_all **args:uuid**=<*Netscalar_VM_*

Where:

- <*Xen_host_UUID*> - The UUID of the XenServer host.
- <*Netscalar_VM_UUID*> - The UUID of the Citrix ADC VPX instance

2. Boot the Citrix ADC VPX instance.

Configuring VLAN on the SR-IOV Interface

> **Important**
>
> While you are assigning the SR-IOV VFs to the Citrix ADC VPX instance, make sure that you specify MAC address 00:00:00:00:00:00 for the VFs.

To use the SR-IOV virtual functions in link aggregation mode, you need to disable spoof checking for virtual functions that you have created. On the XenServer host, use the following command to disable spoof checking:

ip link set <*interface_name*> **vf** <*VF_id*> **spoofchk** off

Where:

- <interface_name> is the interface name.
- <VF_id> is the virtual function ID.

After disabling spoof checking for all the Virtual Function that you have created, restart the Citrix ADC VPX instance and configure link aggregation. For instructions, see Configure link aggregation.

Configure VLAN on the SR-IOV interface

You can configure VLAN on the SR-IOV Virtual Functions, for instructions, see Configuring a VLAN.

> **Important**
>
> Make sure that the XenServer host does not contain VLAN settings for the VF interface.

Install a Citrix ADC VPX instance on VMware ESX

September 28, 2021

Before installing Citrix ADC VPX instances on VMware ESX, make sure that VMware ESX Server is installed on a machine with adequate system resources. To install a Citrix ADC VPX instance on VMware ESXi, you use the VMware vSphere client. The client or tool must be installed on a remote machine that can connect to VMware ESX through the network.

This section includes the following topics:

- Prerequisites
- Installing a Citrix ADC VPX instance on VMware ESX

> **Important**
>
> You cannot install standard VMware Tools or upgrade the VMware Tools version available on a Citrix ADC VPX instance. VMware Tools for a Citrix ADC VPX instance are delivered as part of the Citrix ADC software release.

Prerequisites

Before you begin installing a virtual appliance, do the following:

- Install VMware ESX on hardware that meets the minimum requirements.
- Install VMware Client on a management workstation that meets the minimum system requirements.
- Download the Citrix ADC VPX appliance setup files.
- Label the physical network ports of VMware ESX.
- Obtain VPX license files. For more information about Citrix ADC VPX instance licenses, see the *Citrix ADC VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

VMware ESX hardware requirements

The following table describes the minimum system requirements for VMware ESX servers running Citrix ADC VPX nCore virtual appliance.

Table 1. Minimum system requirements for a VMware ESX server running a Citrix ADC VPX instance

Component	Requirement
-	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT) enabled. To run Citrix ADC VPX instance, hardware support for virtualization must be enabled on the VMware ESX host. Make sure that the BIOS option for virtualization support is not disabled. For more information, see your BIOS documentation.
RAM	3 GB
Disk space	40 GB of disk space available

Component	Requirement
Network	One 1-Gbps NIC (NIC); Two 1-Gbps NICs recommended

For information about installing VMware ESX, see http://www.vmware.com/.

The following table lists the virtual computing resources that the VMware ESX server must provide for each VPX nCore virtual appliance.

Table 2. Minimum virtual computing resources required for running a Citrix ADC VPX instance

Component	Requirement
Memory	4 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	1. In ESX, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher.
Disk space	20 GB

> Note
>
> This is in addition to any disk requirements for the hypervisor.

For production use of VPX virtual appliance, the full memory allocation must be reserved. CPU cycles (in MHz) equal to at least the speed of one CPU core of the ESX must be reserved.

VMware vSphere client system requirements

VMware vSphere is a client application that can run on Windows and Linux operating systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 3. Minimum system requirements for VMware vSphere client installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the “vSphere Compatibility Matrixes” PDF file at http://kb.vmware.com/.

Component	Requirement
CPU	750 MHz; 1 gigahertz (GHz) or faster recommended
RAM	1 GB; 2 GB recommended
NIC (NIC)	100 Mbps or faster NIC

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 4. Minimum system requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
NIC (NIC)	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.

Downloading the Citrix ADC VPX setup files

The Citrix ADC VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com, click the **New Users link**, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

Citrix.com > **Downloads > Citrix ADC > Virtual Appliances**.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-71.44_nc_64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-71.44_nc_64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-71.44_nc_64.mf)

Label the physical network ports of VMware ESX

Before installing a VPX virtual appliance, label of all the interfaces that you plan to assign to virtual appliances, in a unique format, for example, NS_NIC_1_1, NS_NIC_1_2, and so on. In large deployments, labeling in a unique format helps in quickly identifying the interfaces that are allocated to the VPX virtual appliance among other interfaces used by other virtual machines, such as Windows and Linux. Such labeling is especially important when different types of virtual machines share interfaces.

To label the physical network ports of VMware ESX server, follow these steps:

1. Log on to the VMware ESX server by using the vSphere client.
2. On the vSphere client, select the Configuration tab, and then click Networking.
3. At the top-right corner, click Add Networking.
4. In the Add Network Wizard, for **Connection Type**, select **Virtual Machine**, and then click Next.
5. Scroll through the list of vSwitch physical adapters, and choose the physical port that maps to interface 1/1 on the virtual appliances.
6. Enter the label of the interface, for example, **NS_NIC_1_1** as the name of the vSwitch that is associated with interface 1/1 of the virtual appliances.
7. Click Next to finish the vSwitch creation. Repeat the procedure, beginning with step 2, to add any additional interfaces to be used by your virtual appliances. Label the interfaces sequentially, in the correct format (for example, NS_NIC_1_2).

Install a Citrix ADC VPX instance on VMware ESX

After you have installed and configured VMware ESX, you can use the VMware vSphere client to install virtual appliances on the VMware ESX server. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running VMware ESX.

To install Citrix ADC VPX instances on VMware ESX by using VMware vSphere Client, follow these steps:

1. Start the VMware vSphere client on your workstation.
2. In the **IP address / Name** text box, type the IP address of the VMware ESX server that you want to connect to.
3. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click Login.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the Citrix ADC VPX instance setup files, select the .ovf file, and click **Next**.

6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the ESX host. Click **Next** to start installing a virtual appliance on VMware ESX. When installation is complete, a pop-up window informs you of the successful installation.
7. You are now ready to start the Citrix ADC VPX instance. In the navigation pane, select the Citrix ADC VPX instance that you have installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.
8. If you want to install another virtual appliance, repeat from Step 6.

> **Note**
>
> By default, the Citrix ADC VPX instance uses E1000 network interfaces.
>
> After the installation, you can use vSphere client or vSphere Web Client to manage virtual appliances on VMware ESX.
>
> For the VLAN tagging feature to work, on the VMware ESX, set the port group's VLAN ID to All (4095) on the vSwitch of VMware ESX server. For more information about setting a VLAN ID on the vSwitch of VMware ESX server, see http://www.vmware.com/pdf/esx3_vlan_wp.pdf.

Migrate a Citrix ADC VPX instance by using VMware vMotion

You can migrate a Citrix ADC VPX instance by using VMware vSphere vMotion.

Follow these usage guidelines:

- VMware does not support the vMotion feature on virtual machines configured with PCI Passthrough and SR-IOV interfaces.
- Supported interfaces are E1000 and VMXNET3. To use vMotion on your VPX instance, ensure that the instance is configured with a supported interface.
- For more information about how to migrate an instance by using VMware vMotion, see the VMware documentation.

Configure a Citrix ADC VPX instance to use VMXNET3 network interface

September 14, 2021

After you have installed and configured the Citrix ADC VPX instance on the VMware ESX, you can use the VMware vSphere web client to configure the virtual appliance to use VMXNET3 network interfaces.

To configure Citrix ADC VPX instances to use VMXNET3 network interfaces by using the VMware vSphere Web Client:

1. In the vSphere Web Client, select Hosts and Clusters.

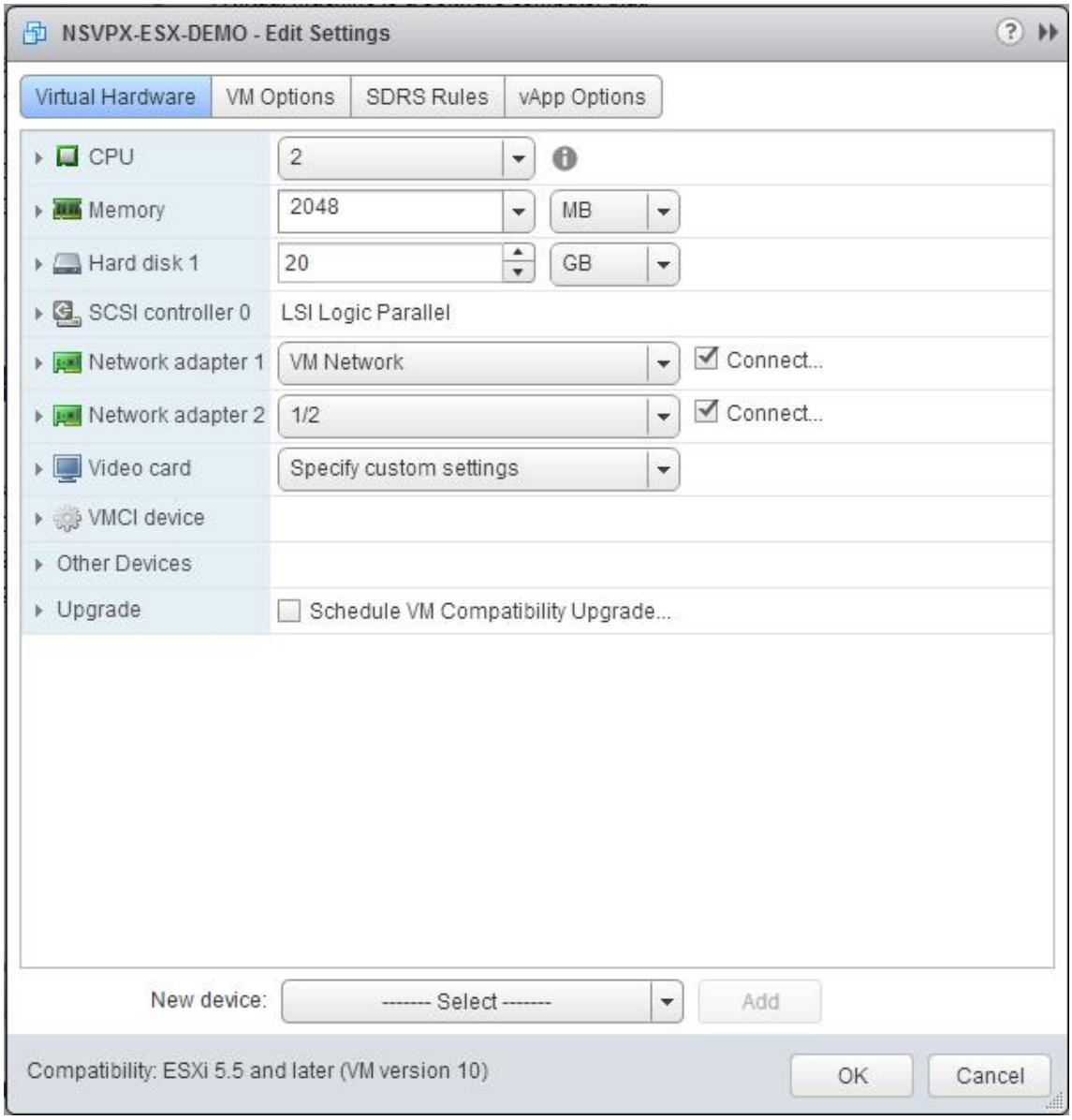
2. Upgrade the Compatibility setting of the Citrix ADC VPX instance to ESX, as follows:

   a. Power off the Citrix ADC VPX instance.

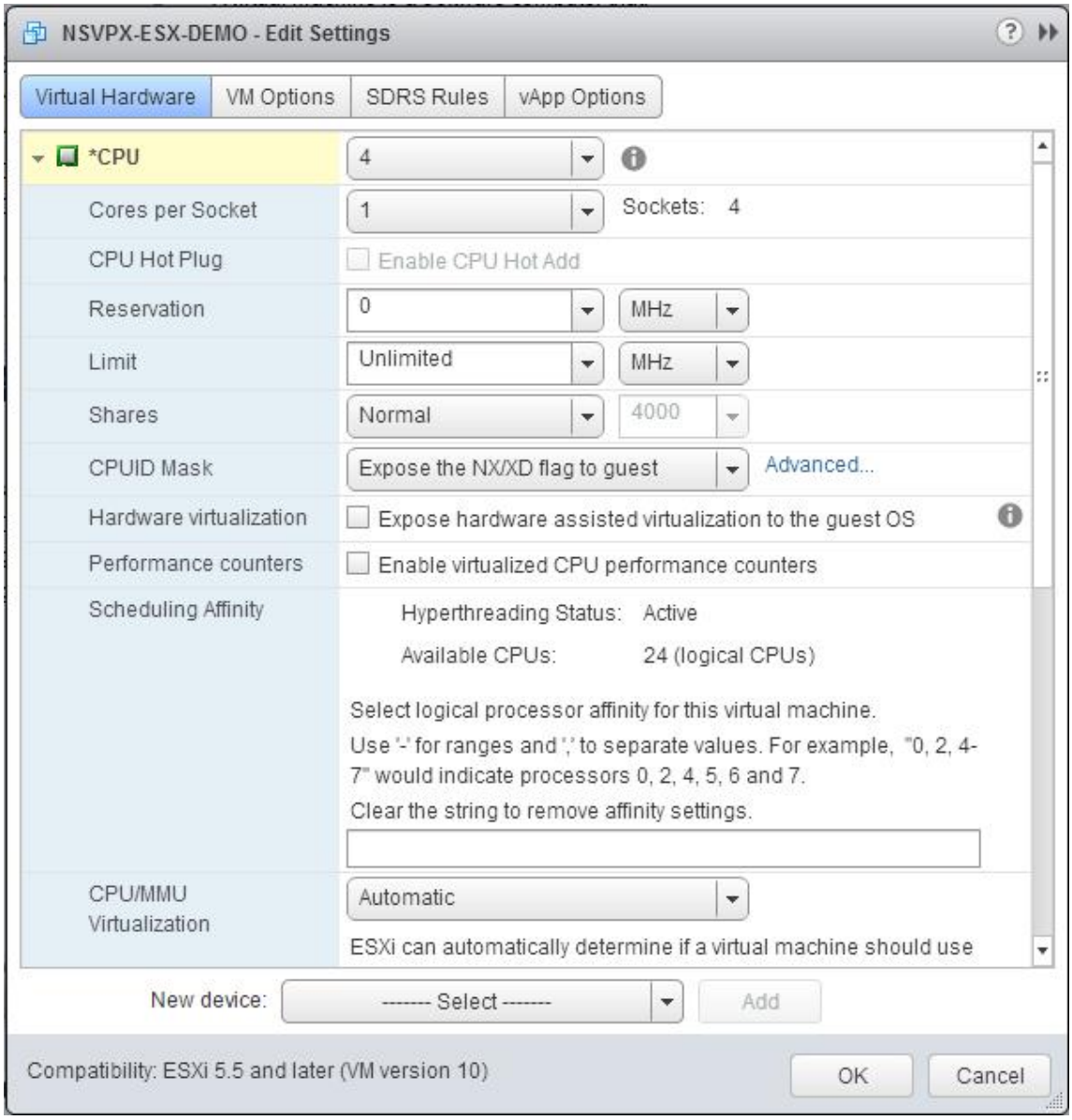
   b. Right-click the Citrix ADC VPX instance and select Compatibility > Upgrade VM Compatibility.

   c. In the Configure VM Compatibility dialog box, select ESXi 5.5 and later from the Compatible with drop-down list and click OK.

3. Right-click on the Citrix ADC VPX instance and click Edit Settings.



4. In the <virtual_appliance> - Edit Settings dialog box, click the CPU section.



5. In the CPU section, update the following:

   - Number of CPUs
   - Number of Sockets
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

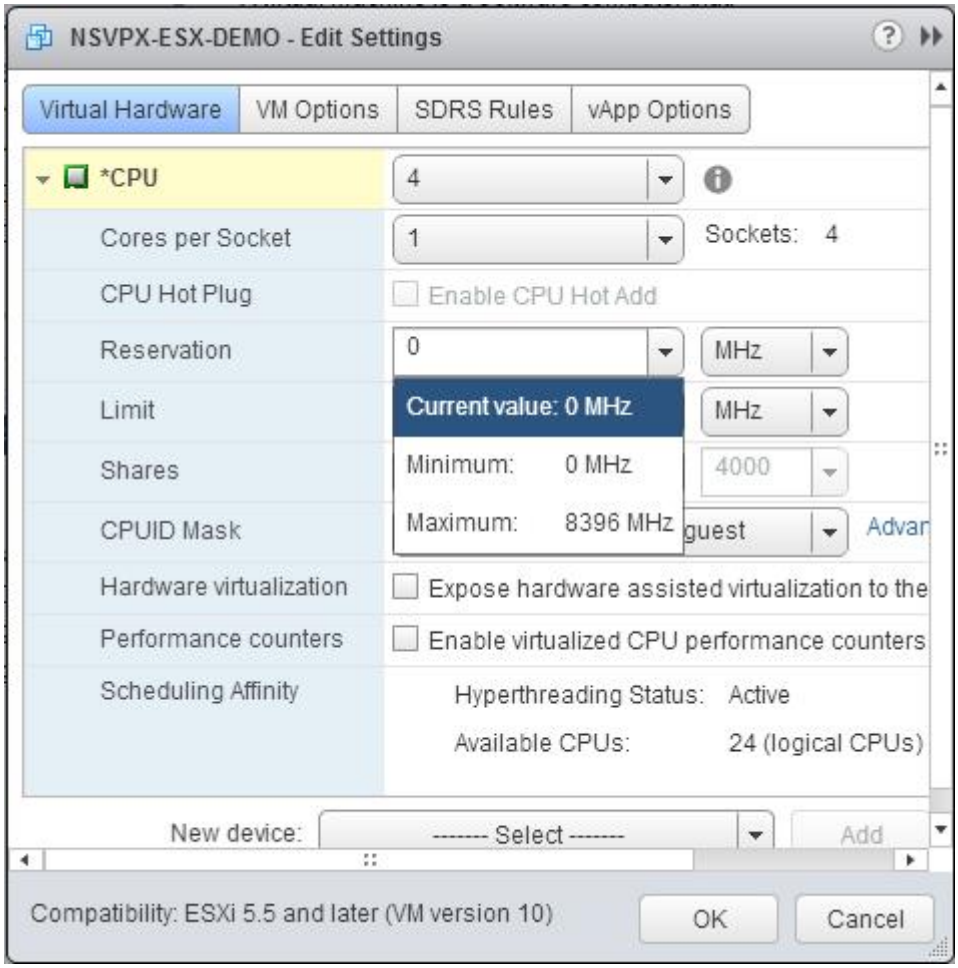
   a. In the CPU drop-down list, select the number of CPUs to assign to the virtual appliance.

   b. In the Cores per Socket drop-down list, select the number of sockets.

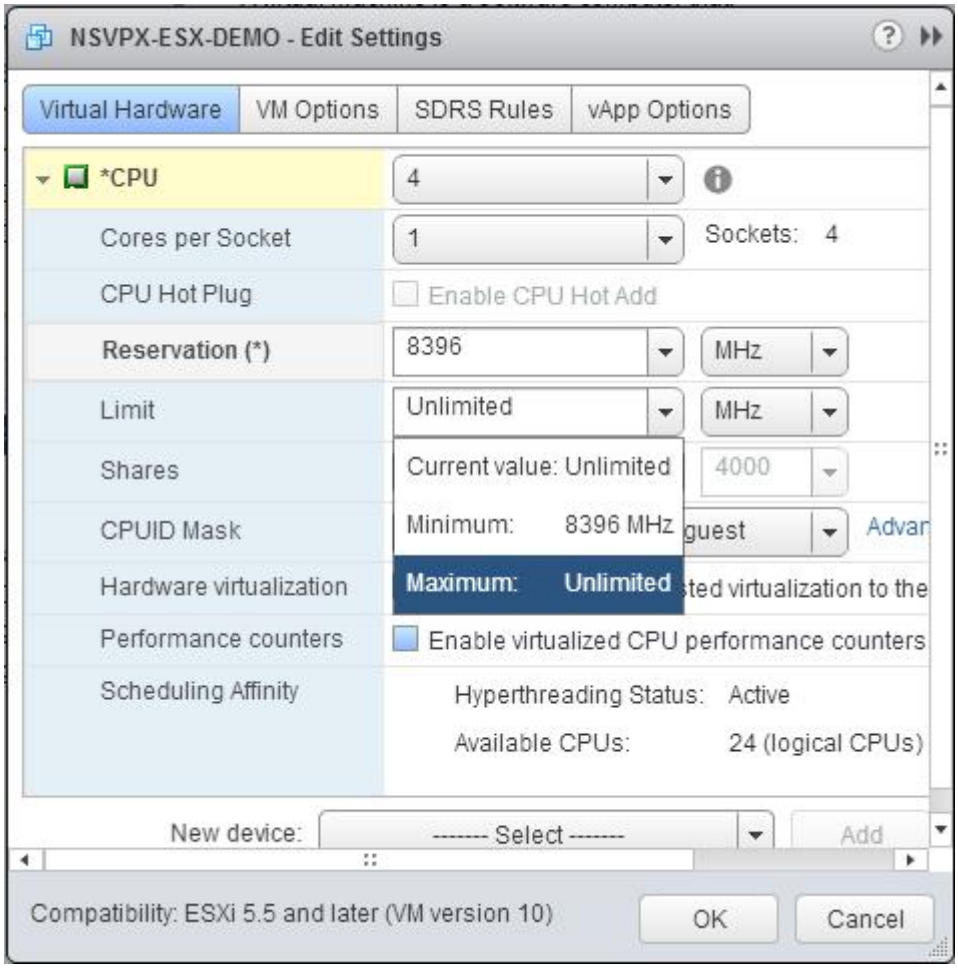
   c. (Optional) In the CPU Hot Plug field, select or unselect the Enable CPU Hot Add check box.

Note: Citrix recommends accepting the default (disabled).

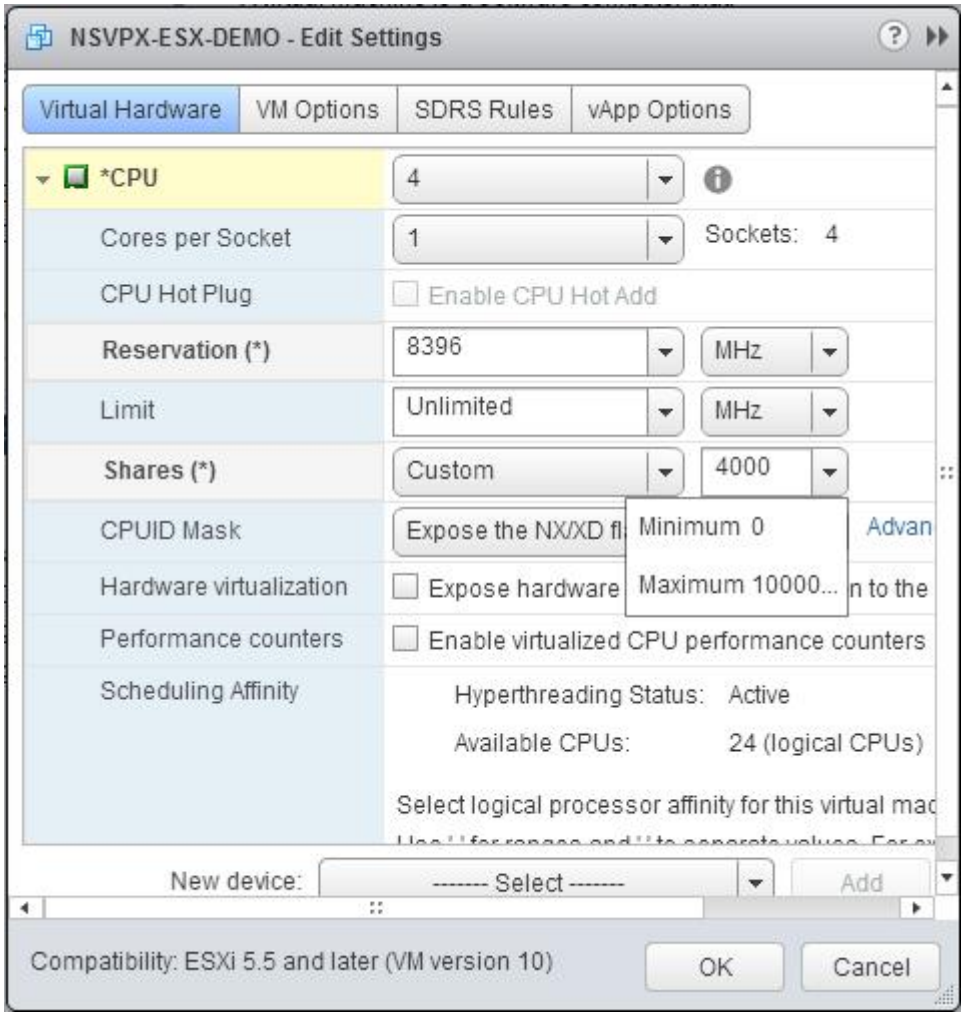
d. In the Reservation drop-down list, select the number that is shown as the maximum value.



e. In the Limit drop-down list, select the number that is shown as the maximum value.



f. In the Shares drop-down lists, select Custom and the number that is shown as the maximum value.



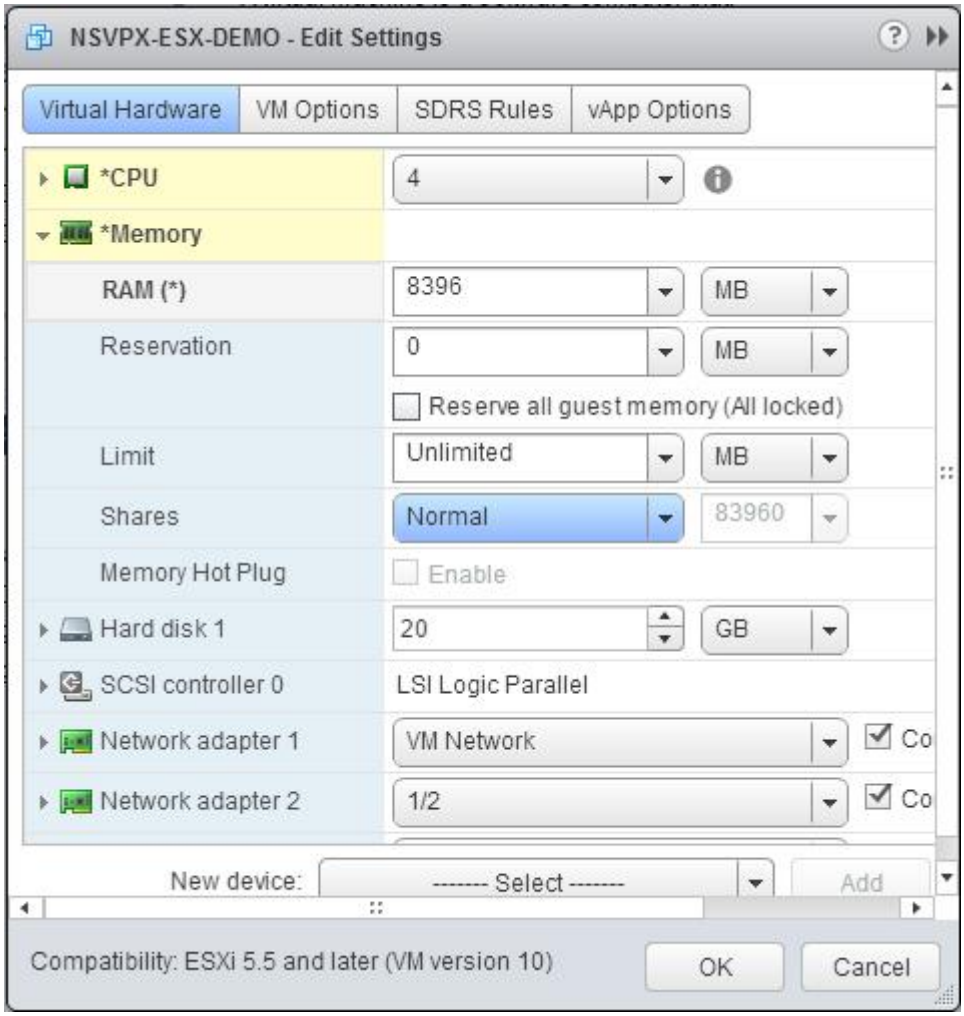
6. In the Memory section, update the following:

   - Size of RAM
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

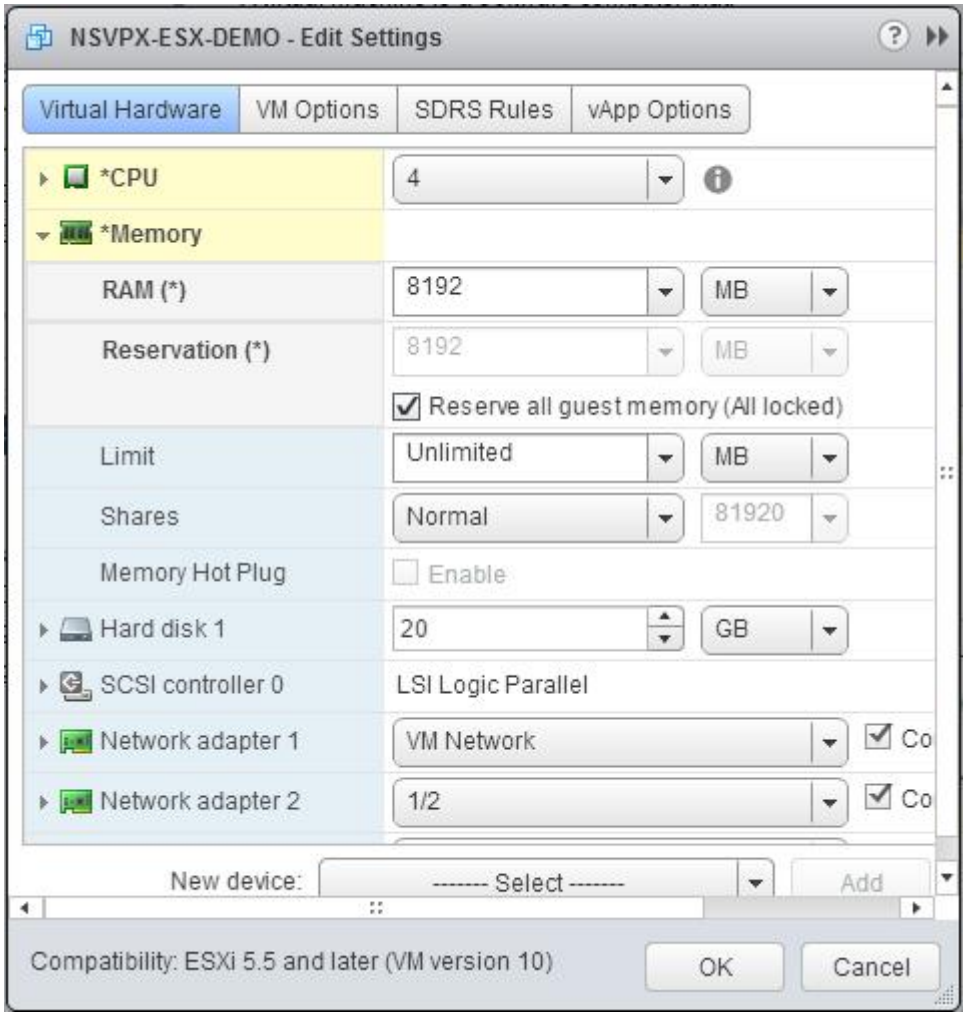
   a. In the RAM drop-down list, select the size of the RAM. It must be the number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the RAM must be 4 x 2 GB = 8 GB.

   Note: For an Advanced or Premium edition of the Citrix ADC VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

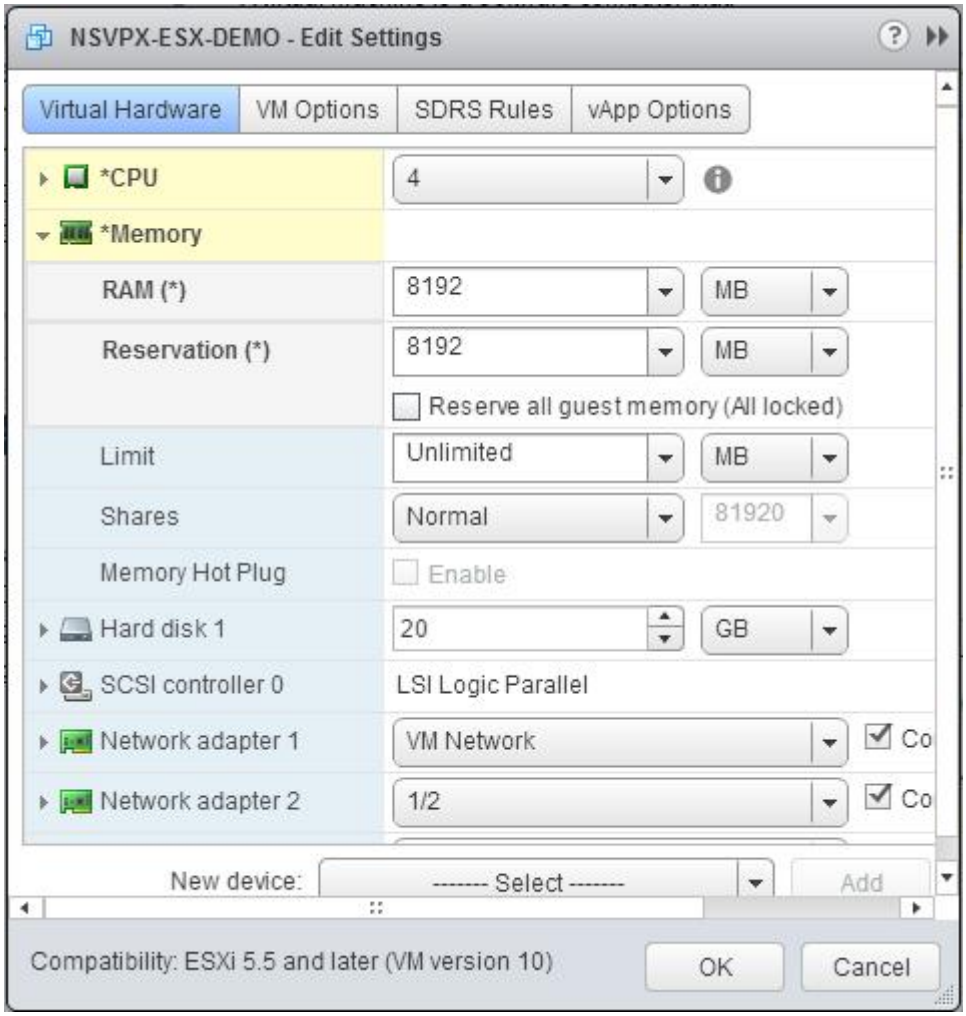


b. In the Reservation drop-down list, enter the value for the memory reservation, and select the Reserve all guest memory (All locked) check box. The memory reservation must be the number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation must be 4 x 2 GB = 8 GB.

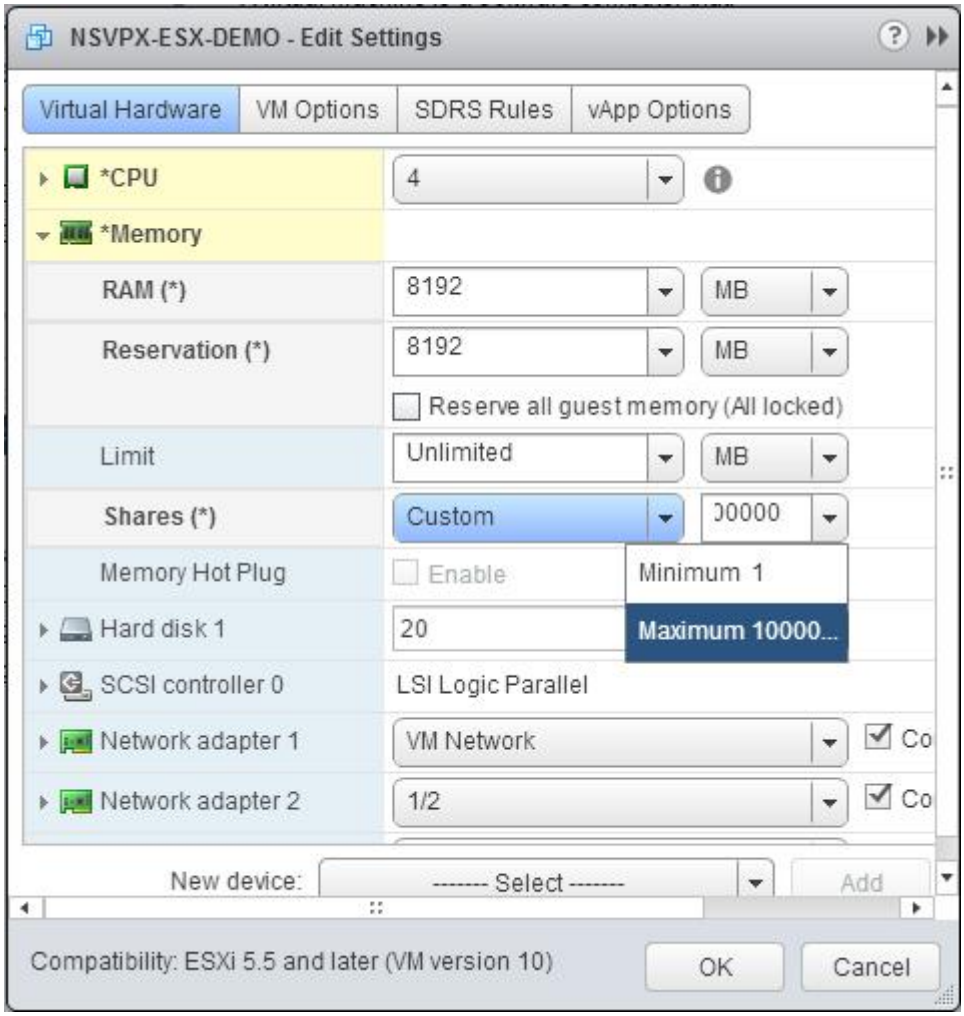
Note: For an Advanced or Premium edition of the Citrix ADC VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.



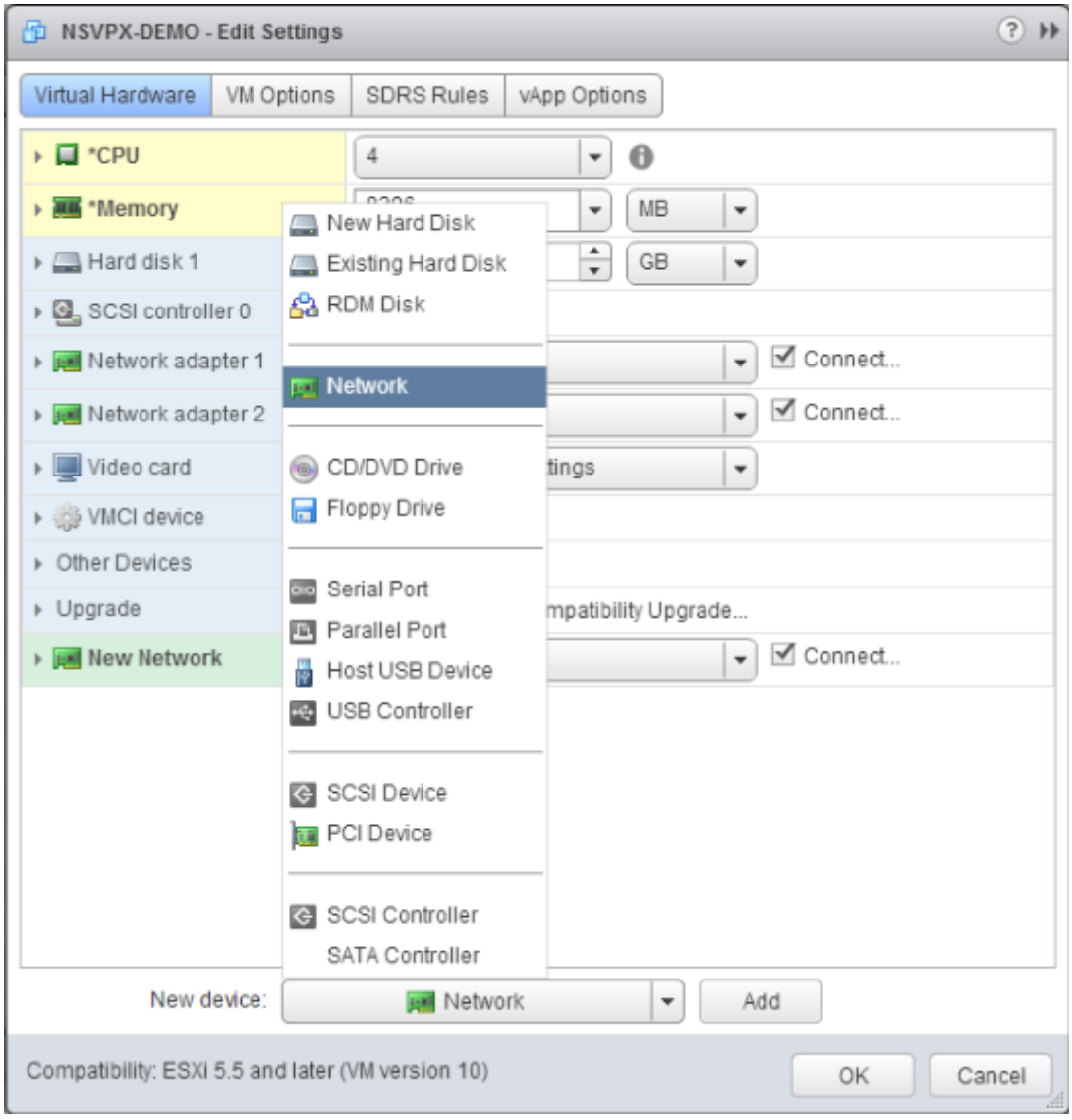
c. In the Limit drop-down list, select the number that is shown as the maximum value.



d. In the Shares drop-down lists, select Custom and the number that is shown as the maximum value.



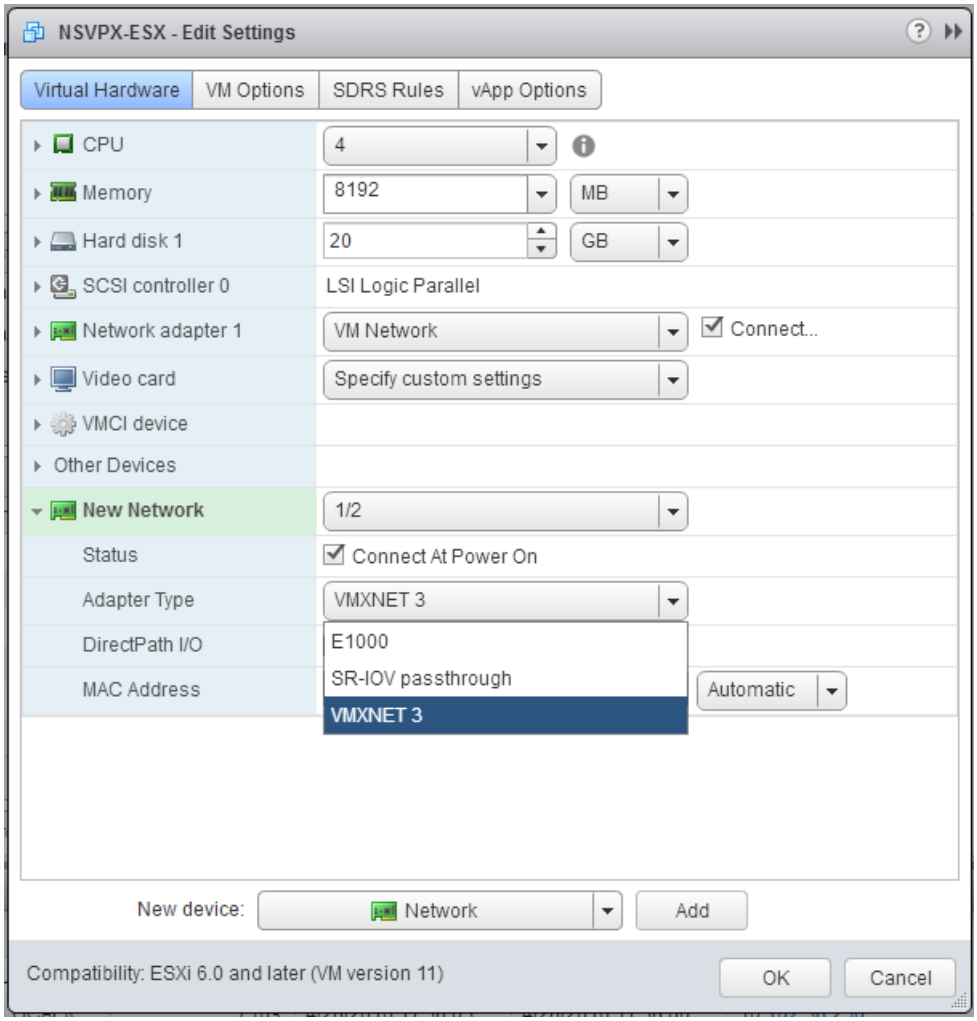
7. Add a VMXNET3 network interface. From the New device drop-down list, select Network and click Add.



8. In the New Network section, from the drop-down list, select the network interface, and do the following:

   a. In the Adapter Type drop-down list, select VMXNET3.

   > **Important**
   >
   > The default E1000 network interface and VMXNET3 cannot coexist, make sure that you remove the E1000 network interface and use VMXNET3 (0/1) as the management interface.



9. Click OK.
10. Power on the Citrix ADC VPX instance.
11. Once the Citrix ADC VPX instance powers on, you can use the following command to verify the configuration:

    ```
    show interface summary
    ```

The output must show all the interfaces that you configured:

```
> show interface summary
-------------------------------------------------------------------------

      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------

1     0/1        1500       00:0c:29:89:1d:0e  NetScaler Vir...rface,
    VMXNET3
```



```
6  2     1/1        9000       00:0c:29:89:1d:18    NetScaler Vir...rface,
       VMXNET3
7  3     1/2        9000       00:0c:29:89:1d:22    NetScaler Vir...rface,
       VMXNET3
8  4     LO/1       9000       00:0c:29:89:1d:0e    Netscaler Loopback
      interface
```

> **Note**
>
> After you add a VMXNET3 interface and restart the Citrix ADC VPX appliance, the VMware ESX hypervisor might change the order in which the NIC is presented to the VPX appliance. So, network adapter 1 might not always remain 0/1, resulting in loss of management connectivity to the VPX appliance. To avoid this issue, change the virtual network of the network adapter accordingly.
>
> This is a VMware ESX hypervisor limitation.

Configure a Citrix ADC VPX instance to use SR-IOV network interface

September 14, 2021

After you have installed and configured the Citrix ADC VPX instance on VMware ESX, you can use the VMware vSphere web client to configure the virtual appliance to use single root I/O v virtualization (SR-IOV) network interfaces.

Limitations

A Citrix ADC VPX configured with SR-IOV network interface has the following limitations:

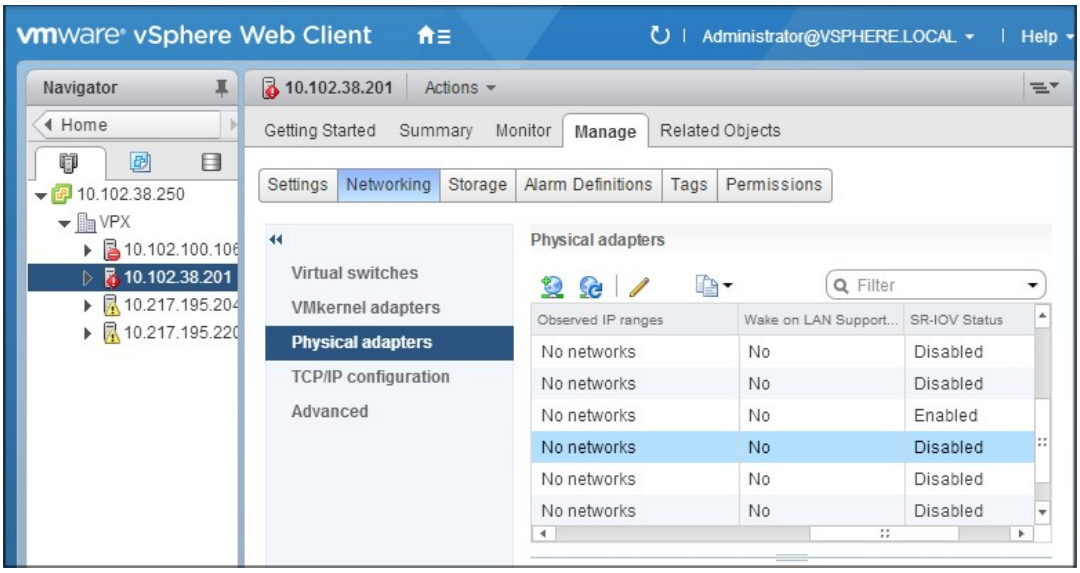
- The following features are not supported on SR-IOV interfaces using the Intel 82599 10G NIC on ESX VPX:
  - L2 mode switching
  - Static Link Aggregation and LACP
  - Clustering
  - Admin partitioning [Shared VLAN mode]
  - High Availability [Active - Active mode]
  - Jumbo frames
  - IPv6
- The following features are not supported on the SR-IOV interface with an Intel 82599 10G NIC on KVM VPX:
  - Static Link Aggregation and LACP
  - L2 mode switching

- Clustering
- Admin partitioning [Shared VLAN mode]
- High Availability [Active – Active mode]
- Jumbo frames
- IPv6
- VLAN configuration on Hypervisor for SR-IOV VF interface through `ip link` command is not supported

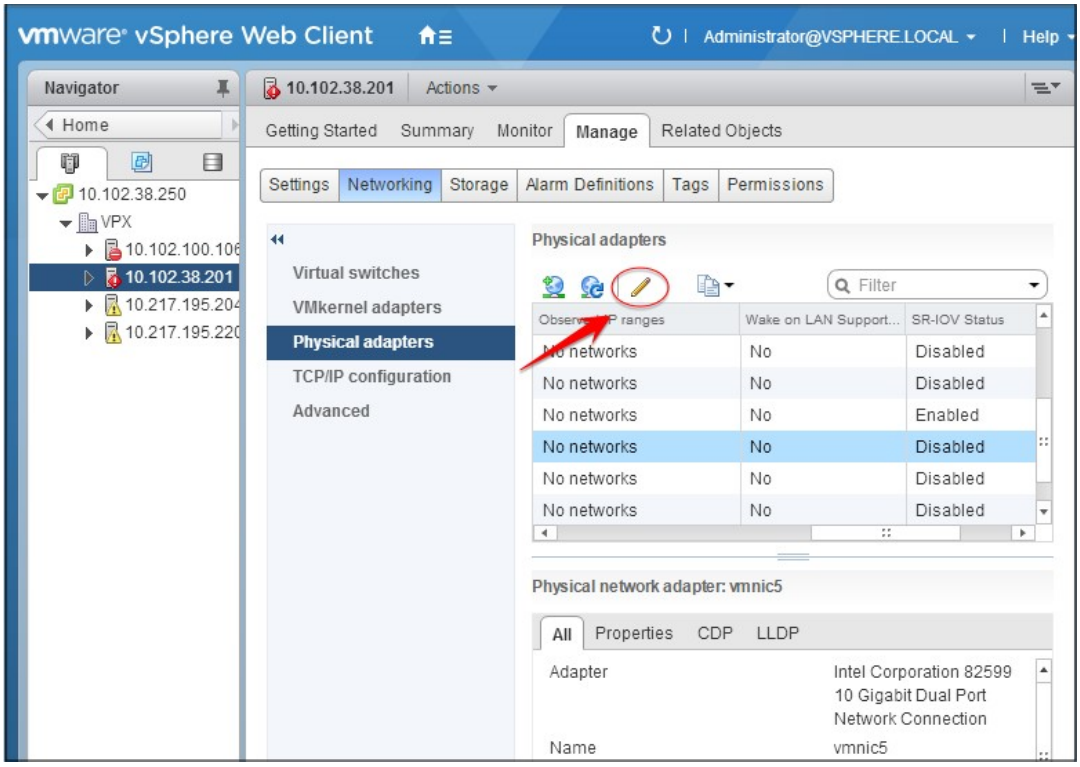
Prerequisite

Make sure that you:

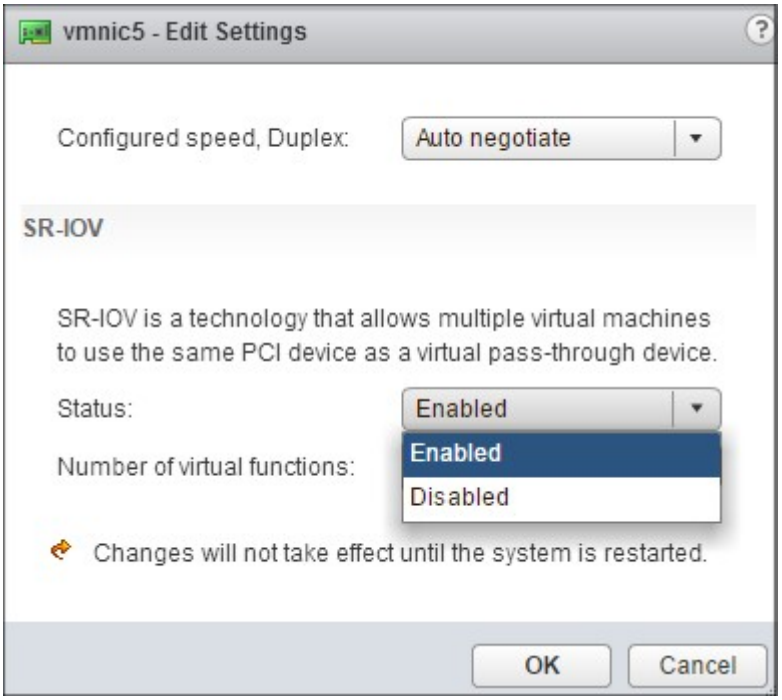
- Add the Intel 82599 NIC (NIC) to the ESX Host. IXGBE driver version 3.7.13.7.14iov is recommended.
- Enable SR-IOV on the host physical adapter, as follows:
  1. In the vSphere Web Client, navigate to the Host.
  2. On the **Manage > Networking** tab, select **Physical adapters**. The SR-IOV Status field shows whether a physical adapter supports SR-IOV.



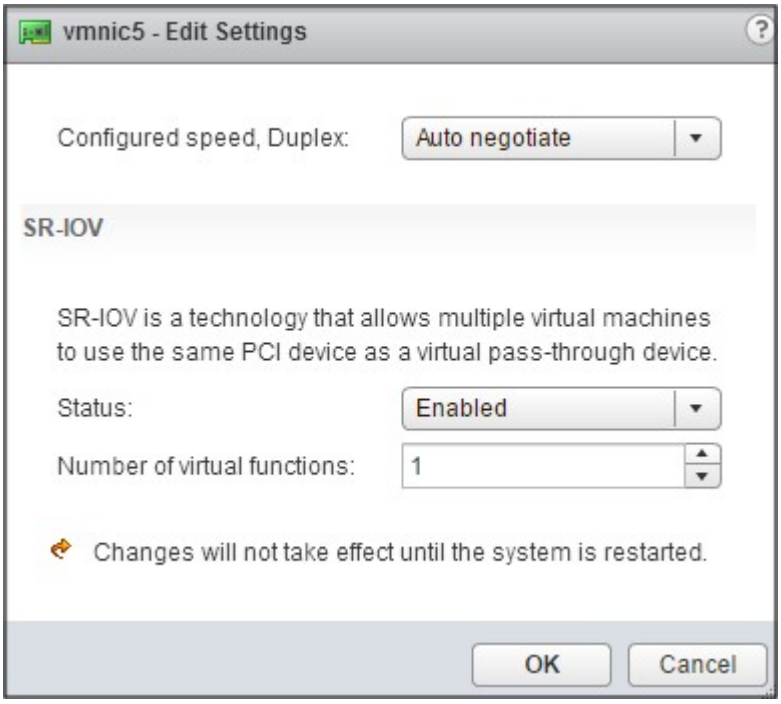
  3. Select the physical adapter, and then click the pencil icon to open the **Edit Settings** dialog box.



4. Under SR-IOV, select **Enabled** from the **Status** drop-down list.



5. In the **Number of virtual functions** field, enter the number of virtual functions that you want to configure for the adapter.



6. Click **OK**.
7. Restart the host.

- Create a Distributed Virtual Switch (DVS) and `Portgroups`. For instructions, see the VMware Documentation.

  > **Note**
  >
  > Citrix has qualified the SR-IOV configuration on DVS and `Portgroups` only.

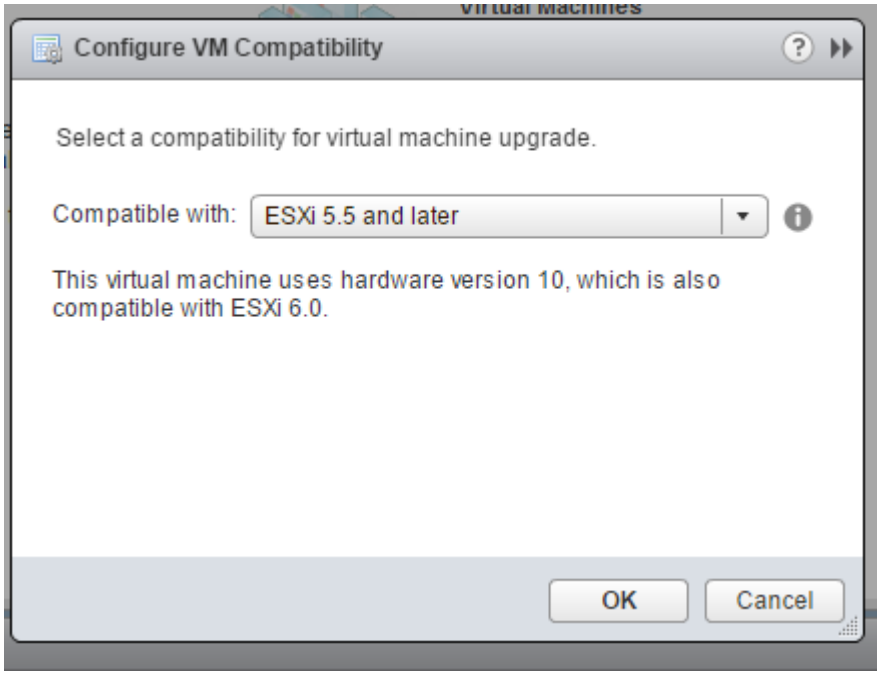
To configure Citrix ADC VPX instances to use SR-IOV network interface by using VMware vSphere Web Client:

1. In the vSphere Web Client, select **Hosts and Clusters**.
2. Upgrade the Compatibility setting of the Citrix ADC VPX instance to ESX 5.5 or later, as follows:

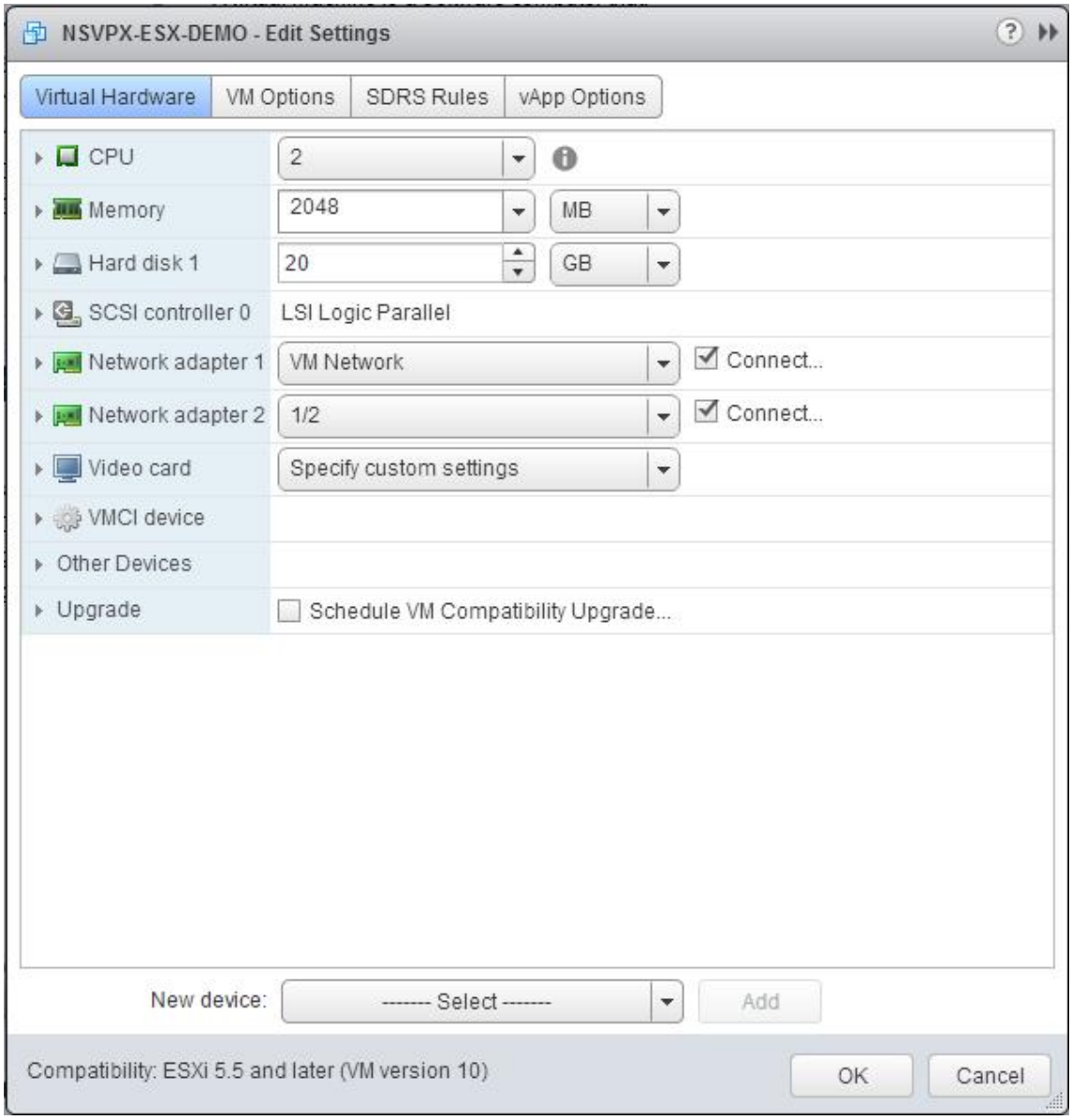
   a. Power off the Citrix ADC VPX instance.

   b. Right-click the Citrix ADC VPX instance and select **Compatibility > Upgrade VM Compatibility**.

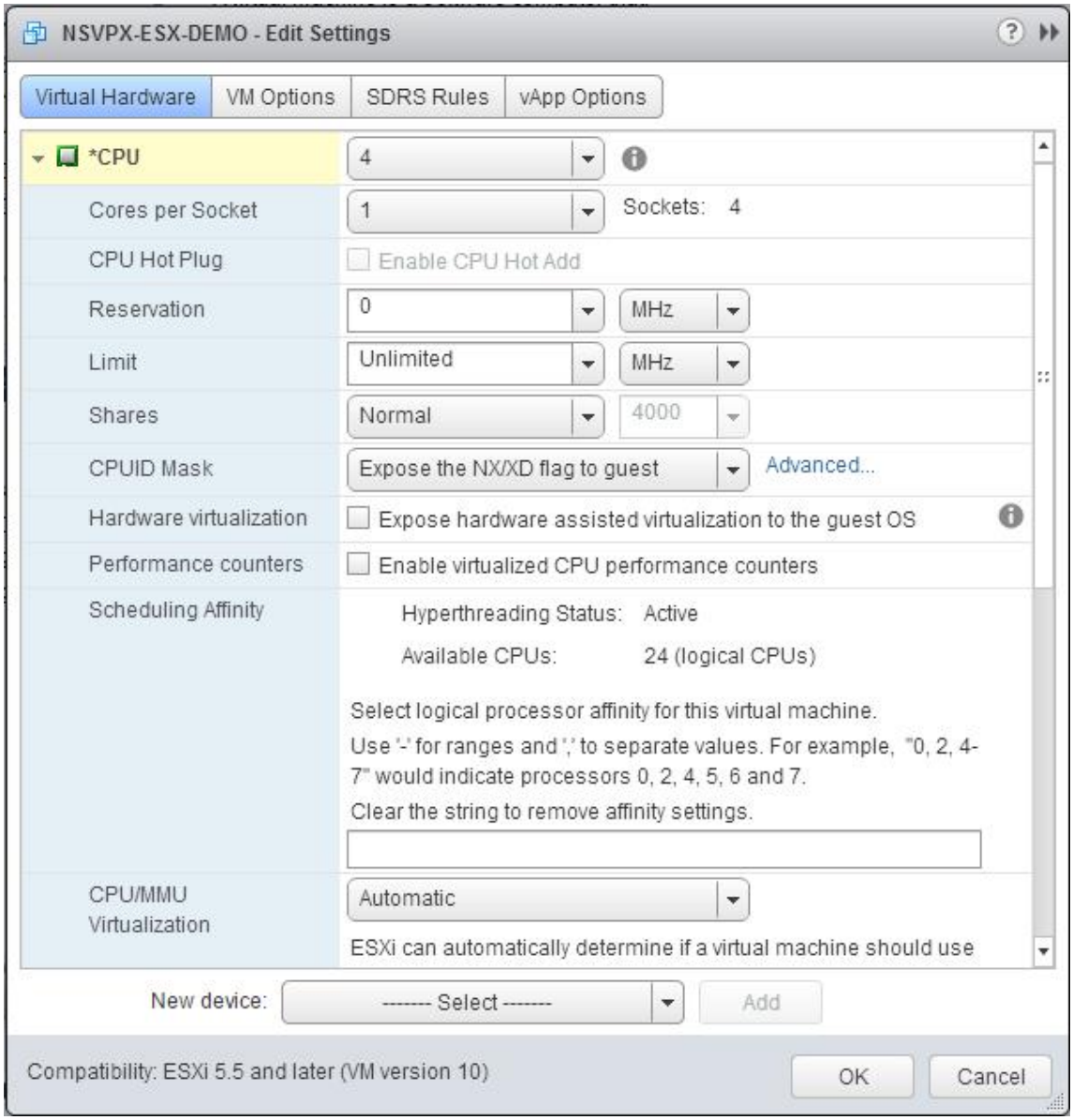
   c. In the **Configure VM Compatibility** dialog box, select **ESXi 5.5 and later** from the **Compatible with** drop-down list and click **OK**.



3. Right-click on the Citrix ADC VPX instance and click **Edit Settings**.



4. In the **<virtual_appliance> - Edit Settings** dialog box, click the **CPU** section.



5. In the **CPU** section, update the following settings:

   - Number of CPUs
   - Number of Sockets
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

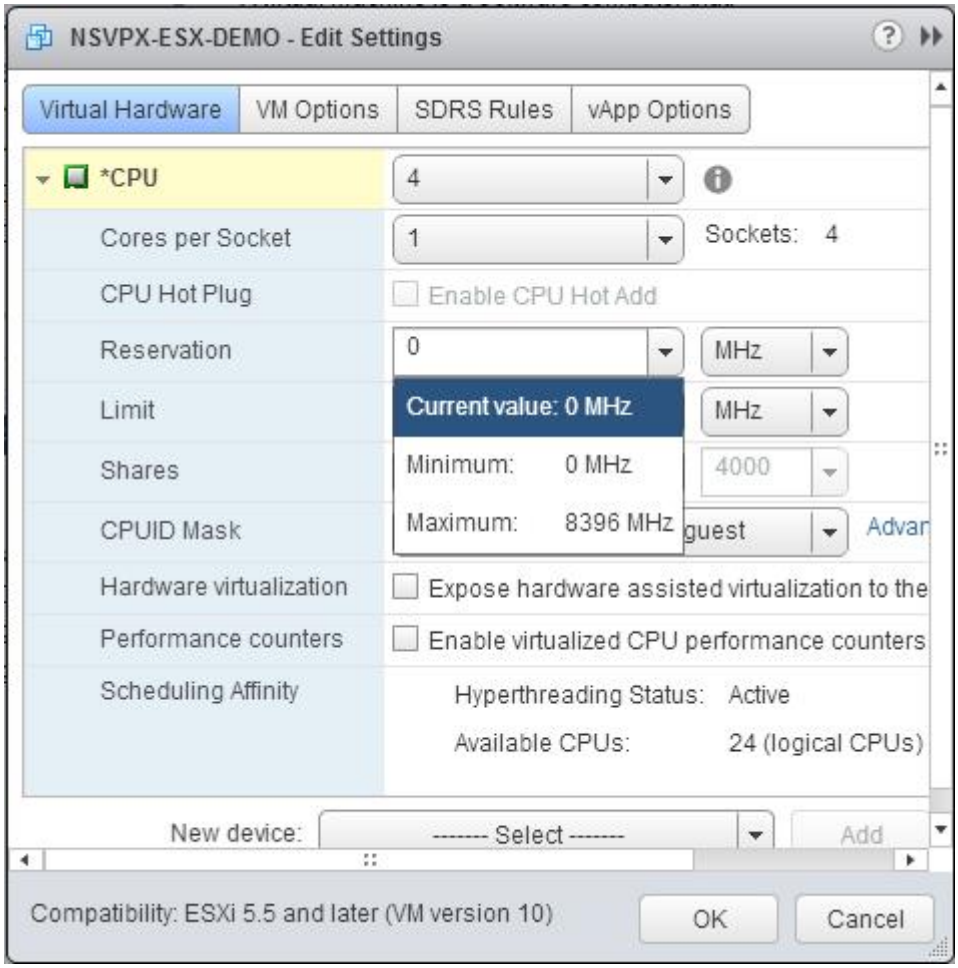
   a. In the **CPU** drop-down list, select the number of CPUs to assign to the virtual appliance.

   b. In the **Cores per Socket** drop-down list, select the number of sockets.

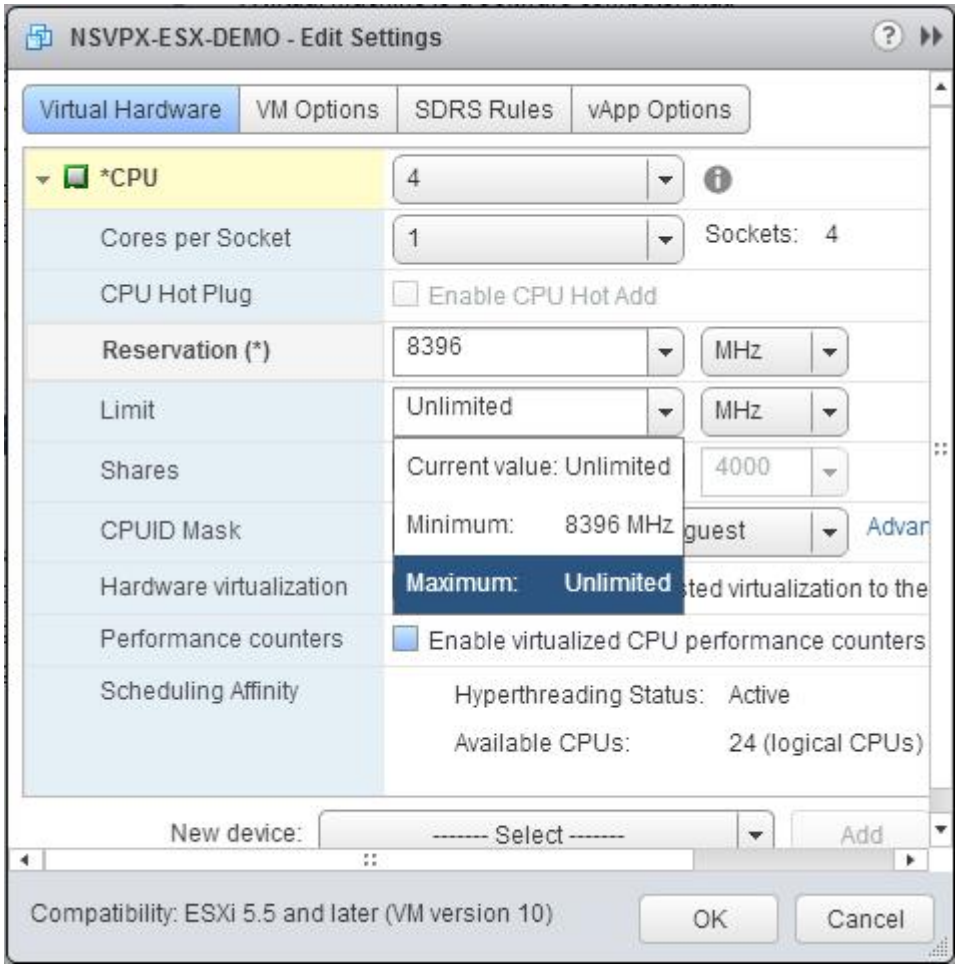
   c. (Optional) In the **CPU Hot Plug** field, select or clear the **Enable CPU Hot Add** check box.

Note: Citrix recommends accepting the default (disabled).

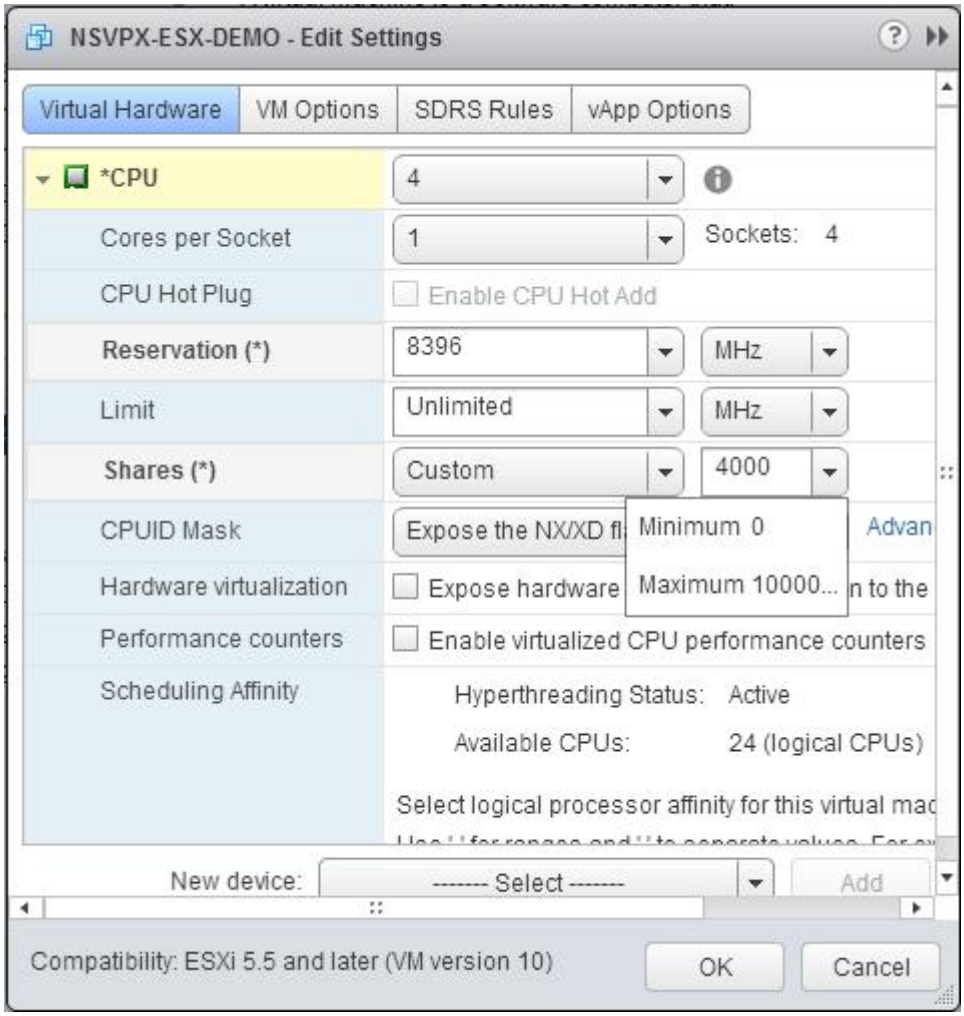
d. In the **Reservation** drop-down list, select the number that is shown as the maximum value.



e. In the **Limit** drop-down list, select the number that is shown as the maximum value.



f. In the **Shares** drop-down lists, select **Custom** and the number that is shown as the maximum value.



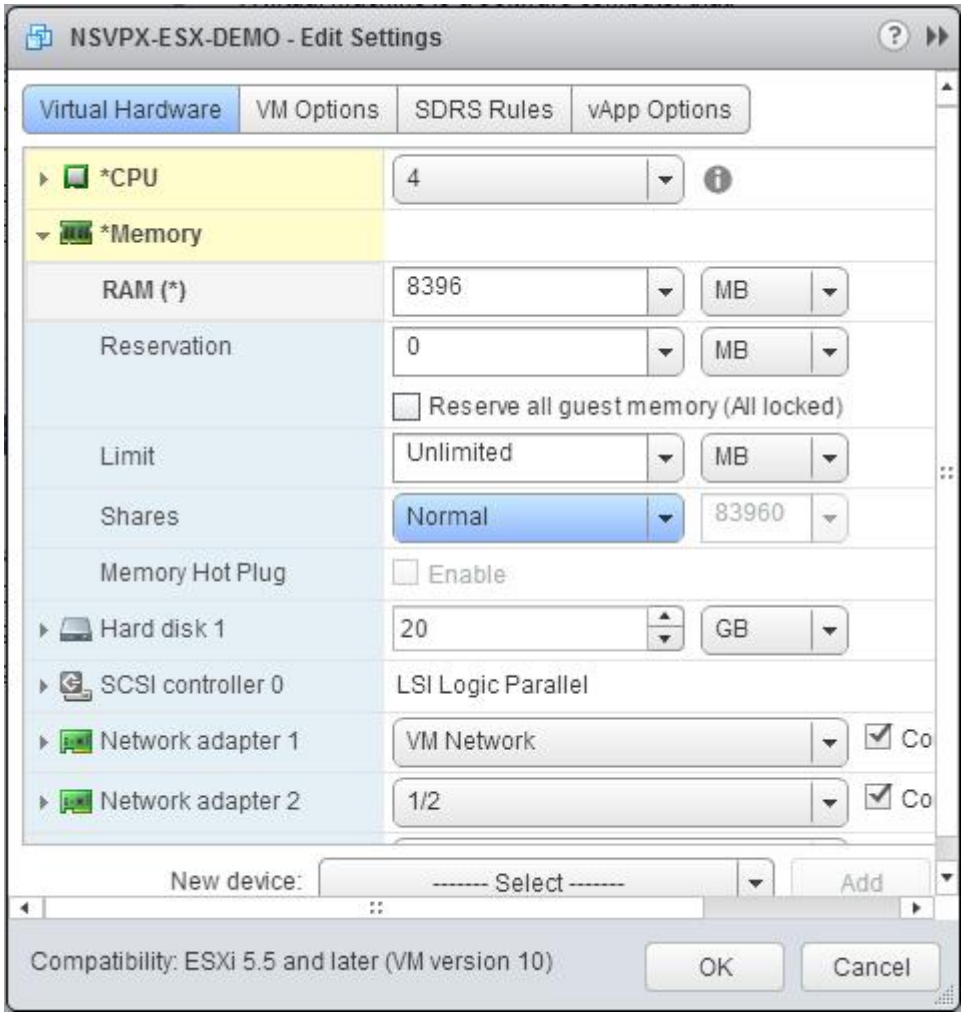
6. In the **Memory** section, update the following settings:

   - Size of RAM
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

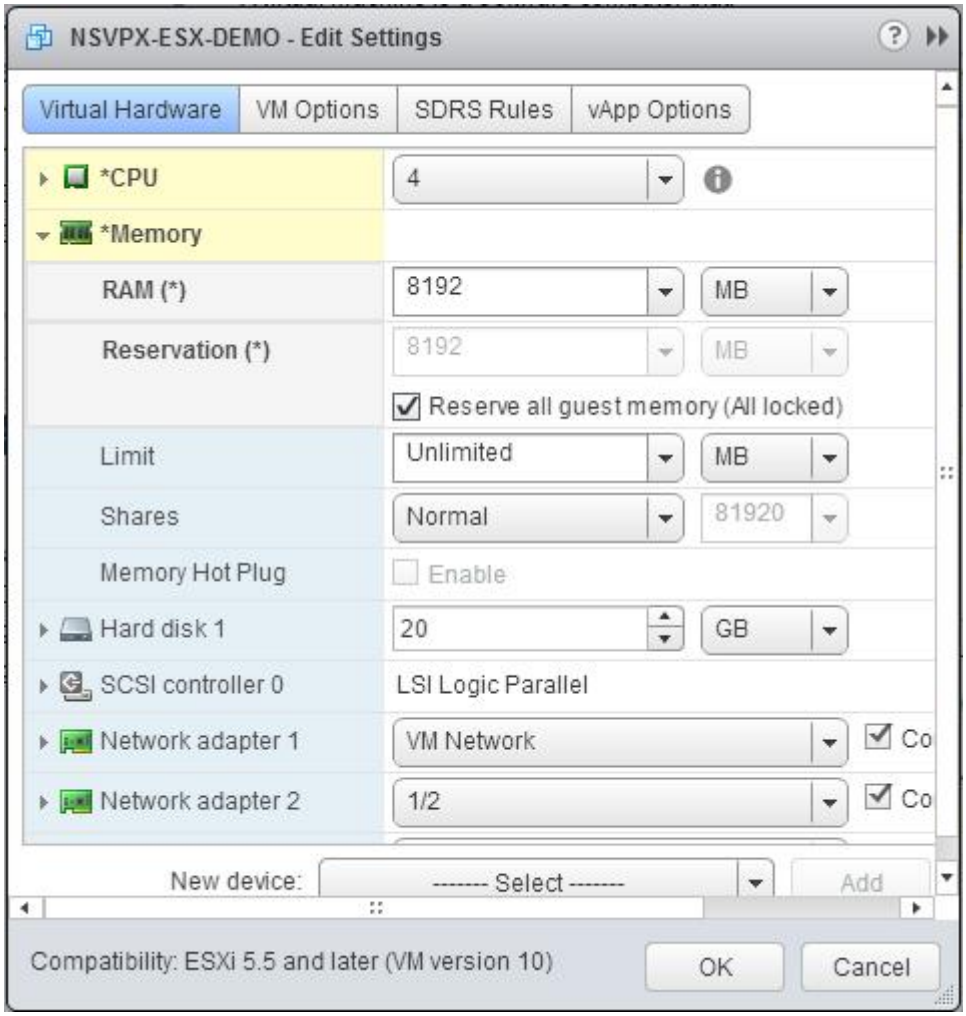
   a. In the **RAM** drop-down list, select the size of the RAM. It must be the number of vCPUs x 2 GB. For example, if the number of vCPU is 4 then RAM = 4 x 2 GB = 8 GB.

   Note: For Advanced or Premium edition of the Citrix ADC VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

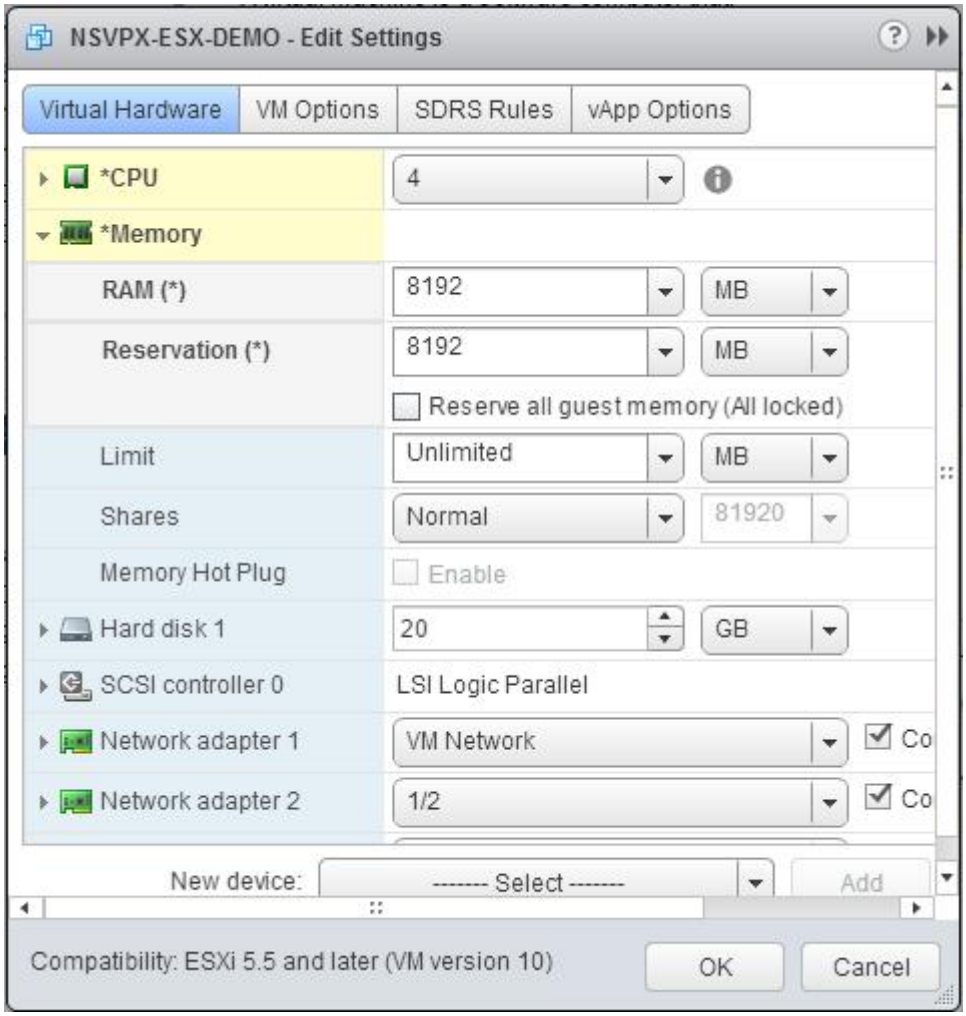


b. In the **Reservation** drop-down list, enter the value for the memory reservation, and select the **Reserve all guest memory (All locked)** check box. The memory reservation must be number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation must be 4 x 2 GB = 8 GB.

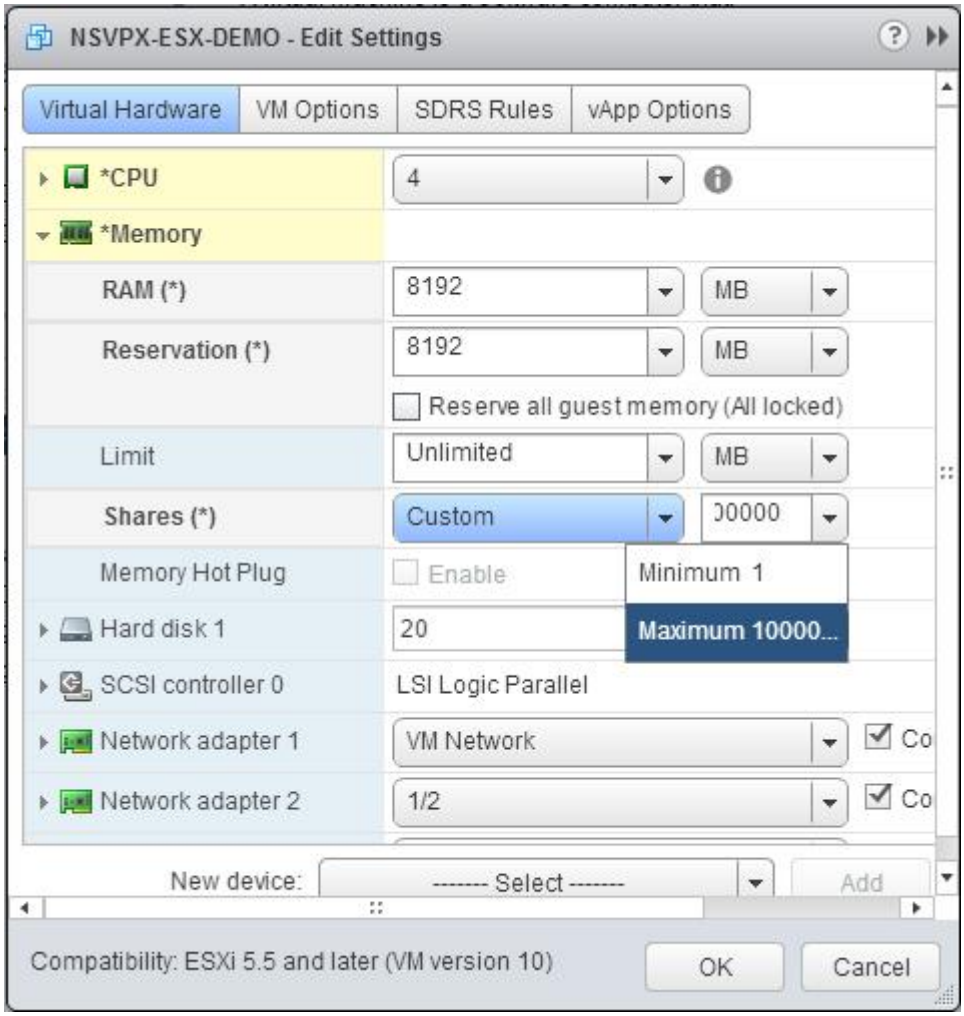
Note: For Advanced or Premium edition of the Citrix ADC VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.



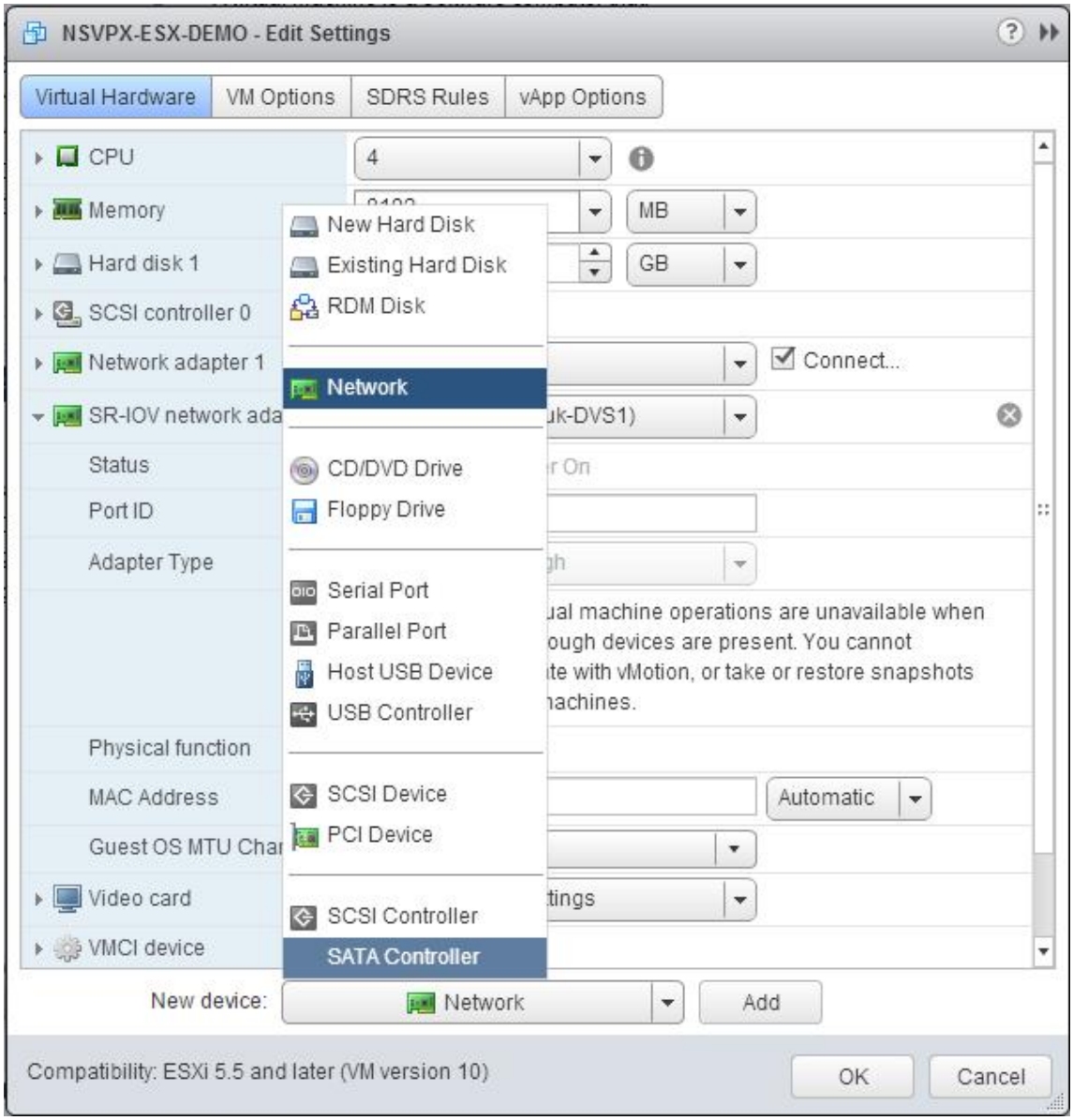
c. In the **Limit** drop-down list, select the number that is shown as the maximum value.



d. In the **Shares** drop-down lists, select **Custom**, and select the number that is shown as the maximum value.

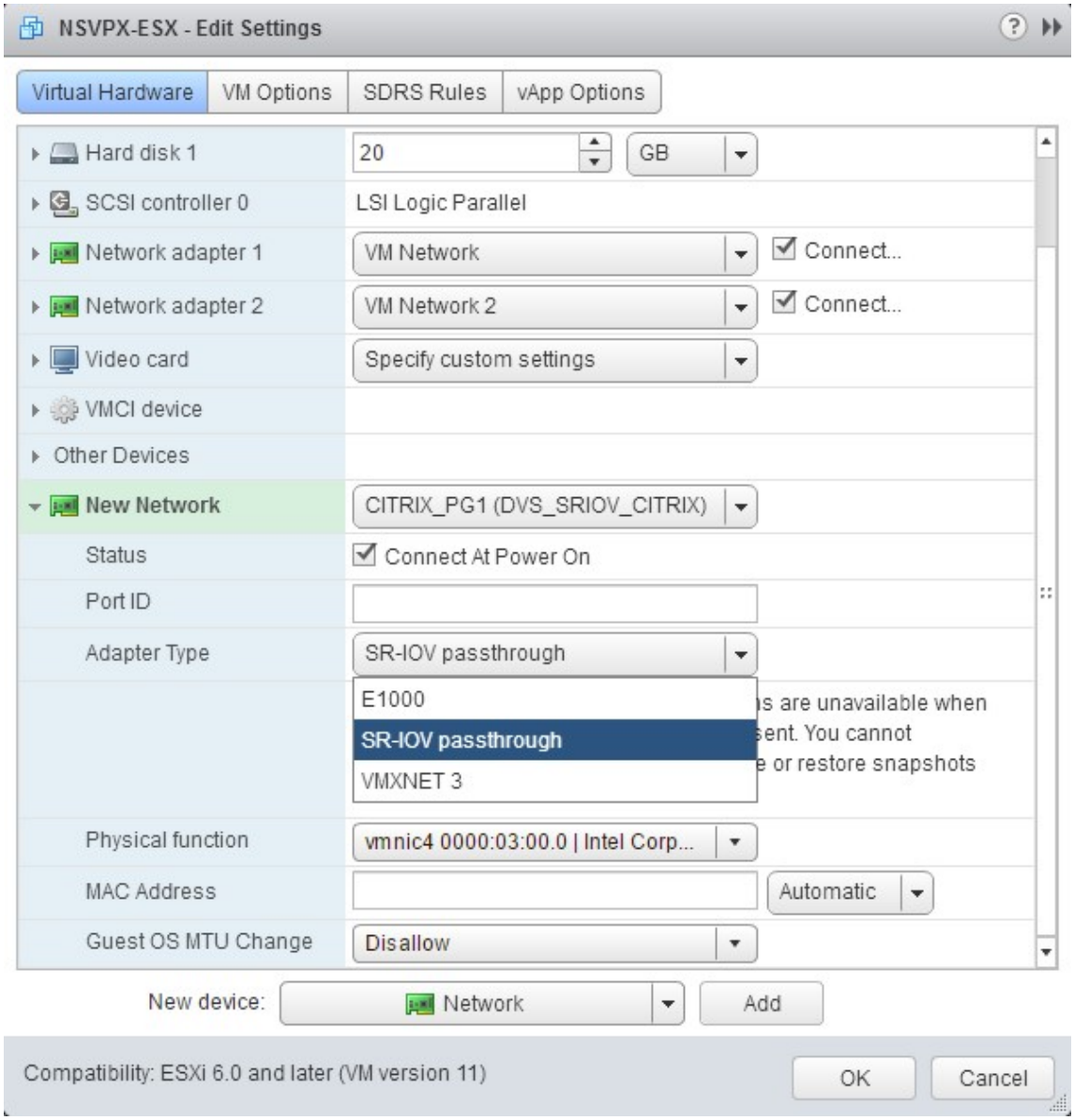


7. Add an SR-IOV network interface. From the **New device** drop-down list, select **Network** and click **Add**.

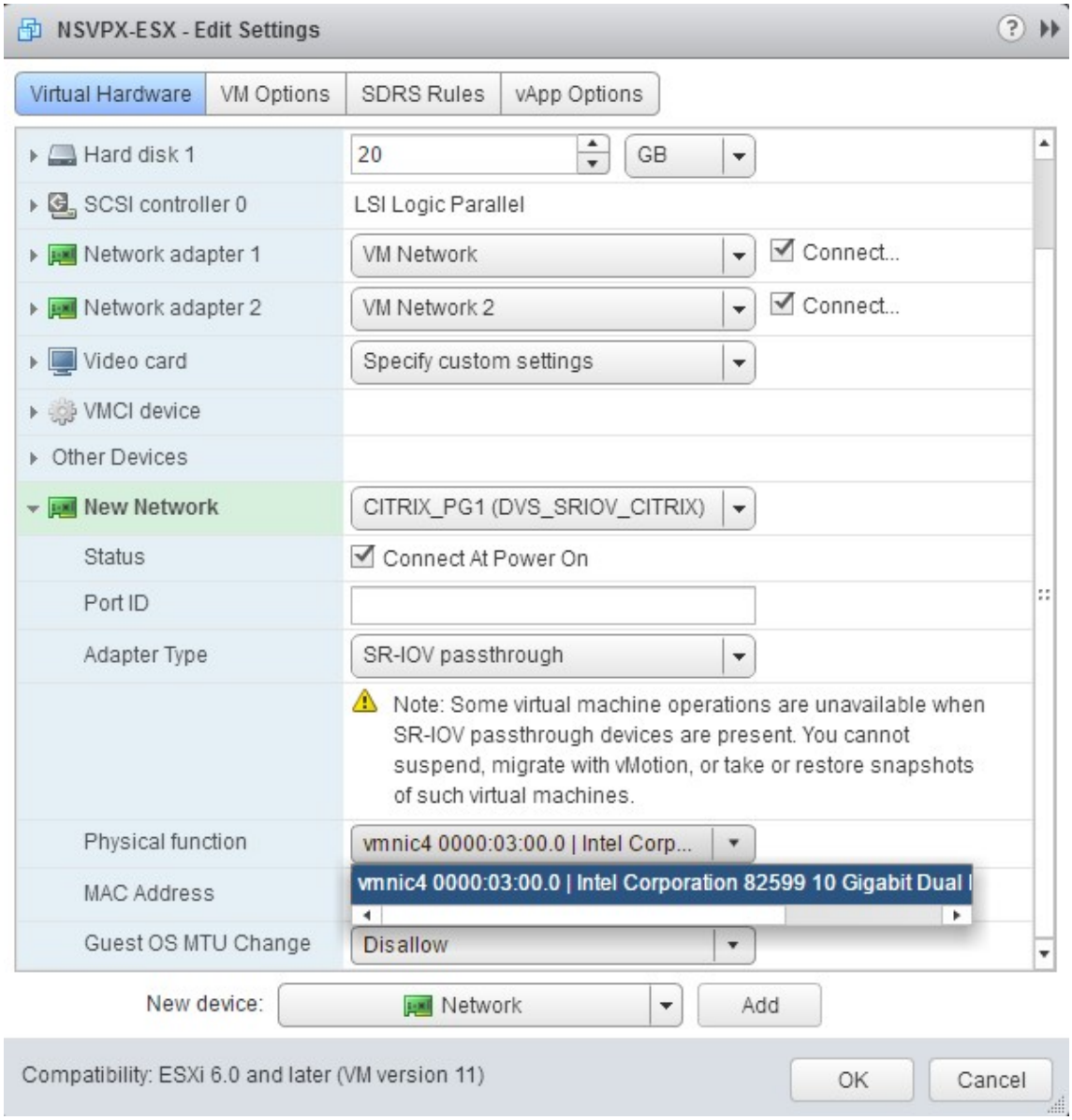


8. In the **New Network** section. From the drop-down list, select the `Portgroup` that you created, and do the following:

   a. In the **Adapter Type** drop-down list, select **SR-IOV passthrough**.

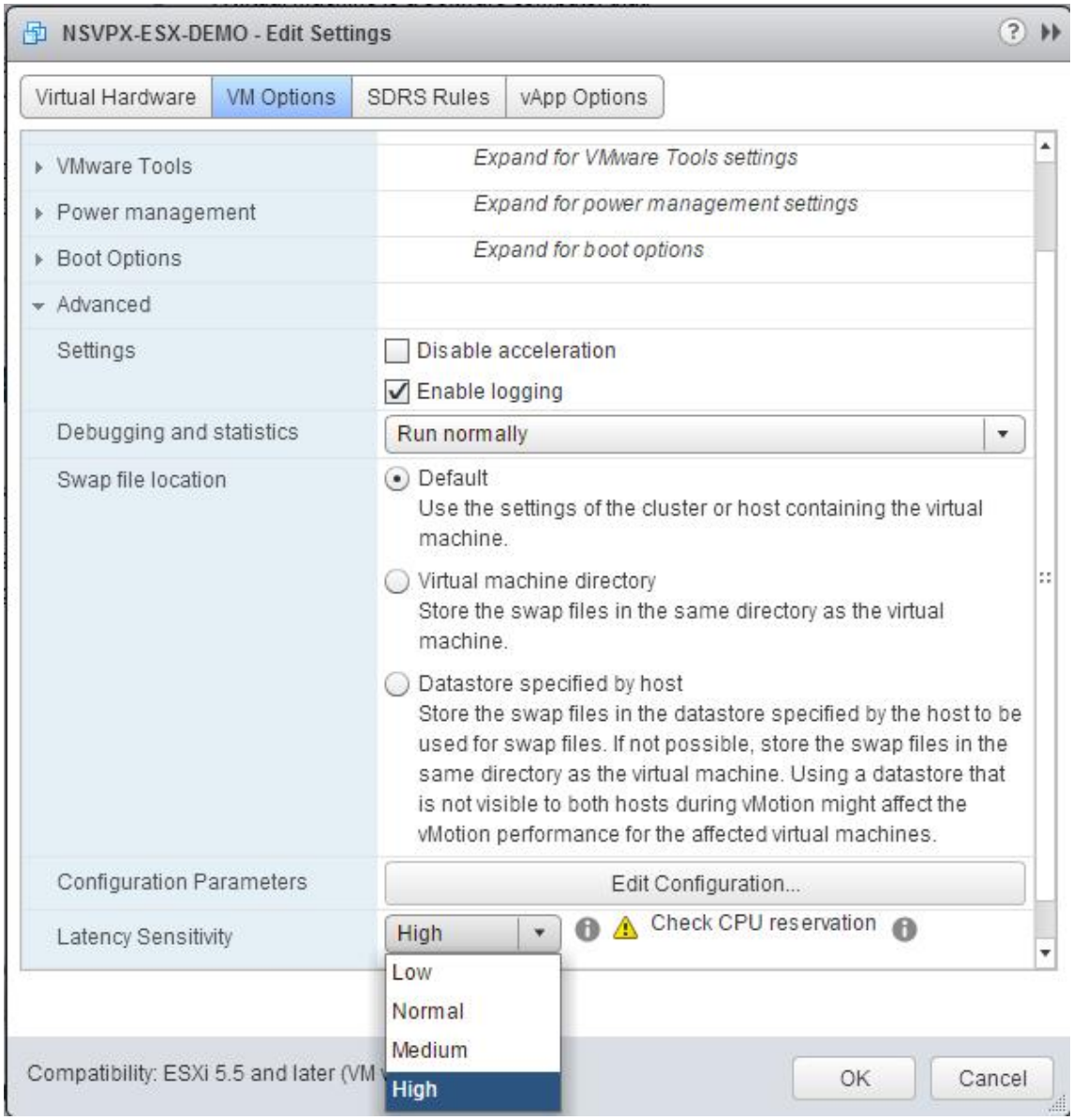


b. In the **Physical function** drop-down list, select the physical adapter mapped with the `Portgroup`.



c. In the **Guest OS MTU Change** drop-down list, select **Disallow**.

9. In the **<virtual_appliance> - Edit Settings** dialog box, click the **VM Options** tab.
10. On the **VM Options** tab, select the **Advanced** section. From the **Latency Sensitivity** drop-down list, select **High**.



11. Click **OK**.
12. Power on the Citrix ADC VPX instance.
13. Once the Citrix ADC VPX instance powers on, you can use the following command to verify the configuration:

    ```
    show interface summary
    ```

The output must show all the interfaces that you configured:

```
> show interface summary
-------------------------------------------------------------------------------

      Interface  MTU       MAC                Suffix
```



```
4 ----------------------------------------------------------------------------

5 1     0/1        1500       00:0c:29:1b:81:0b    NetScaler Virtual
     Interface
6 2     10/1       1500       00:50:56:9f:0c:6f    Intel 82599 10G VF
     Interface
7 3     10/2       1500       00:50:56:9f:5c:1e    Intel 82599 10G VF
     Interface
8 4     10/3       1500       00:50:56:9f:02:1b    Intel 82599 10G VF
     Interface
9 5     10/4       1500       00:50:56:9f:5a:1d    Intel 82599 10G VF
     Interface
10 6    10/5       1500       00:50:56:9f:4e:0b    Intel 82599 10G VF
     Interface
11 7    LO/1       1500       00:0c:29:1b:81:0b    Netscaler Loopback
     interface
12  Done
13 > show inter 10/1
14 1)      Interface 10/1 (Intel 82599 10G VF Interface) #1
15         flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>
16         MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0
              h21m53s
17         Actual: media FIBER, speed 10000, duplex FULL, fctl NONE,
              throughput 10000
18         LLDP Mode: NONE,                LR Priority: 1024
19
20         RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527)
              Stalls(0)
21         TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls
              (0)
22         NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
23         Bandwidth thresholds are not set.
24  Done
```

Migrating the Citrix ADC VPX from E1000 to SR-IOV or VMXNET3 Network Interfaces

September 14, 2021

May 24, 2018

You can configure your exiting Citrix ADC VPX instances that use E1000 network interfaces to use SR-

IOV or VMXNET3 network interfaces.

To configure an existing Citrix ADC VPX instance to use SR-IOV network interfaces, see Configure a Citrix ADC VPX instance to use SR-IOV network interface.

To configure an existing Citrix ADC VPX instance to use VMXNET3 network interfaces, see Configure a Citrix ADC VPX instance to use VMXNET3 network interface.

Configure a Citrix ADC VPX instance to use PCI passthrough network interface

September 14, 2021

Overview

After you have installed and configured a Citrix ADC VPX instance on VMware ESX Server, you can use the vSphere Web Client to configure the virtual appliance to use PCI passthrough network interfaces.

The PCI passthrough feature allows a guest virtual machine to directly access physical PCI and PCIe devices connected to a host.

Prerequisites

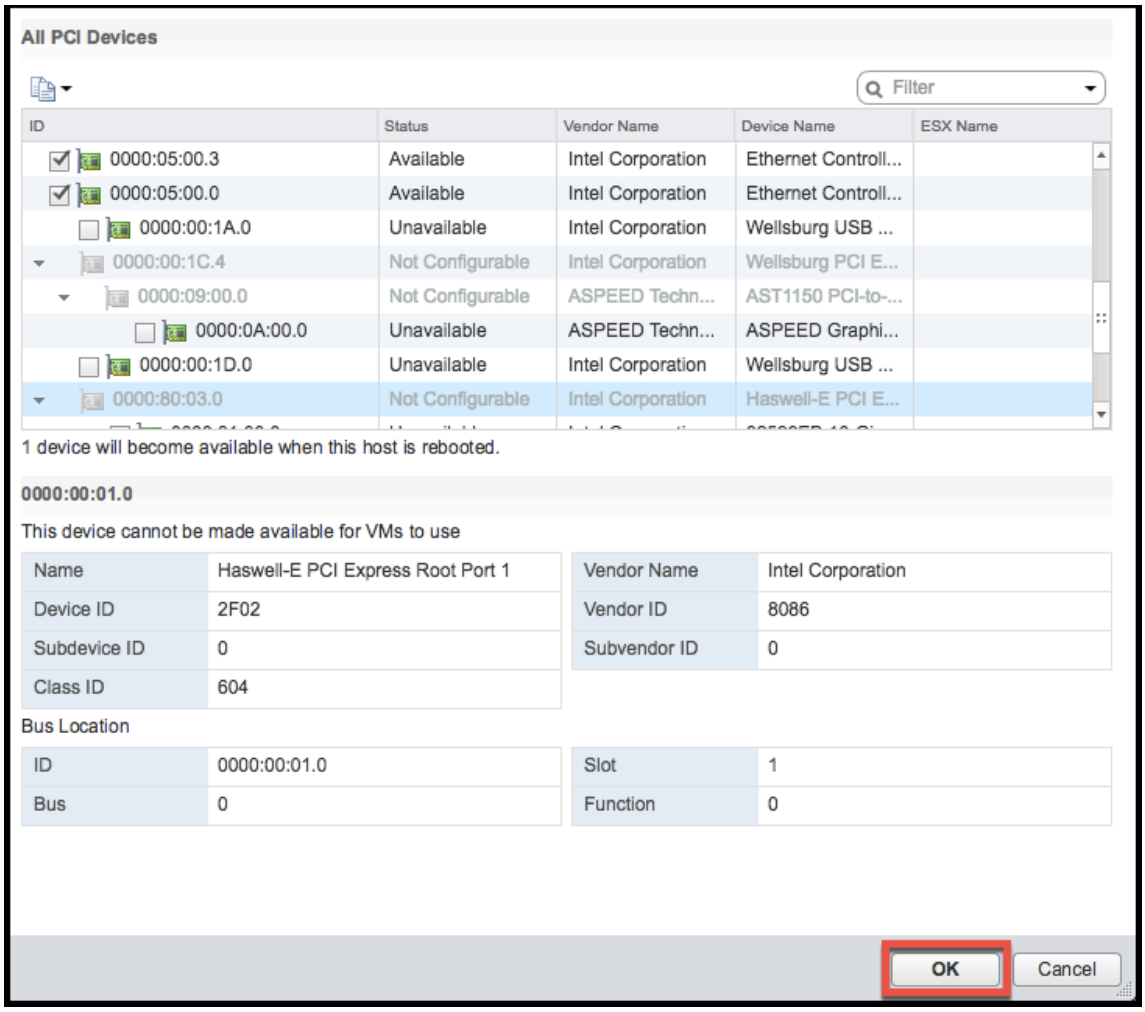
- The firmware version of the Intel XL710 NIC on the host is 5.04.
- A PCI passthrough device connected to and configured on the host
- Supported NICs:
  - Intel X710 10G NIC
  - Intel XL710 Dual Port 40G NIC
  - Intel XL710 Single Port 40G NIC

Configure passthrough devices on a host

Before configuring a passthrough PCI device on a virtual machine, you must configure it on the host machine. Follow these steps to configure passthrough devices on a host.

1. Select the host from the Navigator panel of the vSphere Web Client.
2. Click **Manage** > **Settings** > **PCI Devices**. All available passthrough devices are displayed.
3. Right-click the device that you want to configure and click **Edit**.
4. The **Edit PCI Device Availability** window appears.

5. Select the devices to be used for passthrough and click **OK**.

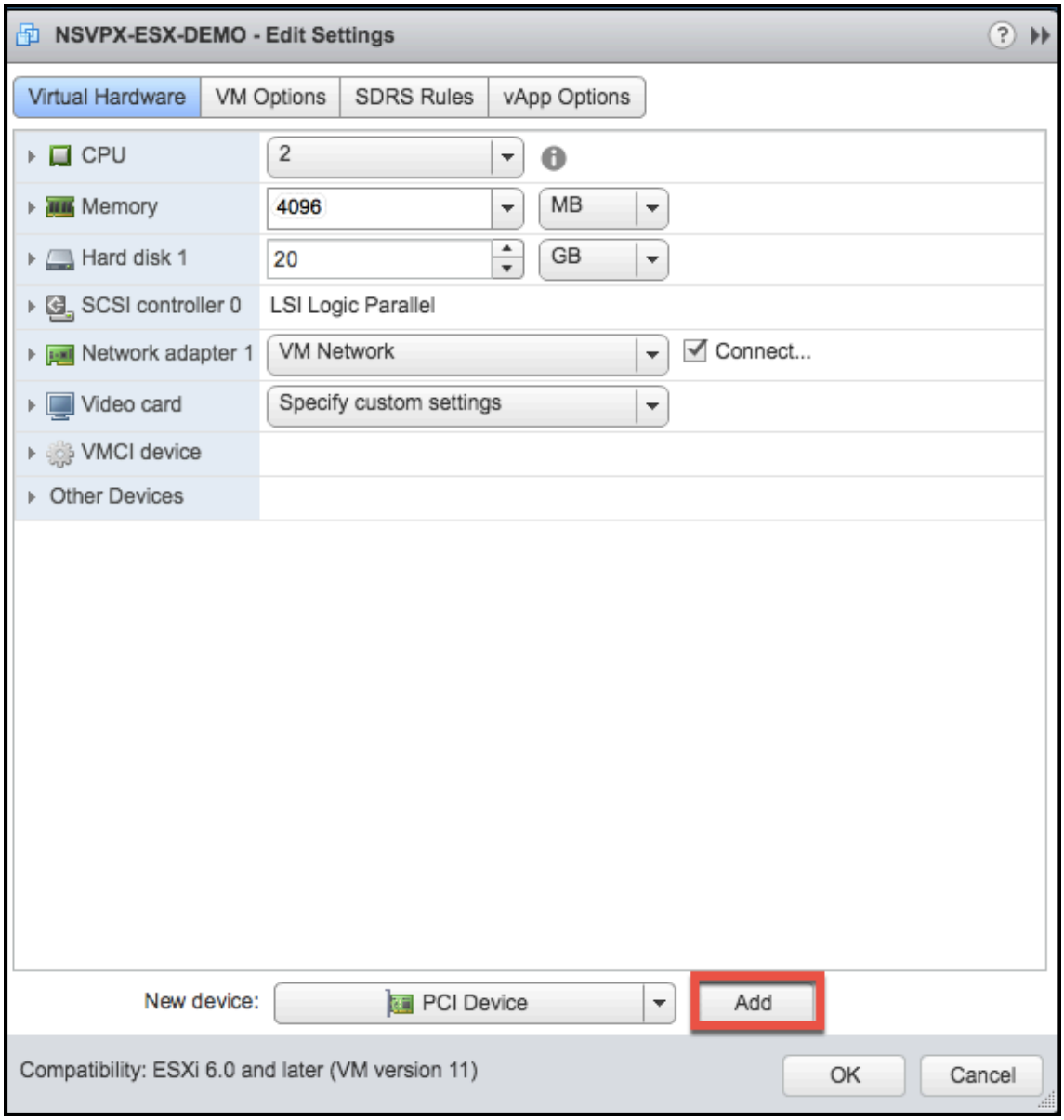


6. Restart the host machine.

Configure passthrough devices on a Citrix ADC VPX instance

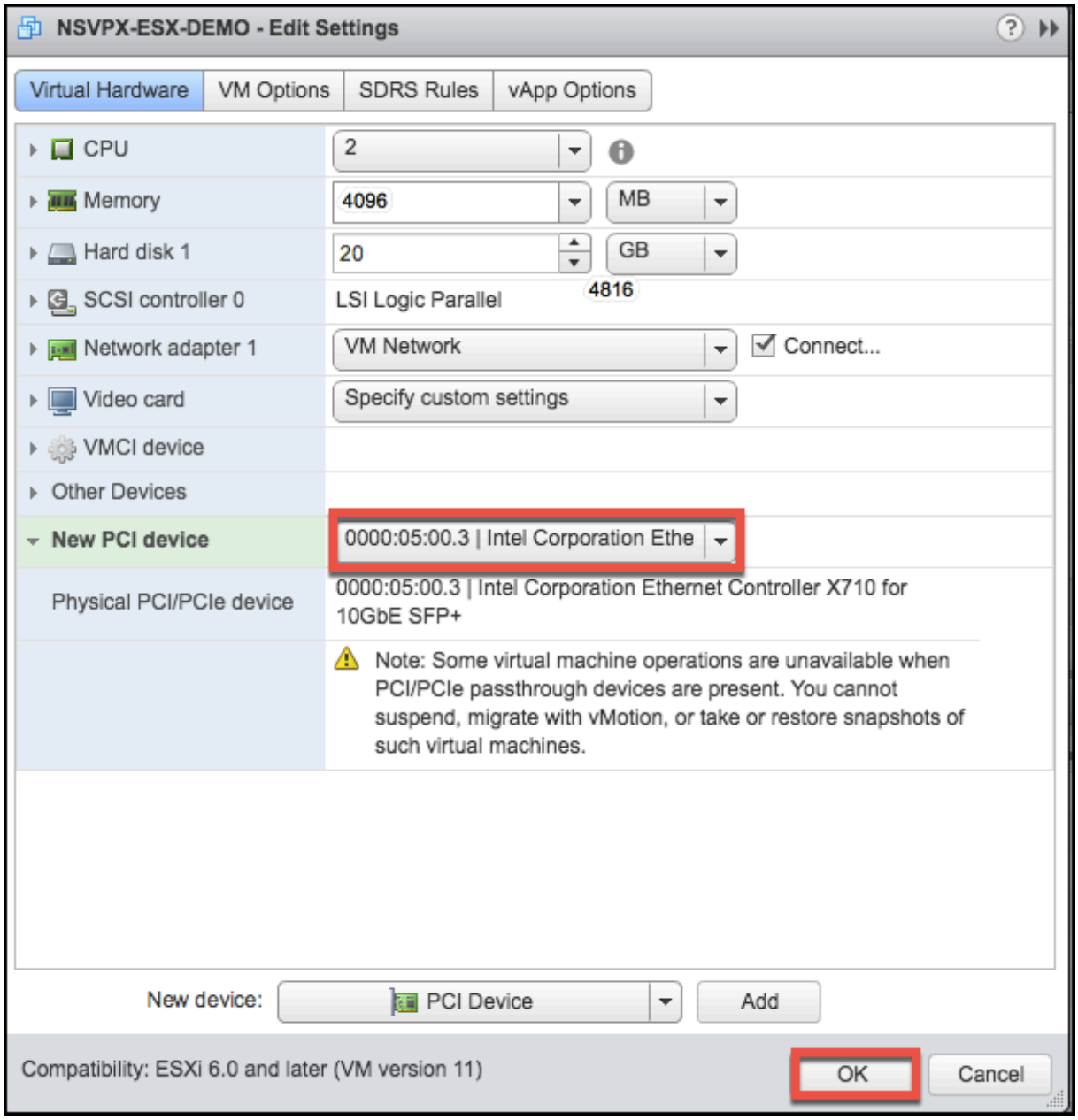
Follow these steps to configure a passthrough PCI device on a Citrix ADC VPX instance.

1. Power off the virtual machine.
2. Right-click the virtual machine and select **Edit Settings**.
3. On the **Virtual Hardware** tab, select **PCI Device** from the **New Device** drop-down menu, and click **Add**.



4. Expand **New PCI device** and select the passthrough device to connect to the virtual machine from the drop-down list and click **OK**.

> **Note**
>
> VMXNET3 network interface and PCI Passthrough Network Interface cannot coexist.



1. Power on the guest virtual machine.

You have completed the steps to configuring Citrix ADC VPX to use PCI passthrough network interfaces.

Apply Citrix ADC VPX configurations at the first boot of the Citrix ADC appliance on VMware ESX hypervisor

September 29, 2021

You can apply the Citrix ADC VPX configurations during the first boot of the Citrix ADC appliance on the VMware ESX hypervisor. Therefore in certain cases, a specific setup or VPX instance is brought up in much lesser time.

For more information on Preboot user data and its format, see Apply Citrix ADC VPX configurations at the first boot of the Citrix ADC appliance in cloud.

> Note:
>
> To bootstrap using preboot user data in ESX, default gateway config must be passed in <NS-CONFIG> section. For more information on the content of the <NS-CONFIG> tag, see Sample-<NS-CONFIG>-section.

Sample <NS-CONFIG> section:

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

How to provide preboot user data on ESX hypervisor

You can provide preboot user data on ESX hypervisor in the following two ways:

- Using CD/DVD ISO
- Using OVF Property

Provide user data using CD/DVD ISO

You can use VMware vSphere client to inject user data into the VM as an ISO image using the CD/DVD drive.

Follow these steps to provide user data using CD/DVD ISO:

1. Create a file with file name `userdata` that contains the preboot user data content. For more information on the content of the <NS-CONFIG> tag, see Sample <NS-CONFIG> section.

   > Note: File name must be strictly used as `userdata`.

2. Store the `userdata` file in a folder, and build an ISO image using the folder.

   You can build an ISO image with `userdata` file by the following two methods:

   - Using any image processing tool such as PowerISO.
   - Using `mkisofs` command in Linux.

   The following sample configuration shows how to generate an ISO image using the `mkisofs` command in Linux.

```
root@ubuntu:~/sai/14jul2021# ls -l total 4
drwxr-xr-x 2 root root 4096 Jul 14 12:32 esx_preboot_userdata
root@ubuntu:~/sai/14jul2021#
root@ubuntu:~/sai/14jul2021# ls -l esx_preboot_userdata/total 4
-rw-r--r-- 1 root root 3016 Jul 14 12:32 userdata
root@ubuntu:~/sai/14jul2021# mkisofs -o esx_preboot_userdata.iso
    ./esx_preboot_userdata
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
root@ubuntu:~/sai/14jul2021# ls -lh
total 356K
drwxr-xr-x 2 root root 4.0K Jul 14 12:32 esx_preboot_userdata
-rw-r--r-- 1 root root 352K Jul 14 12:34 esx_preboot_userdata.iso

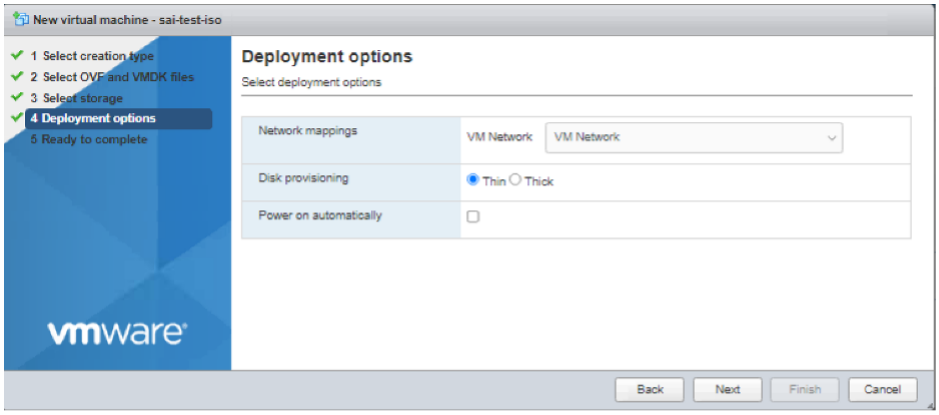
root@ubuntu:~/sai# ls preboot_userdata_155_193 userdata
root@ubuntu:~/sai# mkisofs -o preboot_userdata_155_193.iso ./
    preboot_userdata_155_193
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
```



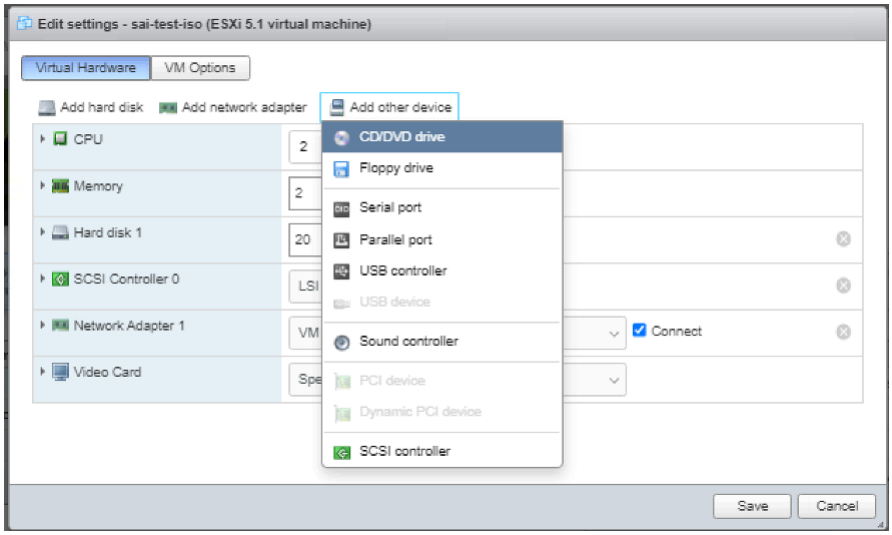
```
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)

<!--NeedCopy-->
```

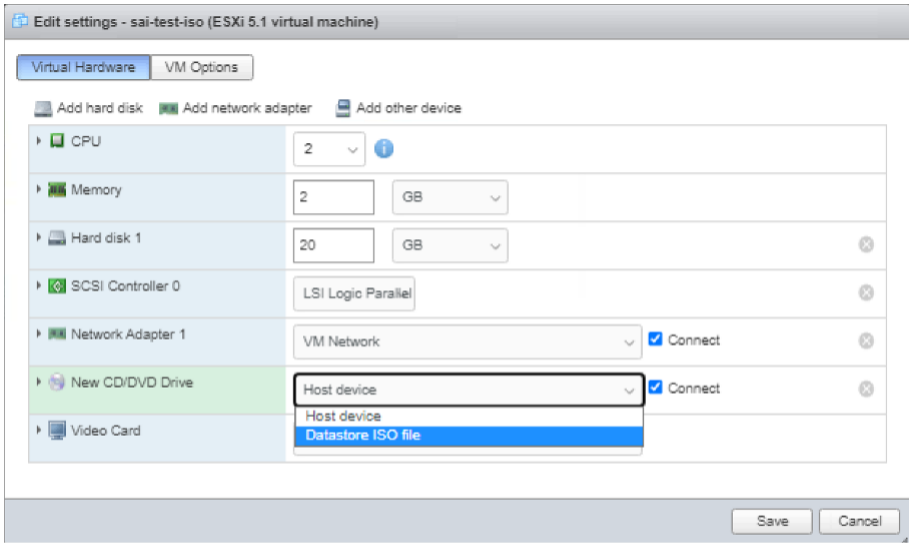
3. Provision the Citrix ADC VPX instance using standard deployment process to create the VM. But do not power on the VM automatically.



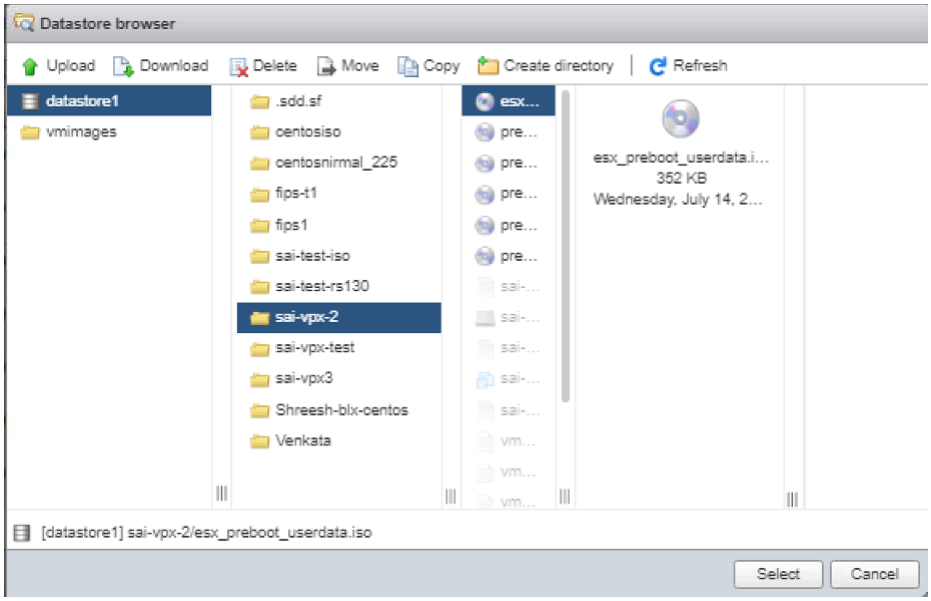
4. After the VM is successfully created, attach the ISO file as CD/DVD drive to the VM.



5. Navigate to **New CD/DVD Drive** and choose **Datastore ISO file** from the drop-down menu.



6. Select a Datastore in the vSphere Client.



7. Power on the VM.

Provide user data using OVF property

Follow these steps to provide user data using OVF property.

1. Create a file with user data content.


```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

2. Encode the user data content with Base64 encoding. You can perform the Base64 encoding using the following two methods:

   - In Linux, use the following command:

     ```
     base64 <userdata-filename>  >   <outuput-file>
     <!--NeedCopy-->
     ```

     Example:

     ```
     base64 esx_userdata.xml  > esx_userdata_b64
     <!--NeedCopy-->
     ```

     ```
     root@ubuntu:~/sai/14jul2021# base64 esx_userdata.xml > esx_userdata_b64
     root@ubuntu:~/sai/14jul2021#
     root@ubuntu:~/sai/14jul2021# cat esx_userdata_b64
     PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+CglhZGQgcm91dGUgMC4wLjAuMCAw
     LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KCiAgICA8TlMtQk9PVFNUUkFQPgog
     ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1CT09U
     U1RSQVA+CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
     VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAgICAg
     ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAgICAg
     ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5FVC1N
     QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
     RS1DT05GSUc+CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg==
     ```

   - Use online tools to encode user data content, for example, Base64 Encode and Decode.

3. Include a **Product** section in the OVF template of a Citrix ADC VPX instance on ESX hypervisor.

   Sample Product section:

   ```
   <ProductSection>

     <Info>Information about the installed software</Info>
     <Product>NSVPX-VSK Template</Product>
   ```



```
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```

4. Provide the base64 encoded user data as the `ovf:value` for `guestinfo.userdata` property in the Product section.

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
     userConfigurable="true"
    ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
       CglhZGQgcm91dGUgMC4wLjAuMCAw
    LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KICAgICA8TlMtQk9PVFNUUkFQ

    ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C

    U1RSQVA+
       CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT

    VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg

    ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGggIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg

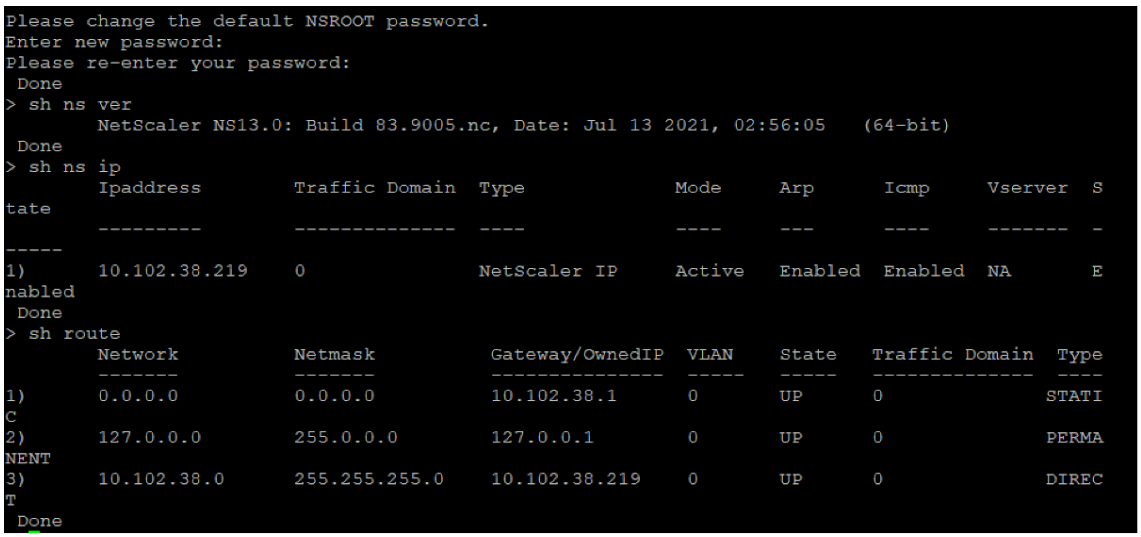
    ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgPFNVQk5FV

    QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
       CiAgICAgICAgPC9NR01ULULUlOVEVSRkFD
    RS1DT05GSUc+
       CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
```



```
            ==">
18
19      <Label>Userdata</Label>
20      <Description> Userdata for ESX VPX </Description>
21    </Property>
22
23  </ProductSection>
24  <!--NeedCopy-->
```

5. Use the modified OVF template with Product section for the VM deployment.



Install a Citrix ADC VPX instance on VMware cloud on AWS

September 14, 2021

The VMware Cloud (VMC) on AWS enables you to create cloud software-defined data centers (SDDC) on AWS with the desired number of ESX hosts. The VMC on AWS supports Citrix ADC VPX deployments. VMC provides a user interface same as on-prem vCenter. It functions identical to the ESX-based Citrix ADC VPX deployments.

Prerequisites

Before you begin installing a virtual appliance, do the following:

- One VMware SDDC must be present with at least one host.
- Download the Citrix ADC VPX appliance setup files.
- Create appropriate network segments on VMware SDDC to which the virtual machines connect.

- Obtain VPX license files. For more information about Citrix ADC VPX instance licenses, see the *Citrix ADC VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

VMware cloud hardware requirements

The following table lists the virtual computing resources that the VMware SDDC must provide for each VPX nCore virtual appliance.

Table 1. Minimum virtual computing resources required for running a Citrix ADC VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	In VMware SDDC, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher.
Disk space	20 GB

> Note
>
> This is in addition to any disk requirements for the hypervisor.

For production use of the VPX virtual appliance, the full memory allocation must be reserved.

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. The following table describes the minimum system requirements.

Table 2. Minimum system requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the “OVF Tool User Guide” PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
NIC	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.

Downloading the Citrix ADC VPX setup files

The Citrix ADC VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com. Click the **New Users link**, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

Citrix.com > **Downloads > Citrix ADC > Virtual Appliances**.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-79.64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-79.64.mf)

Install a Citrix ADC VPX instance on VMware cloud

After you have installed and configured VMware SDDC, you can use the SDDC to install virtual appliances on the VMware cloud. The number of virtual appliances that you can install depends on the amount of memory available on the SDDC.

To install Citrix ADC VPX instances on VMware cloud, follow these steps:

1. Open VMware SDDC on your workstation.
2. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click Login.
3. On the **File** menu, click **Deploy OVF Template**.
4. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the Citrix ADC VPX instance setup files, select the .ovf file, and click **Next**.

   > **Note:** By default, the Citrix ADC VPX instance uses E1000 network interfaces. To deploy ADC with the VMXNET3 interface, modify the OVF to use VMXNET3 interface instead of E1000.

5. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the VMware SDDC. Click **Next** to start installing a virtual appliance on VMware SDDC.

6. You are now ready to start the Citrix ADC VPX instance. In the navigation pane, select the Citrix ADC VPX instance that you have installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.
7. If you want to install another virtual appliance, repeat from Step 6.
8. Specify the management IP address from the same segment that is selected to be the management network. The same subnet is used for the Gateway.
9. The VMware SDDC requires that NAT and firewall rules are created explicitly for all private IP addresses belonging to network segments.

Install a Citrix ADC VPX instance on Microsoft Hyper-V server

September 14, 2021

To install Citrix ADC VPX instances on Microsoft Windows Server, you must first install Windows Server, with the Hyper-V role enabled, on a machine with adequate system resources. While installing the Hyper-V role, be sure to specify the NICs on the server that Hyper-V uses to create the virtual networks. You can reserve some NICs for the host. Use Hyper-V Manager to perform the Citrix ADC VPX instance installation.

Citrix ADC VPX instance for Hyper-V is delivered in virtual hard disk (VHD) format. It includes the default configuration for elements such as CPU, network interfaces, and hard-disk size and format. After you install Citrix ADC VPX instance, you can configure the network adapters on virtual appliance, add virtual NICs, and then assign the Citrix ADC IP address, subnet mask, and gateway, and complete the basic configuration of the virtual appliance.

After the initial configuration of the VPX instance, if you want to upgrade the appliance to the latest software release, see Upgrade a Citrix ADC VPX standalone appliance

> **Note**
>
> Intermediate System-to-Intermediate System (ISIS) protocol is not supported on the Citrix ADC VPX virtual appliance hosted on the HyperV-2012 platform.

Prerequisites for installing Citrix ADC VPX instance on Microsoft servers

Before you begin installing a virtual appliance, do the following:

- Enable the Hyper-V role on Windows Servers. For more information, see http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspx.
- Download the virtual appliance setup files.

- Obtain Citrix ADC VPX instance license files. For more information about Citrix ADC VPX instance licenses, see the *Citrix ADC VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

Microsoft server hardware requirements

The following table describes the minimum system requirements for Microsoft servers.

Table 1. Minimum system requirements for Microsoft servers

Component	Requirement
CPU	1.4 GHz 64-bit processor
RAM	8 GB
Disk Space	32 GB or greater

The following table lists the virtual computing resources for each Citrix ADC VPX instance.

Table 2. Minimum virtual computing resources required for running a Citrix ADC VPX instance

Component	Requirement
RAM	4 GB
Virtual CPU	2
Disk Space	20 GB
Virtual Network Interfaces	1

Download the Citrix ADC VPX setup files

The Citrix ADC VPX instance for Hyper-V is delivered in virtual hard disk (VHD) format. You can download the files from the Citrix website. You need a Citrix account to log in. If you do not have a Citrix account, access the home page at http://www.citrix.com, click **Sign In > My account > Create Citrix Account**, and follow the instructions to create a Citrix account.

To download the Citrix ADC VPX instance setup files, follow these steps:

1. In a web browser, go to http://www.citrix.com/.
2. Sign in with your user name and password.
3. Click **Downloads**.

4. In **Select a Product** drop-down menu, select **Citrix ADC (NetScaler ADC)**.
5. Under **Citrix ADC Release X.X > Virtual Appliances**, click **Citrix ADC VPX Release X.X**
6. Download the compressed file to your server.

Install the Citrix ADC VPX instance on Microsoft servers

After you have enabled the Hyper-V role on Microsoft Server and extracted the virtual appliance files, you can use Hyper-V Manager to install Citrix ADC VPX instance. After you import the virtual machine, you need to configure the virtual NICs by associating them to the virtual networks created by Hyper-V.

You can configure a maximum of eight virtual NICs. Even if the physical NIC is DOWN, the virtual appliance assumes that the virtual NIC is UP, because it can still communicate with the other virtual appliances on the same host (server).

> **Note**
>
> You cannot change any settings while the virtual appliance is running. Shut down the virtual appliance and then make changes.

To install Citrix ADC VPX instance on Microsoft Server by using Hyper-V Manager:

1. To start Hyper-V Manager, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the navigation pane, under **Hyper-V Manage**r, select the server on which you want to install Citrix ADC VPX instance.
3. On the **Action** menu, click **Import Virtual Machin**e.
4. In the **Import Virtual Machine** dialog box, in **Location**, specify the path of the folder that contains the Citrix ADC VPX instance software files, and then select **Copy the virtual machine (create a new unique ID)**. This folder is the parent folder that contains the Snapshots, Virtual Hard Disks, and Virtual Machines folders.
5. Note: If you received a compressed file, make sure that you extract the files into a folder before you specify the path to the folder.
6. Click **Import**.
7. Verify that the virtual appliance that you imported is listed under **Virtual Machines**.
8. To install another virtual appliance, repeat steps **2** through **6**.

> **Important**
>
> Make sure that you extract the files to a different folder in step **4**.

Auto-provision a Citrix ADC VPX instance on Hyper-V

Auto-provisioning of Citrix ADC VPX instance is optional. If auto-provisioning is not done, the virtual appliance provides an option to configure the IP address and so on.

To auto-provision Citrix ADC VPX instance on Hyper-V, follow these steps.

1. Create an ISO9660 compliant ISO image using the xml file as depicted in the example. Make sure that the name of the xml file is **userdata**.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe=`"http://schemas.dmtf.org/ovf/environment/1`
    "

xmlns:xsi=`"http://www.w3.org/2001/XMLSchema-instance`"

oe:id=""

xmlns=`"http://schemas.dmtf.org/ovf/environment/1`">

<PlatformSection>

<Kind>HYPER-V</Kind>

<Version>2013.1</Version>

<Vendor>CISCO</Vendor>

<Locale>en</Locale>

</PlatformSection>

<PropertySection>

<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
    />

<Property oe:key="com.citrix.netscaler.platform" oe:value="NS1000V
    "/>

<Property oe:key="com.citrix.netscaler.orch\_env" oe:value="cisco-
    orch-env"/>

<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="
    10.102.100.122"/>

<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
    255.255.255.128"/>

```



```
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="
    10.102.100.67"/></PropertySection>

</Environment>
<!--NeedCopy-->
```

2. Copy the ISO image to hyper-v server.
3. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**. You can also select the virtual appliance and then right click and select **Settings**. The **Settings** window for the selected virtual appliance is displayed.
4. In the **Settings** window, under the hardware section, click **IDE Controller**.
5. In the right window pane, select **DVD Drive** and click **Add**. The DVD Drive is added under the **IDE Controller** section in the left window pane.
6. Select the **DVD Drive** added in step 5. In the right window pane, select the **Image file radio** button and click **Browse** and select the ISO image that you copied on Hyper-V server, in step 2.
7. Click **Apply**.

> **Note**
>
> The virtual appliance instance comes up in the default IP address, when:
>
> - The DVD drive is attached and the ISO file is not provided.
> - The ISO file does not include the user data file.
> - The user data file name or format is not correct.

To configure virtual NICs on the Citrix ADC VPX instance, follow these steps:

1. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**.
2. In the **Settings for <virtual appliance name>** dialog box, click **Add Hardware** in the left pane.
3. In the right pane, from the list of devices, select **Network Adapter**.
4. Click **Add**.
5. Verify that **Network Adapter (not connected)** appears in the left pane.
6. Select the network adapter in the left pane.
7. In the right pane, from the **Network** menu, select the virtual network to connect the adapter to.
8. To select the virtual network for other network adapters that you want to use, repeat steps **6** and **7**.
9. Click **Apply**, and then click **OK**.

To configure the Citrix ADC VPX instance:

1. Right-click the virtual appliance that you previously installed, and then select **Start**.
2. Access the console by double-clicking the virtual appliance.
3. Type the Citrix ADC IP address, subnet mask, and gateway for your virtual appliance.

You have completed the basic configuration of your virtual appliance. Type the IP address in a Web browser to access the virtual appliance.

> **Note**
>
> You can also use virtual machine (VM) template to provision Citrix ADC VPX instance using SCVMM.
>
> If you use Microsoft Hyper-V NIC teaming solution with NetScaler VPX instances, see article CTX224494 for more information.

Install a Citrix ADC VPX instance on Linux-KVM platform

September 14, 2021

To set up a Citrix ADC VPX for the Linux-KVM platform, you can use the graphical Virtual Machine Manager (Virtual Manager) application. If you prefer the Linux-KVM command line, you can use the `virsh` program.

The host Linux operating system must be installed on suitable hardware by using virtualization tools such as KVM Module and QEMU. The number of virtual machines (VMs) that can be deployed on the hypervisor depends on the application requirement and the chosen hardware.

After you provision a Citrix ADC VPX instance, you can add more interfaces.

Limitations and usage guidelines

General recommendations

To avoid unpredictable behavior, apply the following recommendations:

- Do not change the MTU of the VNet interface associated with the VPX VM. Shut down the VPX VM before modifying any configuration parameters, such as Interface modes or CPU.
- Do not force a shutdown of the VPX VM. That is, do not use the **Force off** command.
- Any configurations done on the host Linux might or might not be persistent, depending on your Linux distribution settings. You can choose to make these configurations persistent to ensure consistent behavior across reboots of host Linux operating system.
- The Citrix ADC package has to be unique for each of the Citrix ADC VPX instance provisioned.

Limitations

- Live migration of a VPX instance that runs on KVM is not supported.

Prerequisites for installing a Citrix ADC VPX instance on Linux-KVM platform

September 14, 2021

Check the minimum system requirements for a Linux-KVM server running on a Citrix ADC VPX instance.

CPU requirement:

- 64-bit x86 processors with the hardware virtualization feature included in Intel VT-X processors.

To test whether your CPU supports the Linux host, enter the following command at the host Linux shell prompt:

```
1 *.egrep '^flags.*(vmx|svm)' /proc/cpuinfo*
2 <!--NeedCopy-->
```

If the **BIOS** settings for the preceding extension are disabled, you must enable them in the BIOS.

- Provide at least 2 CPU cores to Host Linux.
- There is no specific recommendation for processor speed, but higher the speed, the better the performance of the VM application.

Memory (RAM) requirement:

Minimum 4 GB for the host Linux kernel. Add more memory as required by the VMs.

Hard disk requirement:

Calculate the space for Host Linux kernel and VM requirements. A single Citrix ADC VPX VM requires 20 GB of disk space.

Software requirements

The Host kernel used must be a 64-bit Linux kernel, release 2.6.20 or later, with all virtualization tools. Citrix recommends newer kernels, such as 3.6.11-4 and later.

Many Linux distributions such as Red Hat, CentOS, and Fedora, have tested kernel versions and associated virtualization tools.

Guest VM hardware requirements

Citrix ADC VPX supports IDE and virtIO hard disk type. The Hard Disk Type has been configured in the XML file, which is a part of the Citrix ADC package.

Networking requirements

Citrix ADC VPX supports virtIO para-virtualized, SR-IOV, and PCI Passthrough network interfaces.

For more information about the supported network interfaces, see:

- Provision the Citrix ADC VPX instance by using the Virtual Machine Manager
- Configure a Citrix ADC VPX instance to use SR-IOV network interfaces
- Configure a Citrix ADC VPX instance to use PCI passthrough network interfaces

Source Interface and Modes

The source device type can be either Bridge or MacVTap. In MacVTap, four modes are possible - VEPA, Bridge, Private, and Pass-through. Check the types of interfaces that you can use and the supported traffic types, as per the following:

Bridge:

- Linux Bridge.
- `Ebtables` and `iptables` settings on host Linux might filter the traffic on the bridge if you do not choose the correct setting or disable `IPtable` services.

MacVTap (VEPA mode):

- Better performance than a bridge.
- Interfaces from the same lower device can be shared across the VMs.
- Inter-VM communication using the same
- lower device is possible only if the upstream or downstream switch supports VEPA mode.

MacVTap (private mode):

- Better performance than a bridge.
- Interfaces from the same lower device can be shared across the VMs.
- Inter-VM communication using the same lower device is not possible.

MacVTap (bridge mode):

- Better as compared to bridge.
- Interfaces out of the same lower device can be shared across the VMs.
- Inter-VM communication using the same lower device is possible, if the lower device link is UP.

MacVTap (Pass-through mode):

- Better as compared to bridge.
- Interfaces out of the same lower device cannot be shared across the VMs.
- Only one VM can use the lower device.

> Note: For best performance by the VPX instance, ensure that the `gro` and `lro` capabilities are switched off on the source interfaces.

Properties of source interfaces

Make sure that you switch off the generic-receive-offload (`gro`) and large-receive-offload (`lro`) capabilities of the source interfaces. To switch off the `gro` and `lro` capabilities, run the following commands at the host Linux shell prompt.

```
ethtool -K eth6 gro off
ethool -K eth6 lro off
```

Example:

```
    [root@localhost ~]# ethtool -K eth6

                    Offload parameters for eth6:

                                        rx-checksumming: on

                                        tx-checksumming: on

                    scatter-gather: on

                    tcp-segmentation-offload: on

                    udp-fragmentation-offload: off

                    generic-segmentation-offload: on

                    generic-receive-offload: off

                    large-receive-offload: off

                    rx-vlan-offload: on

                    tx-vlan-offload: on

                    ntuple-filters: off

                    receive-hashing: on

    [root@localhost ~]#
<!--NeedCopy-->
```


Example:

If the host Linux bridge is used as a source device, as in the following example, and `lro` capabilities must be switched off on the VNet interfaces, which are the virtual interfaces connecting the host to the guest VMs.

```
    [root@localhost ~]# brctl show eth6_br

    bridge name     bridge  id               STP enabled interfaces

    eth6_br         8000.00e0ed1861ae          no         eth6

                                                          vnet0

                                                          vnet2

    [root@localhost ~]#
<!--NeedCopy-->
```

In the preceding example, the two virtual interfaces are derived from the eth6_br and are represented as vnet0 and vnet2. Run the following commands to switch off `gro` and `lro` capabilities on these interfaces.

```
    ethtool -K vnet0 gro off
            ethtool -K vnet2 gro off
            ethtool -K vnet0 lro off
                     ethtool -K vnet2 lro off
<!--NeedCopy-->
```

Promiscuous mode

The promiscuous mode must be enabled for the following features to work:

- L2 mode
- Multicast traffic processing
- Broadcast
- IPV6 traffic
- virtual MAC
- Dynamic routing

Use the following command to enable the promiscuous mode.

```
[root@localhost ~]# ifconfig eth6 promisc
[root@localhost ~]# ifconfig eth6
eth6       Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3
```



```
            inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link
            UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1
            RX packets:142961 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)

[root@localhost ~]#
<!--NeedCopy-->
```

Module required

For better network performance, make sure the vhost_net module is present in the Linux host. To check the existence of vhost_net module, run the following command on the Linux host:

```
lsmod | grep "vhost\_net"
<!--NeedCopy-->
```

If vhost_net is not yet running, enter the following command to run it:

```
modprobe vhost\_net
<!--NeedCopy-->
```

Provision the Citrix ADC VPX instance by using OpenStack

September 14, 2021

You can provision a Citrix ADC VPX instance in an OpenStack environment either by using the **Nova boot** command (OpenStack CLI) or Horizon (OpenStack dashboard) .

Provisioning a VPX instance, optionally involves using data from the config drive. Config drive is a special configuration drive that attaches to the instance as a CD-ROM device when it boots. This configuration drive can be used to pass networking configuration such as management IP address, network mask, default gateway, and to inject customer scripts.

In a Citrix ADC appliance, the default authentication mechanism is password based. Now, the SSH key-pair authentication mechanism is supported for Citrix ADC VPX instances on the OpenStack environment.

The key-pair (public key and private key) is generated before using the Public Key Cryptography mechanism. You can use different mechanisms, such as Horizon, Puttygen.exe for Windows, and `ssh-`

keygen for the Linux environment, to generate the key pair. Refer to online documentation of respective mechanisms for more information about generating key pair.

Once a key pair is available, copy the private key to a secure location to which authorized persons have access. In OpenStack, public key can be deployed on a VPX instance by using the Horizon or Nova boot command. When a VPX instance is provisioned by using OpenStack, it first detects that the instance is booting in an OpenStack environment by reading a specific BIOS string. This string is "OpenStack Foundation" and for Red Hat Linux distributions it is stored in /etc/nova/release. This is a standard mechanism that is available in all OpenStack implementations based on the KVM hypervisor platform. The drive must have a specific OpenStack label.

If the config drive is detected, the instance attempts to read the network configuration, custom scripts, and SSH key pair if provided.

User data file

The Citrix ADC VPX instance uses a customized OVF file, also known as the user data file, to inject network configuration, custom scripts. This file is provided as part of config drive. Here is an example of a customized OVF file.

```
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1"
xmlns:cs="http://schemas.citrix.com/openstack">
<PlatformSection>
<Kind></Kind>
<Version>2016.1</Version>
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
<Property oe:key="com.citrix.netscaler.orch_env" oe:value="openstack-
   orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
   255.255.255.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1
   "/>
</PropertySection>
```

 
```

```
<cs:ScriptSection>
  <cs:Version>1.0</cs:Version>
    <ScriptSettingSection xmlns="http://schemas.citrix.com/openstack"
       xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Scripts>
          <Script>
                <Type>shell</Type>
                <Parameter>X Y</Parameter>
               <Parameter>Z</Parameter>
               <BootScript>before</BootScript>
                 <Text>
                       #!/bin/bash
                       echo "Hi, how are you" $1 $2 >> /var/sample.txt
                 </Text>
          </Script>
          <Script>
                <Type>python</Type>
                <BootScript>after</BootScript>
                 <Text>
                     #!/bin/python
print("Hello");
                 </Text>
          </Script>
  <Script>
                <Type>perl</Type>
                <BootScript>before</BootScript>
                 <Text>
                     !/usr/bin/perl
my $name = "VPX";
print "Hello, World $name !\n" ;
                 </Text>
          </Script>
          <Script>
               <Type>nscli</Type>
               <BootScript>after</BootScript>
               <Text>
                add vlan 33
bind vlan 33 -ifnum 1/2
               </Text>
          </Script>
        </Scripts>
    </ScriptSettingSection>
</cs:ScriptSection>
</Environment>
<!--NeedCopy--> ```
```


In the OVF file preceding “PropertySection” is used for NetScaler networking configuration while <cs:ScriptSection> is used to enclose all scripts. <Scripts></Scripts> tags are used to bundle all scripts together. Each script is defined in between <Script> </Script> tags. Each script tag has following fields/tags:

a) <Type>: Specifies value for script type. Possible values: Shell/Perl/Python/NSLCI (for NetScaler CLI scripts)

b) <Parameter>: Provides parameters to the script. Each script can have multiple <Parameter> tags.

c) <BootScript>: Specifies script execution point. Possible values for this tag: before/after. “before” specifies script is run before PE comes up. “after” specifies that the script will be run after PE comes up.

d) <Text>: Pastes content of a script.

> **Note**
>
> Currently the VPX instance does not take care of sanitization of scripts. As an administrator, you must check the validity of the script.
>
> Not all sections need to be present. Use an empty “PropertySection” to only define scripts to run on first boot or an empty <cs:ScriptSection> to only define networking configuration.
>
> After the required sections of the OVF file (user data file) are populated, use that file to provision the VPX instance.

Network configuration

As part of the network configuration, the VPX instance reads:

- Management IP address
- Network mask
- Default gateway

After the parameters are successfully read, they are populated in the NetScaler configuration, to allow managing the instance remotely. If the parameters are not read successfully or the config drive is not available, the instance transitions to the default behavior, which is:

- The instance attempts to retrieve the IP address information from DHCP.
- If DHCP fails or times-out, the instance comes up with the default network configuration (192.168.100.1/16).

Customer script

The VPX instance allows to run a custom script during initial provisioning. The appliance supports script of type Shell, Perl, Python, and Citrix ADC CLI commands.

SSH key pair authentication

The VPX instance copies public key, available within the configuration drive as part of instance meta data, into its “authorized_keys” file. This allows the user to access the instance with private key.

> **Note**
>
> When an SSH key is provided, the default credentials (nsroot/nsroot) no longer work, if password-based access is needed, log on with the respective SSH private key and manually set a password.

Before you begin

Before you provision a VPX instance on OpenStack environment, extract the `.qcow2` file from the .tgz file and build

An OpenStack image from the qcow2 image. Follow these steps:

1. Extract the `.qcow2` file from the `.tqz` file by typing the following command

```
tar xvzf <TAR file>
tar xvzf <NSVPX-KVM-12.0-26.2_nc.tgz>
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```

2. Build an OpenStack image using the `.qcoz2` file extracted in step 1 by typing the following command.

```
openstack image create --container-format bare --property
   hw_disk_bus=ide --disk-format qcow2 --file <path to qcow2 file>
    --public <name of the OpenStack image>

glance image-create --name="NS-VPX-12-0-26-2" --property
   hw_disk_bus=ide --ispublic=
true --container-format=bare --disk-format=qcow2< NSVPX-KVM
   -12.0-26.2_nc.qcow2
```

Figure 1: The following illustration provides a sample output for the glance image-create command.


```
+------------------+--------------------------------------------------------+
| Field            | Value                                                  |
+------------------+--------------------------------------------------------+
| checksum         | 154ade3fc7dca7d1706b1d03d7d97552                       |
| container_format | bare                                                   |
| created_at       | 2017-03-13T08:52:31Z                                   |
| disk_format      | qcow2                                                  |
| file             | /v2/images/322c1e0f-cce8-4b7b-b53e-bd8152c388ed/file   |
| id               | 322c1e0f-cce8-4b7b-b53e-bd8152c388ed                   |
| min_disk         | 0                                                      |
| min_ram          | 0                                                      |
| name             | VPX-KVM-12.0-26.2                                      |
| owner            | 58d17d81df5d4406afbb4fdab3a58d79                       |
| properties       | hw_disk_bus='ide'                                      |
| protected        | False                                                  |
| schema           | /v2/schemas/image                                      |
| size             | 784338944                                              |
| status           | active                                                 |
| updated_at       | 2017-03-13T08:52:43Z                                   |
| virtual_size     | None                                                   |
| visibility       | public                                                 |
+------------------+--------------------------------------------------------+
```

Provisioning the VPX instance

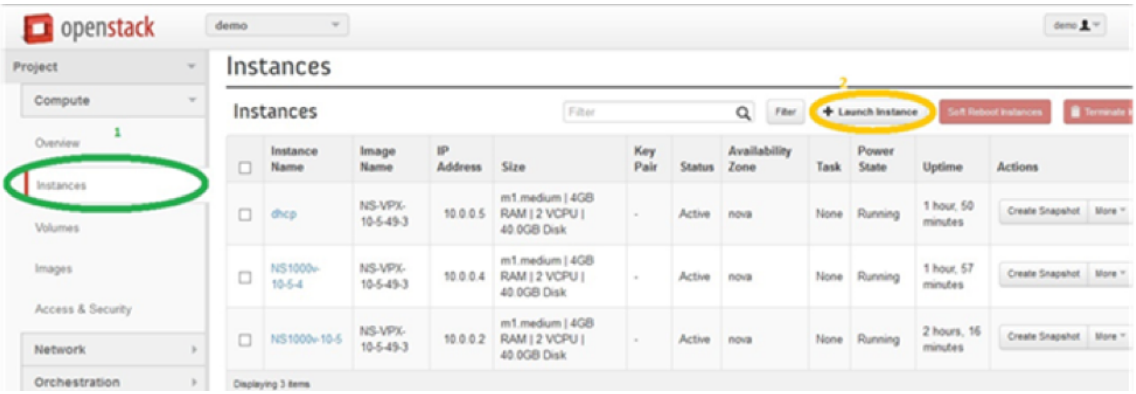
You can provision a VPX instance in two ways by using one of the options:

- Horizon (OpenStack dashboard)
- Nova boot command (OpenStack CLI)

Provision a VPX instance by using the OpenStack dashboard

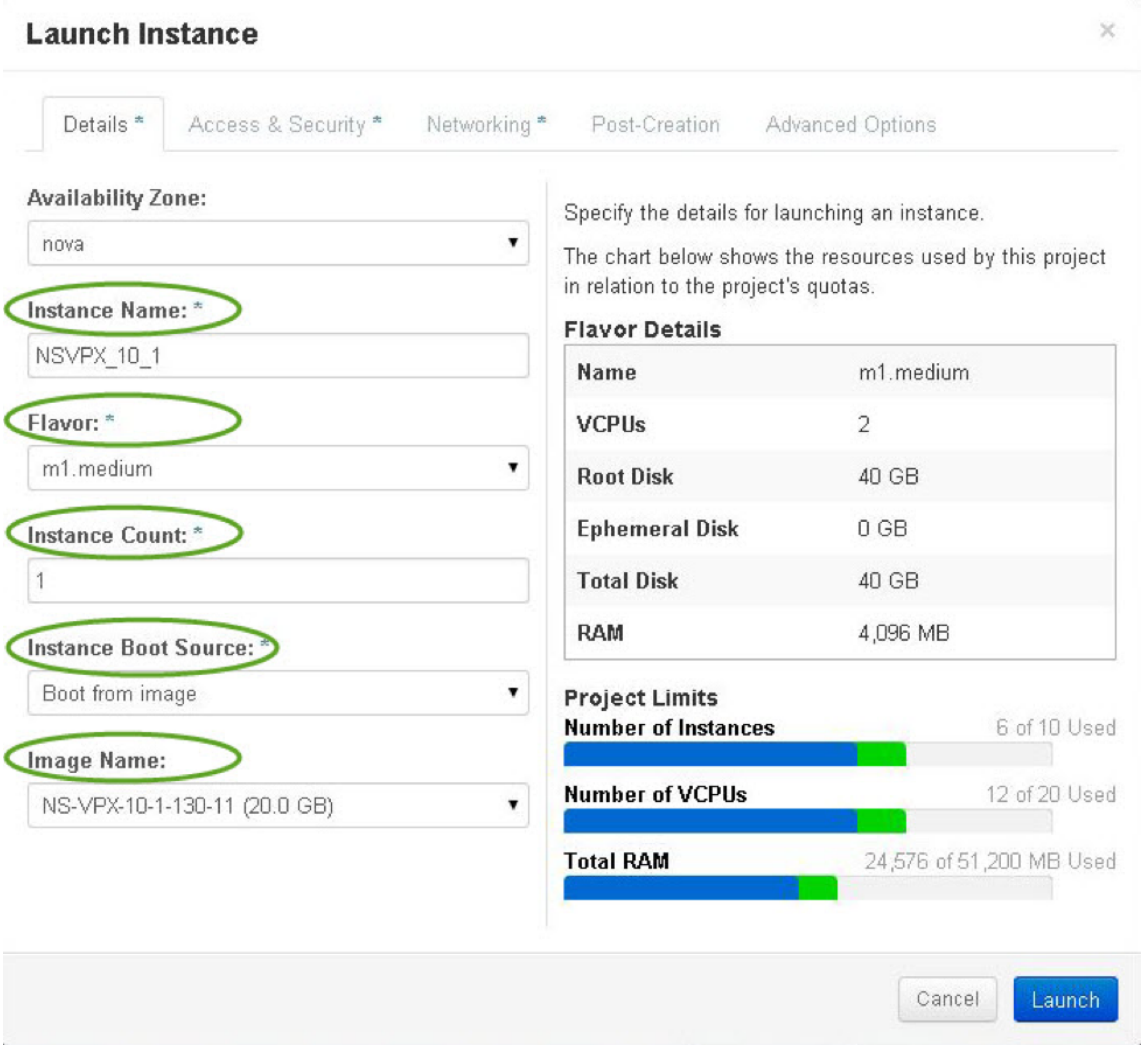
Follow these steps to provision the VPX instance by using Horizon:

1. Log on to the OpenStack dashboard.
2. In the Project panel on the left hand side of the dashboard, select **Instances**.
3. In the Instances panel, click **Launch Instance** to open the Instance Launching wizard.



4. In the Launch Instance wizard, fill in the details, like:

a) Instance Name
b) Instance Flavor
c) Instance Count
d) Instance Boot Source
e) Image Name



5. Deploy a new key pair or an existing key pair through Horizon by completing the following steps:

   a) If you don't have an existing key pair, create the key by using any existing mechanisms. If you've an existing key, skip this step.

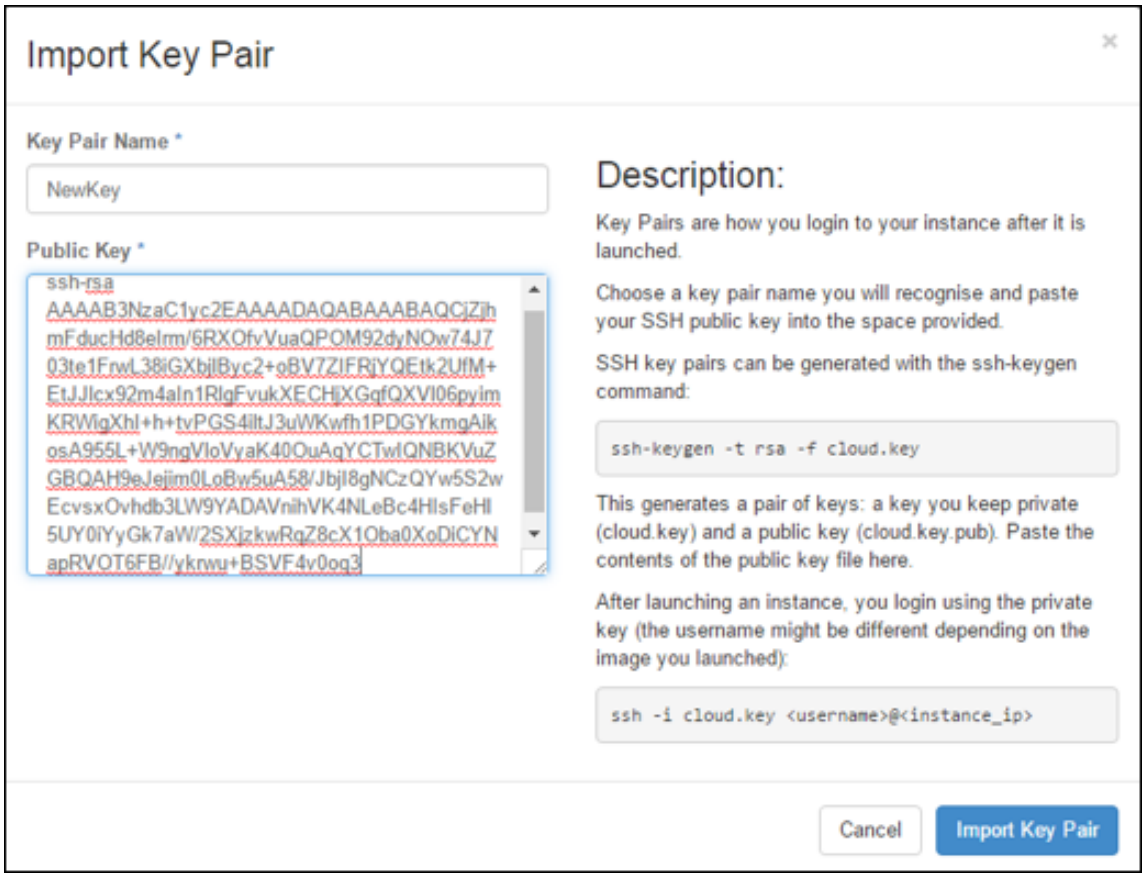
   b) Copy the content of public key.

   c) Go to **Horizon > Instances > Create New Instances**.

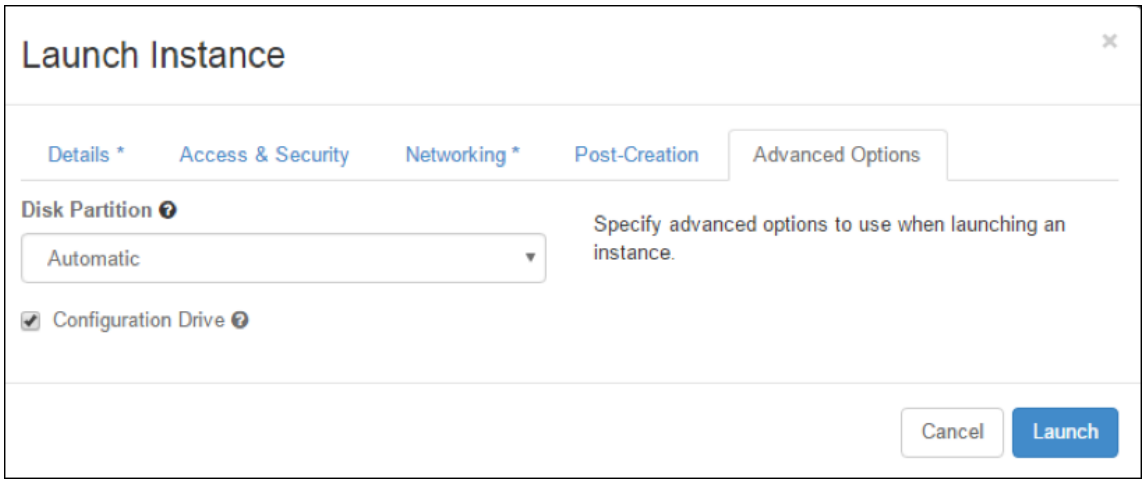
   d) Click **Access & Security**.

   e) Click the + sign next to the **Key Pair** drop-down menu and provide values for shown parameters.

f) Paste public key content in *Public key* box, give a name to the key and click **Import Key Pair**.



6. Click the **Post Creation** tab in the wizard. In Customization Script, add the content of the user data file. The user data file contains the IP address, Netmask and Gateway details, and customer scripts of the VPX instance.
7. After a key pair is selected or imported, check config-drive option and click **Launch**.



Provision the VPX instance by using OpenStack CLI

Follow these steps to provision a VPX instance by using OpenStack CLI.

1. To create an image from qcow2, type the following command:

   ```
   openstack image create --container-format bare --property hw_disk_bus=
   ide --diskformat qcow2 --file NSVPX-OpenStack.qcow2 --public VPX-ToT-
   Image
   ```

2. To select an image for creating an instance, type the following command:

   ```
   openstack image list | more
   ```

3. To create an instance of a particular flavor, type the following command to choose a flavor ID/-Name of from a list:

   ```
   openstack flavor list
   ```

4. To attach a NIC to a particular network, type the following command to choose a network ID from a network list:

   ```
   openstack network list
   ```

5. To create an instance, type the following command:

   ```
   1 openstack server create --flavor FLAVOR_ID --image IMAGE_ID --key-
       name KEY_NAME
   2 --user-data USER_DATA_FILE_PATH --config-drive True --nic net-id=
       net-uuid
   3 INSTANCE_NAME
   4 openstack server create --image VPX-ToT-Image --flavor m1.medium
       --user-data
   5 ovf.xml --config-drive True --nic net-id=2734911b-ee2b-48d0-a1b6-3
       efd44b761b9
   6 VPX-ToT
   ```

Figure 2: The following illustration provides a sample output.


```
+--------------------------------------+--------------------------------------------------------------------------+
| Field                                | Value                                                                    |
+--------------------------------------+--------------------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                                   |
| OS-EXT-AZ:availability_zone          |                                                                          |
| OS-EXT-SRV-ATTR:host                 | None                                                                     |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | None                                                                     |
| OS-EXT-SRV-ATTR:instance_name        | instance-000001c2                                                        |
| OS-EXT-STS:power_state               | 0                                                                        |
| OS-EXT-STS:task_state                | scheduling                                                               |
| OS-EXT-STS:vm_state                  | building                                                                 |
| OS-SRV-USG:launched_at               | None                                                                     |
| OS-SRV-USG:terminated_at             | None                                                                     |
| accessIPv4                           |                                                                          |
| accessIPv6                           |                                                                          |
| addresses                            |                                                                          |
| adminPass                            | pFVvMtq7N8Z6                                                             |
| config_drive                         | True                                                                     |
| created                              | 2017-03-13T10:32:59Z                                                     |
| flavor                               | m1.medium (3)                                                            |
| hostId                               |                                                                          |
| id                                   | a1fe991e-3604-43a0-9dd6-59fa0f3749df                                     |
| image                                | VPX-ToT-Image (f0c2f9d1-08f2-4b2e-9943-2ee6bc2edbc7)                     |
| key_name                             | None                                                                     |
| name                                 |VPX-ToT                                                                   |
| os-extended-volumes:volumes_attached | []                                                                       |
| progress                             | 0                                                                        |
| project_id                           | 58d17d81df5d4406afbb4fdab3a58d79                                         |
| properties                           |                                                                          |
| security_groups                      | [{u'name': u'default'}]                                                  |
| status                               | BUILD                                                                    |
| updated                              | 2017-03-13T10:33:00Z                                                     |
| user_id                              | a6347b33916b4eb1b1f76360a9c8f935                                         |
+--------------------------------------+--------------------------------------------------------------------------+
```

Provision the Citrix ADC VPX instance by using the Virtual Machine Manager

September 14, 2021

The Virtual Machine Manager is a desktop tool for managing VM guests. It enables you to create new VM guests and various types of storage, and manage virtual networks. You can access the graphical console of VM guests with the built-in VNC viewer and view performance statistics, either locally or remotely.

After installing your preferred Linux distribution, with KVM virtualization enabled, you can proceed with provisioning virtual machines.

While using the Virtual Machine Manager to provision a Citrix ADC VPX instance, you have two options:

- Enter the IP address, gateway, and netmask manually
- Assign the IP address, gateway, and netmask automatically (auto-provisioning)

You can use two kinds of images to provision a Citrix ADC VPX instance:

- RAW
- QCOW2

You can convert a Citrix ADC VPX RAW image to a QCOW2 image and provision the Citrix ADC VPX instance. To convert the RAW image to a QCOW2 image, type the following command:


```
qemu-img convert -O qcow2 original-image.raw image-converted.qcow
```

For example:

```
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5_nc
.qcow
```

A typical Citrix ADC VPX deployment on KVM includes the following steps:

- Checking prerequisites for Auto-Provisioning a Citrix ADC VPX Instance
- Provisioning the Citrix ADC VPX Instance by Using a RAW Image
- Provisioning the Citrix ADC VPX Instance by Using a QCOW2 Image
- Adding Additional Interfaces to a VPX Instance by using Virtual Machine Manager

Check prerequisites for auto-provisioning a Citrix ADC VPX instance

Auto-provisioning is an optional feature, and it involves using data from the CDROM drive. If this feature is enabled, you need not enter the management IP address, network mask, and default gateway of the Citrix ADC VPX instance during initial setup.

You need to complete the following tasks before you can auto-provision a VPX instance:

1. Create a customized Open Virtualization Format (OVF) XML file or user data file.
2. Convert the OVF file into an ISO image by using an online application (for example PowerISO).
3. Mount the ISO image on the KVM host by using any secure copy (SCP)-based tools.

Sample OVF XML file:

Here's is an example of the contents an OVF XML file, which you can use as a sample to create your file.

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2
3  <Environment xmlns:oe="`http://schemas.dmtf.org/ovf/environment/1"`
4
5  xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance"`
6
7  oe:id=""
8
9  xmlns="`http://schemas.dmtf.org/ovf/environment/1"`
10
11 xmlns:cs="`http://schemas.citrix.com/openstack">`
12
13 <PlatformSection>
14
15 <Kind></Kind>
16
17 <Version>2016.1</Version>
```



```
18
19 <Vendor>VPX</Vendor>
20
21 <Locale>en</Locale>
22
23 </PlatformSection>
24
25 <PropertySection>
26
27 <Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
28
29 <Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
30
31 <Property oe:key="com.citrix.netscaler.orch\_env" oe:value="KVM"/>
32
33 <Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
34
35 <Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
      255.255.255.0"/>
36
37 <Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1
      "/>
38
39 </PropertySection>
40
41 </Environment>
42 <!--NeedCopy-->
```

In the OVF XML file preceding, “PropertySection” is used for NetScaler networking configuration. When you create the file, specify values for the parameters that are highlighted at the end of the example:

- Management IP address
- Netmask
- Gateway

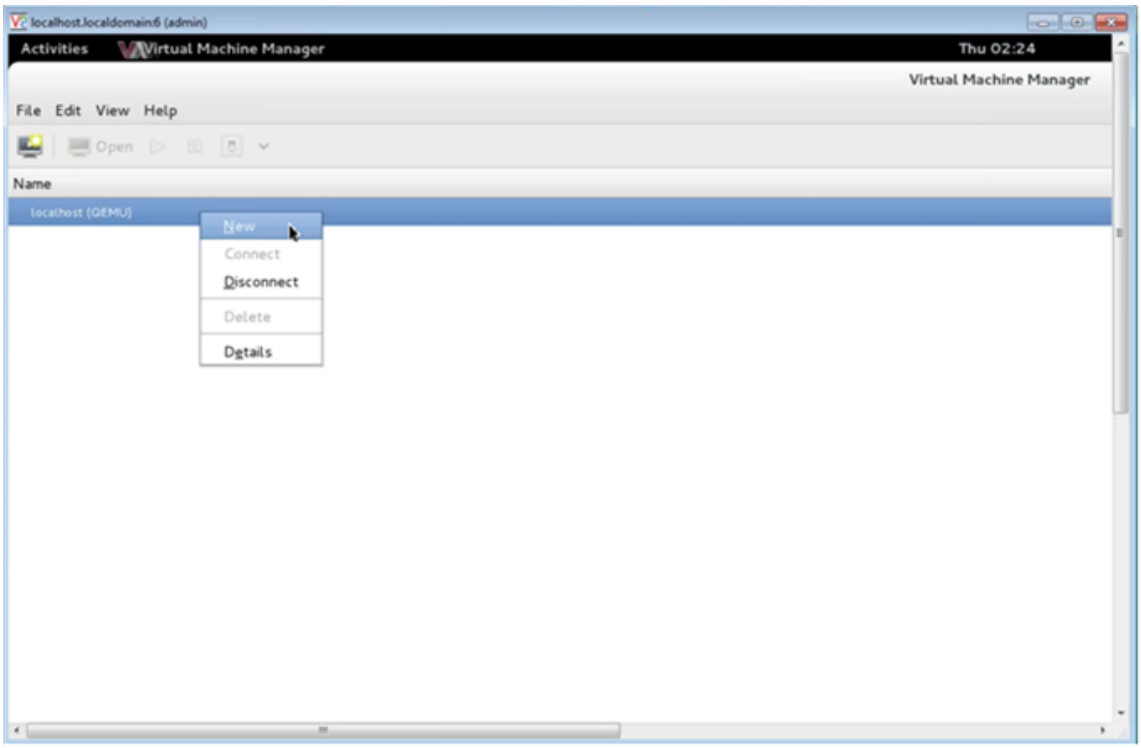
> **Important**
>
> If the OVF file is not properly XML formatted, the VPX instance is assigned the default network configuration, not the values specified in the file.

Provision the Citrix ADC VPX instance by using a RAW image

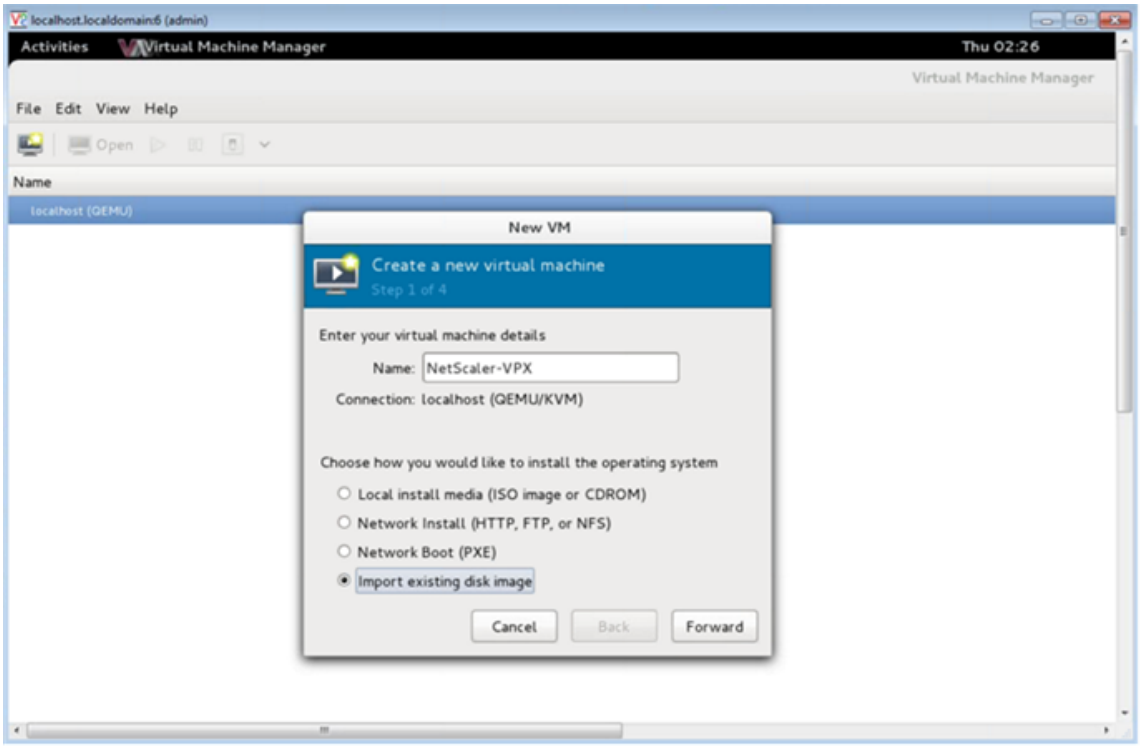
The Virtual Machine Manager enables you to provision a Citrix ADC VPX instancy by using a RAW image.

To provision a Citrix ADC VPX instance by using the Virtual Machine Manager, follow these steps:

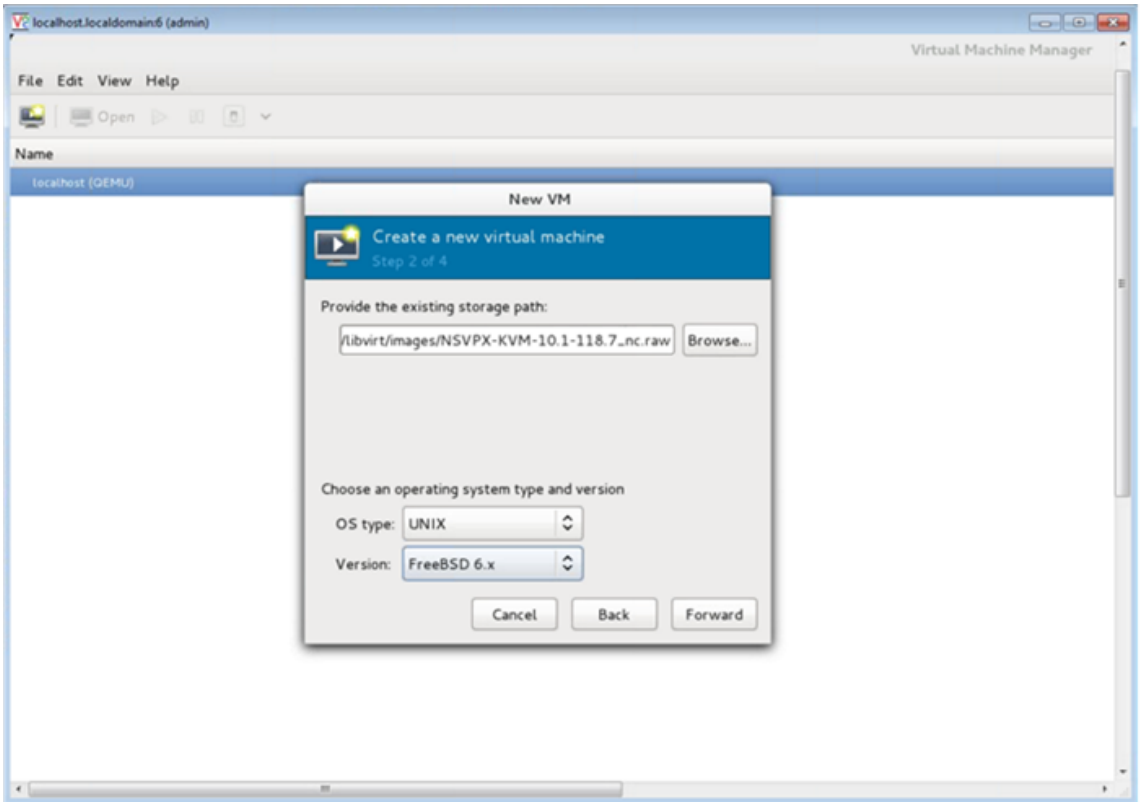
1. Open the Virtual Machine Manager (**Application** > **System Tools** > **Virtual Machine Manager**) and enter the logon credentials in the **Authenticate** window.
2. Click the  icon or right-click **localhost (QEMU)** to create a new Citrix ADC VPX instance.



3. In the **Name** text box, enter a name for the new VM (for example, NetScaler-VPX).
4. In the **New VM** window, under “Choose how you would like to install the operating system,” select **Import existing disk image**, and then and click **Forward**.

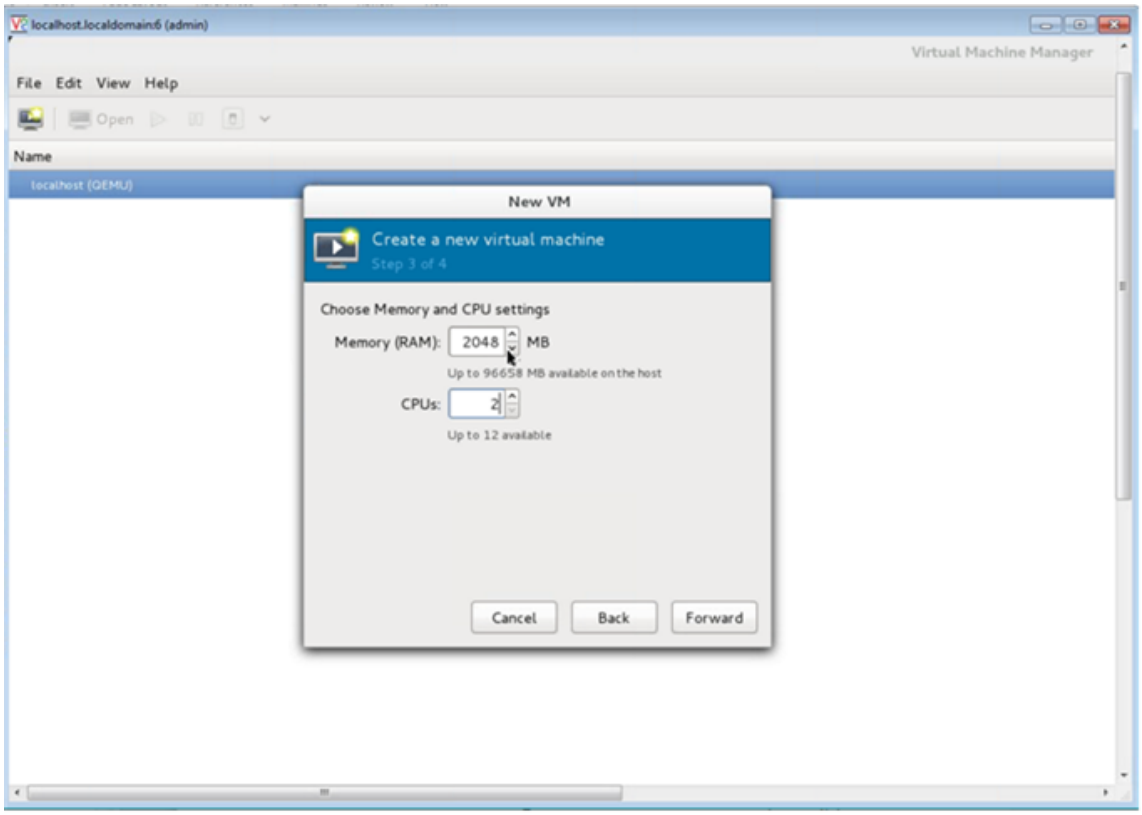


5. In the **Provide the existing storage path** field, navigate the path to the image. Choose the OS type as UNIX and Version as FreeBSD 6.x. Then, click **Forward**.

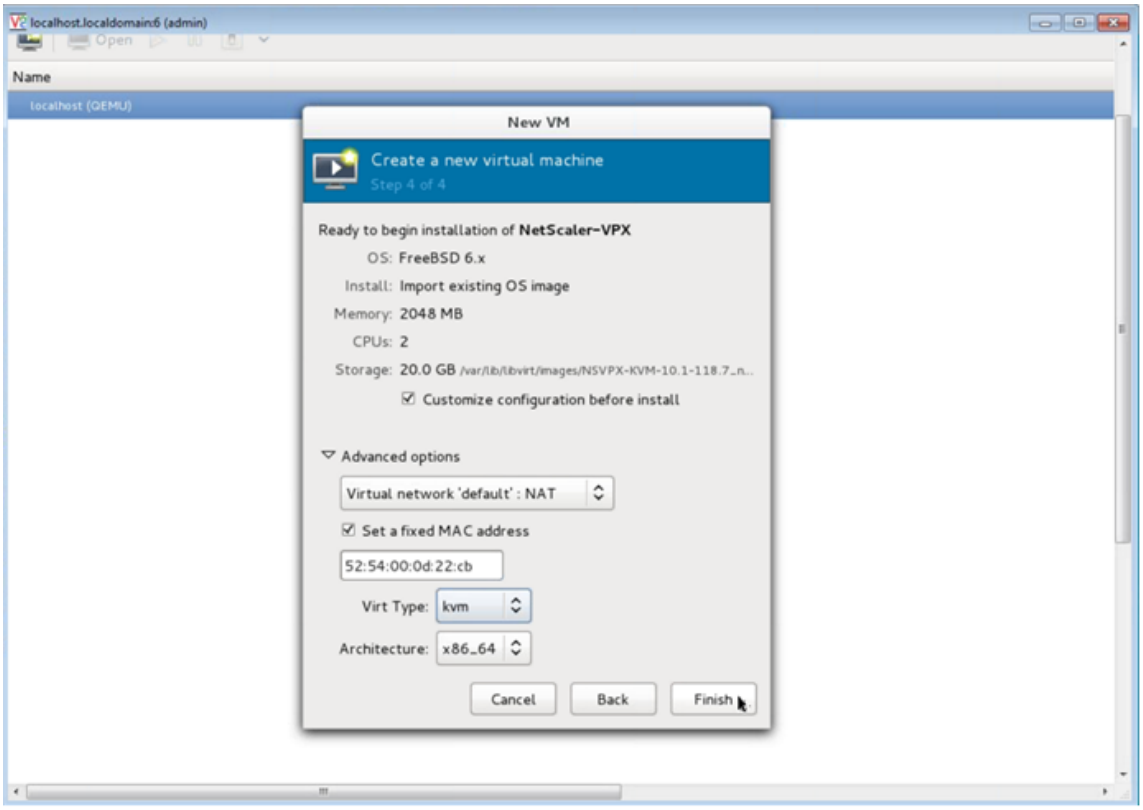


6. Under **Choose Memory and CPU** settings select the following settings, and then click **Forward**:

   - Memory (RAM)— 2048 MB
   - CPUs— 2

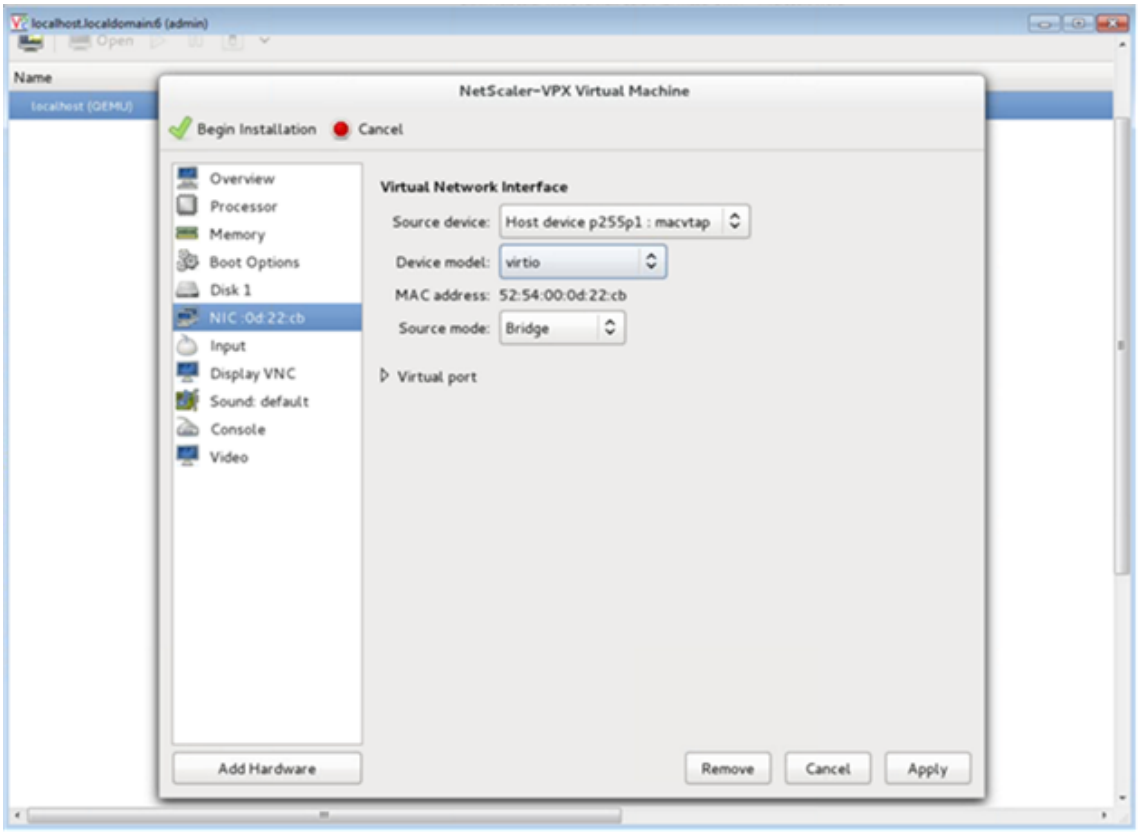


7. Select the **Customize configuration before install** check box. Optionally, under **Advanced options** you can customize the MAC address. Make sure the **Virt Type** selected is KVM and the Architecture selected is x86_64. Click **Finish**.



8. Select a NIC and provide the following configuration:

   - Source device— `ethX macvtap` or Bridge
   - Device model— `virtio`
   - Source mode— Bridge



9. Click **Apply**.
10. If you want to auto-provision the VPX instance, see the section **Enabling Auto-Provisioning by Attaching a CDROM Drive** in this document. Otherwise, click **Begin Installation**. After you have provisioned the Citrix ADC VPX on KVM, you can add more interfaces.

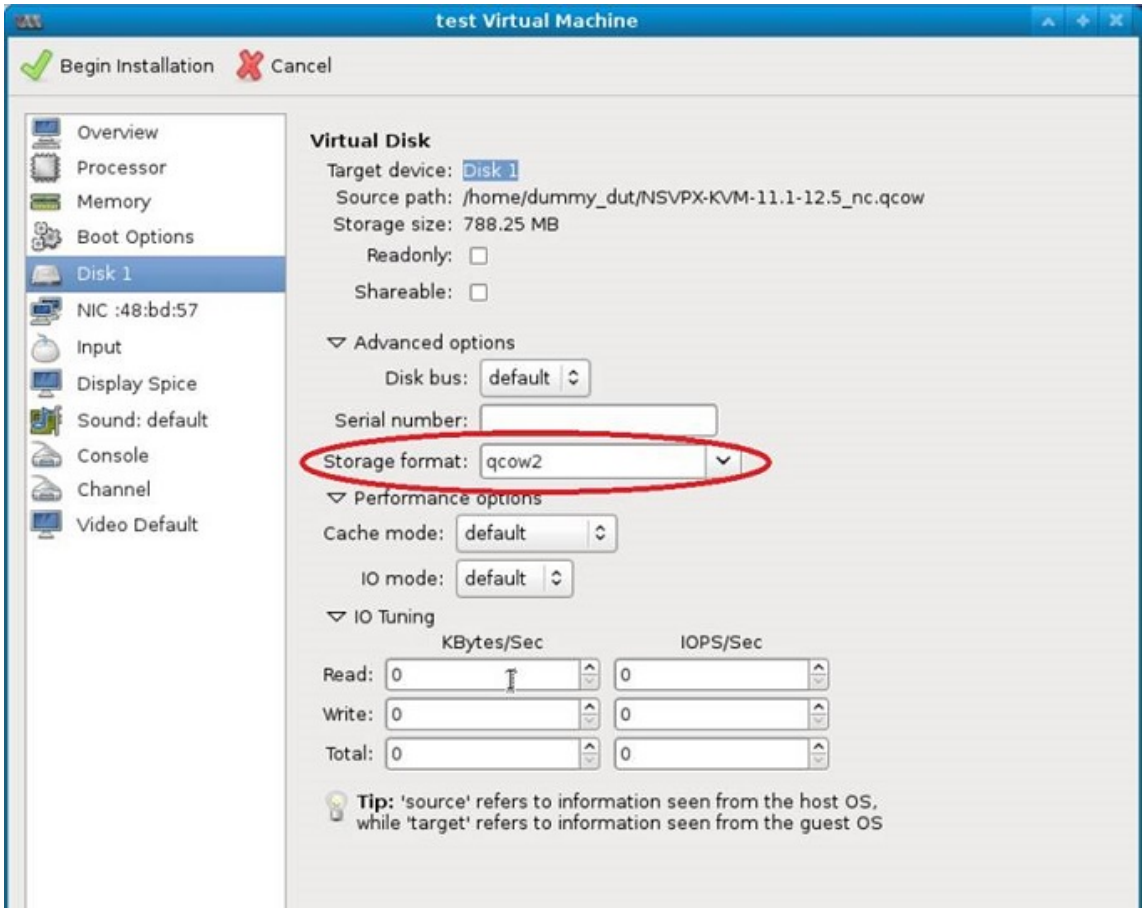
Provision the Citrix ADC VPX instance by using a QCOW2 image

Using the Virtual Machine Manager, you can provision the Citrix ADC VPX instance by using a QCOW2 image.

To provision a Citrix ADC VPX instance by using a QCOW2 image, follow these steps:

1. Follow **step 1** to **step 8** in Provision the Citrix ADC VPX instance by using a RAW image.

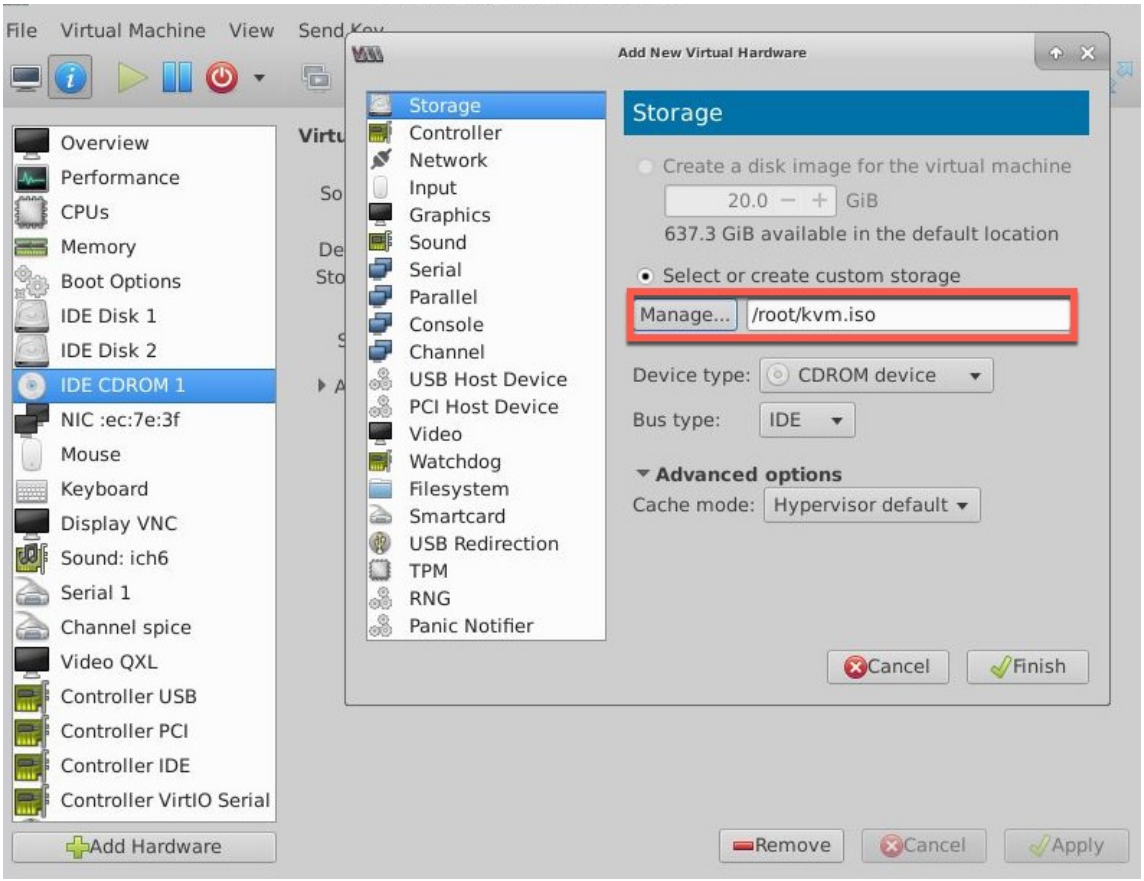
   Note: Ensure that you select **qcow2** image in **step 5**.
2. Select **Disk 1** and click **Advanced options**.
3. Select **qcow2** from the Storage format drop-down list.



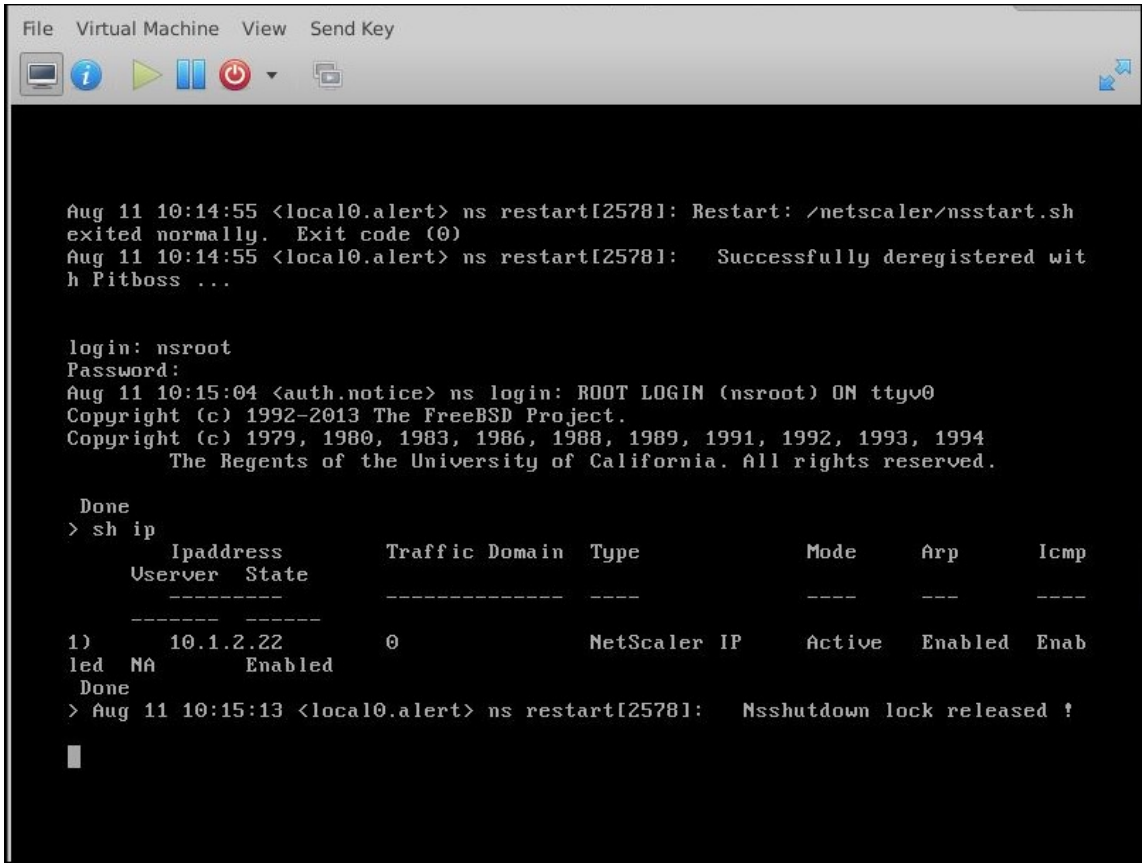
4. Click **Apply**, and then click **Begin Installation**. After you have provisioned the Citrix ADC VPX on KVM, you can add more interfaces.

Enable auto-provisioning by attaching a CDROM drive

1. Click Add **Hardware > Storage > Device type > CDROM device**.
2. Click **Manage** and select the correct ISO file that you mounted in the “Prerequisites for Auto-Provisioning a Citrix ADC VPX Instance” section, and click **Finish**. A new CDROM under Resources on your Citrix ADC VPX instance is created.



3. Power on the VPX instance, and it auto-provisions with the network configuration provided in the OVF file, as shown in the example screen capture.



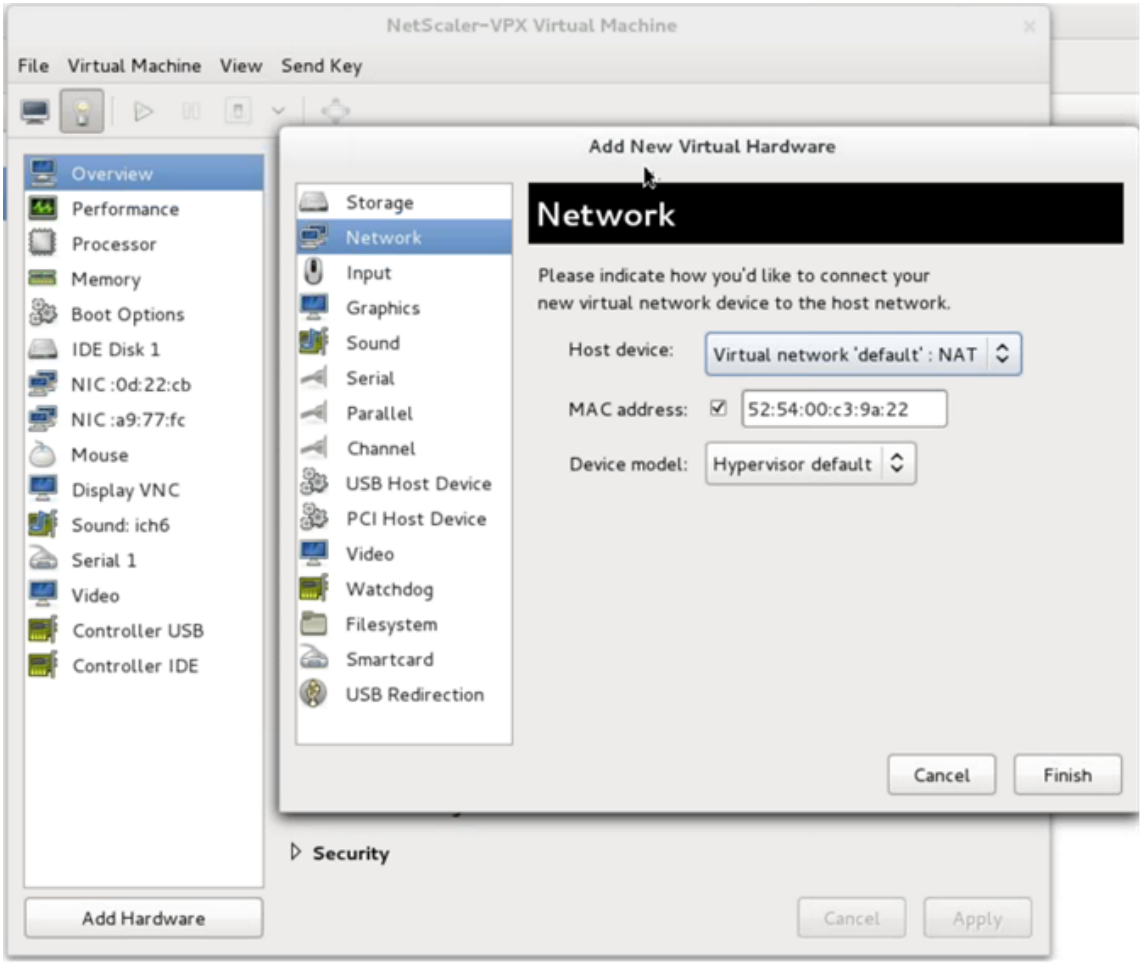
4. If auto-provision fails, the instance comes up with the default IP address (192.168.100.1). In that case, you must complete the initial configuration manually. For more information, see Configure the ADC for the first time.

Add more interfaces to the Citrix ADC VPX instance by using the Virtual Machine Manager

After you have provisioned the NetScaler VPX instance on KVM, you can add additional interfaces.

To add more interfaces, follow these steps.

1. Shut down the NetScaler VPX instance running on the KVM.
2. Right-click the VPX instance and choose **Open** from the pop-up menu.
3. Click the  icon in the header to view the virtual hardware details.
4. Click **Add Hardware**. In the **Add New Virtual Hardware window**, select **Network** from the navigation menu.



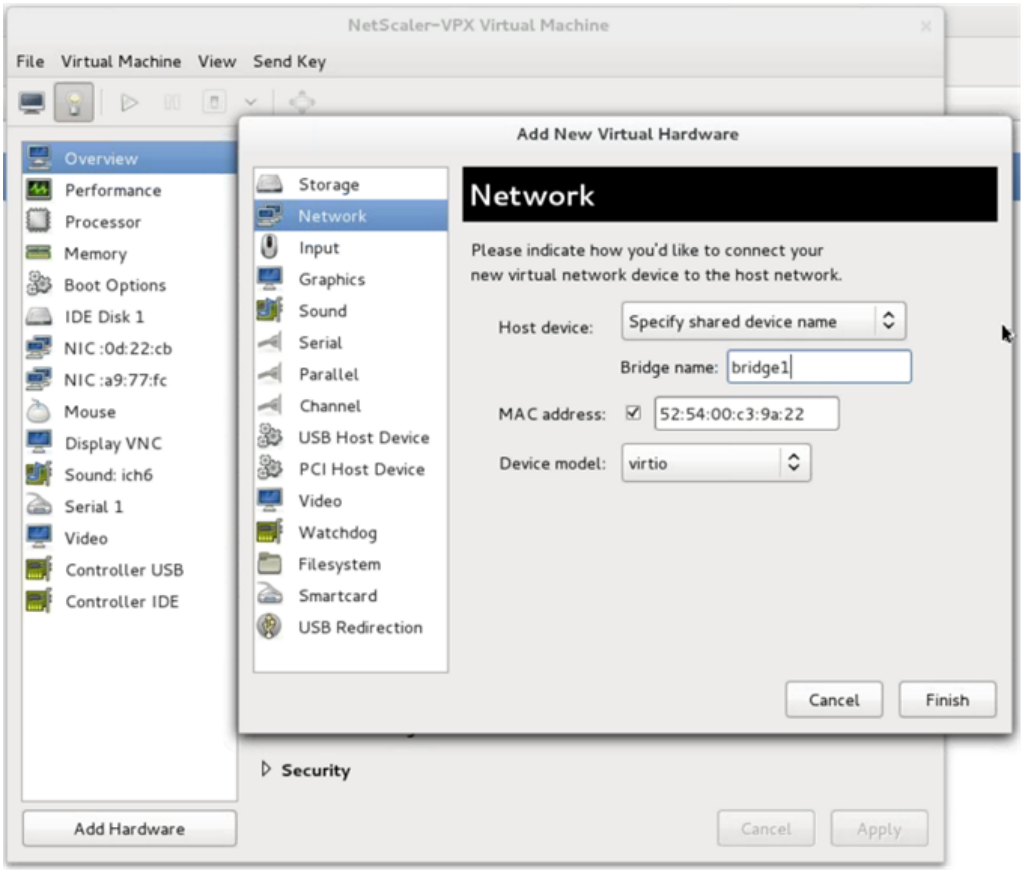
5. In **Host Device** field, select the physical interface type. The host device type can be either Bridge or MacVTap. In case of MacVTap, four modes possible are VEPA, Bridge, Private, and Pass-through.

    a) For Bridge

        i. Host device— Select the “Specify shared device name” option.

        ii. Provide the Bridge name that is configured in the KVM host.

        Note: Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.



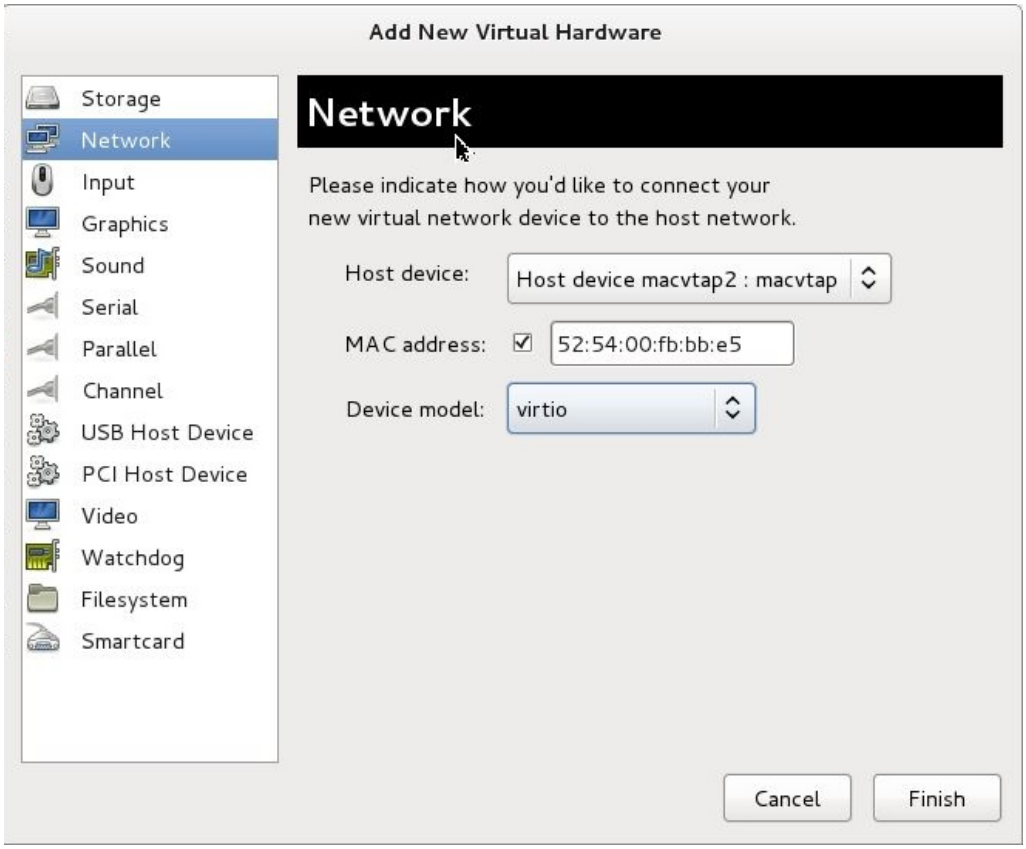
iii. Device model—`virtio`.

iv. Click Finish.

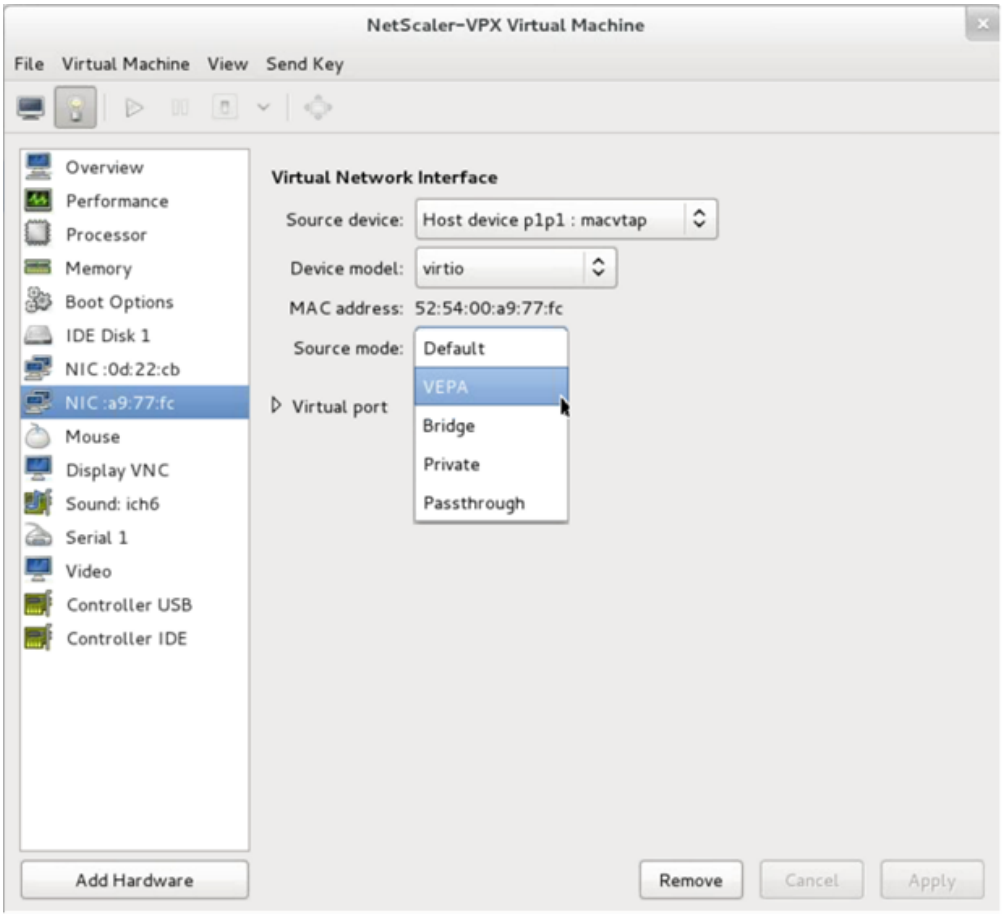
b) For MacVTap

i. Host device—Select the physical interface from the menu.

ii. Device model—`virtio`.



iii. Click Finish. You can view the newly added NIC in the navigation pane.



iv. Select the newly added NIC and select the Source mode for this NIC. The available modes are VEPA, Bridge, Private, and Passthrough. For more details on the interface and modes, see Source Interface and Modes.

v. Click Apply.

6. If you want to auto-provision the VPX instance, see the section “Adding a Config Drive to Enable Auto-Provisioning” in this document. Otherwise, power on the VPX instance to complete the initial configuration manually.

> **Important**
>
> Interface parameter configurations such as speed, duplex, and autonegotiation are not supported.

Configure a Citrix ADC VPX instance to use SR-IOV network interfaces

September 14, 2021

You can configure a Citrix ADC VPX instance running on Linux-KVM platform using single root I/O virtualization (SR-IOV) with the following NICs:

- Intel 82599 10G
- Intel X710 10G
- Intel XL710 40G
- Intel X722 10G

This section describes how to:

- Configure a Citrix ADC VPX Instance to Use SR-IOV Network Interface
- Configure Static LA/LACP on the SR-IOV Interface
- Configure VLAN on the SR-IOV Interface

Limitations

Keep the limitations in mind while using Intel 82599, X710, XL710, and X722 NICs. The following features not supported.

Limitations for Intel 82599 NIC:

- L2 mode switching.
- Admin partitioning (shared VLAN mode).
- High availability (active-active mode).
- Jumbo frames.
- IPv6: You can configure only up to 30 unique IPv6 addresses in a VPX instance if you've at least one SR-IOV interface.
- VLAN configuration on Hypervisor for SRIOV VF interface through `ip link` command is not supported.
- Interface parameter configurations such as speed, duplex, and autonegotiations are not supported.

Limitations for Intel X710 10G, Intel XL710 40G, and Intel X722 10G NICs:

- L2 mode switching.
- Admin partitioning (shared VLAN mode).
- In a cluster, Jumbo frames are not supported when the XL710 NIC is used as a data interface.
- Interface list reorders when interfaces are disconnected and reconnected.
- Interface parameter configurations such as speed, duplex, and auto negotiations are not supported.
- Interface name is 40/X for Intel X710 10G, Intel XL710 40G, and Intel X722 10G NICs
- Up to 16 Intel XL710/X710/X722 SRIOV or PCI passthrough interfaces can be supported on a VPX instance.

> **Note:** For Intel X710 10G, Intel XL710 40G, and Intel X722 10G NICs to support IPv6, you need to enable trust mode on the Virtual Functions (VFs) by typing the following command on the KVM host:
>
> ```
> ## ip link set <PNIC> <VF> trust on
> ```
>
> Example:
>
> ```
> ## ip link set ens785f1 vf 0 trust on
> ```

Prerequisites

Before you configure a Citrix ADC VPX instance to use SR-IOV network interfaces, complete the following prerequisite tasks. See the NIC column for details about how to complete the corresponding tasks.

Task	Intel 82599 NIC	Intel X710, XL710, and X722 NICs
1. Add the NIC to the KVM host.	-	-
2. Download and install the latest Intel driver.	IXGBE driver	I40E driver
3. Block list the driver on the KVM host.	Add the following entry in the /etc/modprobe.d/blacklist.conf file: `blacklist ixgbevf`. Use IXGBE driver version 4.3.15 (recommended).	Add the following entry in the /etc/modprobe.d/blacklist.conf file: `blacklist i40evf`. Use i40e driver version 2.0.26 (recommended).

Task	Intel 82599 NIC	Intel X710, XL710, and X722 NICs
4.Enable SR-IOV Virtual Functions (VFs) on the KVM host. In both the commands in the next two columns: `number_of_VFs` = the number of Virtual VFs that you want to create. `device_name` = the interface name.	If you are using earlier version of kernel 3.8, then add the following entry to the /etc/modprobe.d/ixgbe file and restart the KVM host: `options ixgbe max_vfs =<number_of_VFs>`. If you are using kernel 3.8 version or later, create VFs using the following command: `echo <number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`. See example in figure 1.	If you are using earlier version of kernel 3.8, then add the following entry to the /etc/modprobe.d/i40e.conf file and restart the KVM host: `options i40e max_vfs=<number_of_VFs>`. If you are using kernel 3.8 version or later, create VFs using the following command: `echo<number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`. See example in figure 2.
5. Make the VFs persistent by adding the commands that you used to create VFs, to the rc.local file.	See example in figure 3.	See example in figure 3.

> **Important**
>
> When you create the SR-IOV VFs, ensure that you do not assign MAC addresses to the VFs.

Figure 1: Enable SR-IOV VFs on the KVM host for Intel 82599 10G NIC.

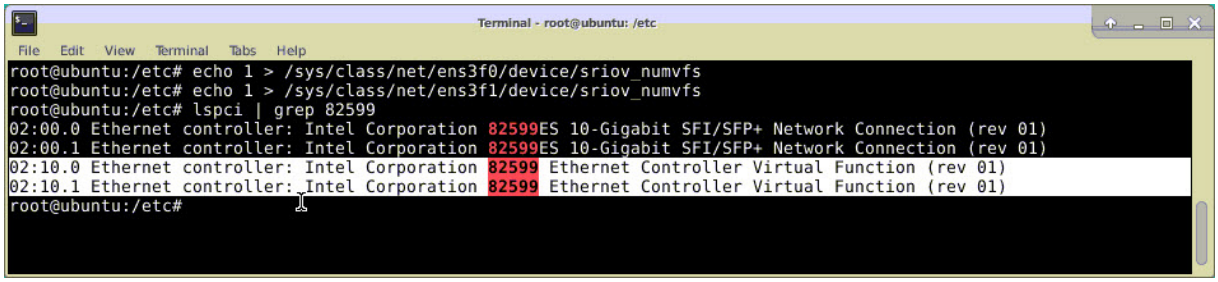


Figure 2: Enable SR-IOV VFs on the KVM host for Intel X710 10G and XL710 40G NICs.


```
root@ubuntu:~# lspci | grep 710
03:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:00.2 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:00.3 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:06.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:06.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.2 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.3 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.2 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.3 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
81:00.0 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 01)
82:00.0 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 02)
82:00.1 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 02)
82:02.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
82:02.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
82:0a.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
82:0a.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
root@ubuntu:~#
```

Figure 3: Enable SR-IOV VFs on the KVM host for Intel X722 10G NIC.

```
root@ubuntu:~# lspci | grep  "37cd"
84:02.0 Ethernet controller: Intel Corporation Device 37cd (rev 04)
84:0a.0 Ethernet controller: Intel Corporation Device 37cd (rev 04)
```

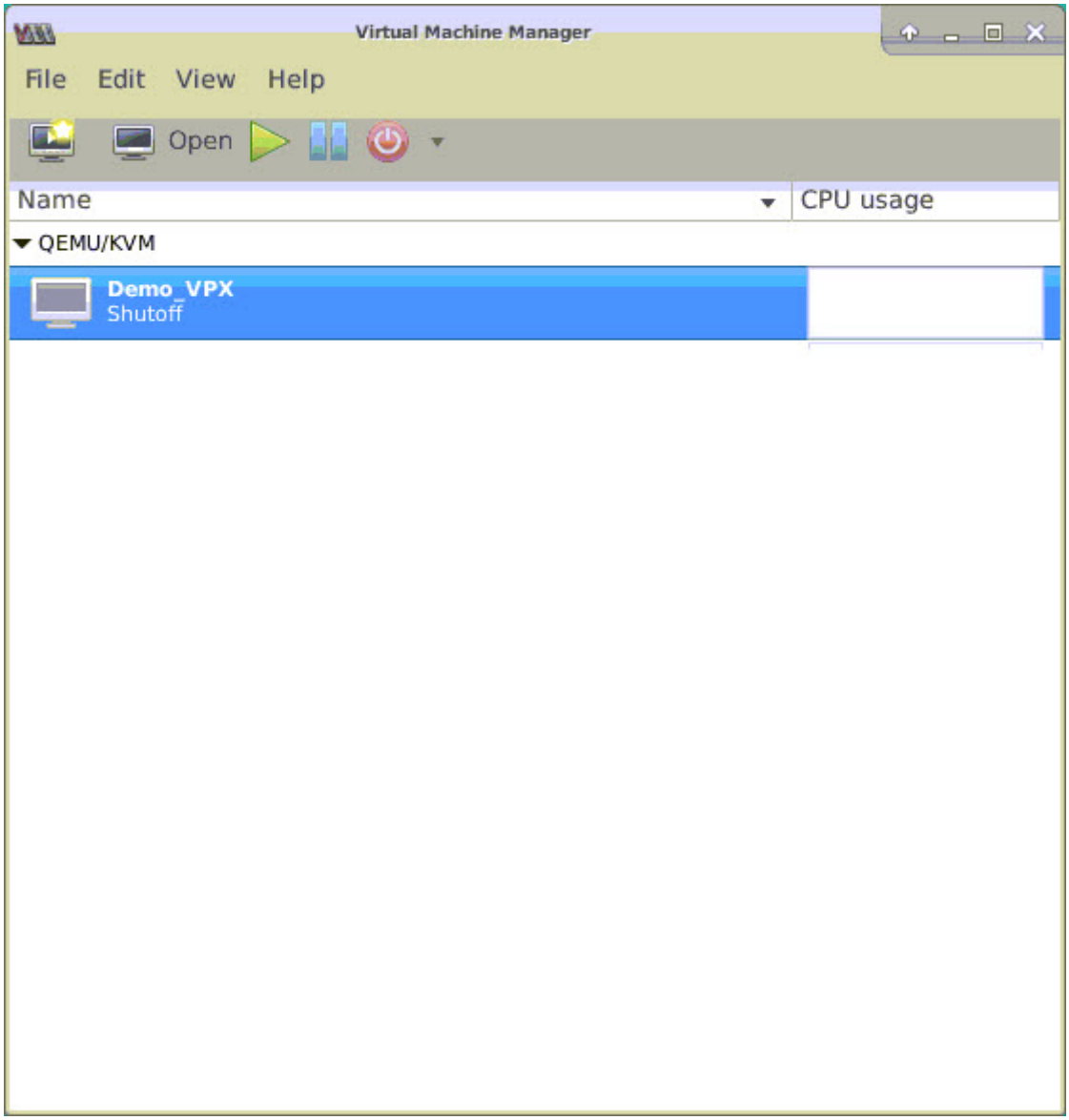
Figure 4: Make the VFs persistent.

```
Terminal - root@ubuntu: /etc
File  Edit  View  Terminal  Tabs  Help
root@ubuntu:/etc# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
echo 1 > /sys/class/net/ens3f0/device/sriov_numvfs
echo 1 > /sys/class/net/ens3f1/device/sriov_numvfs
exit 0
root@ubuntu:/etc#
```

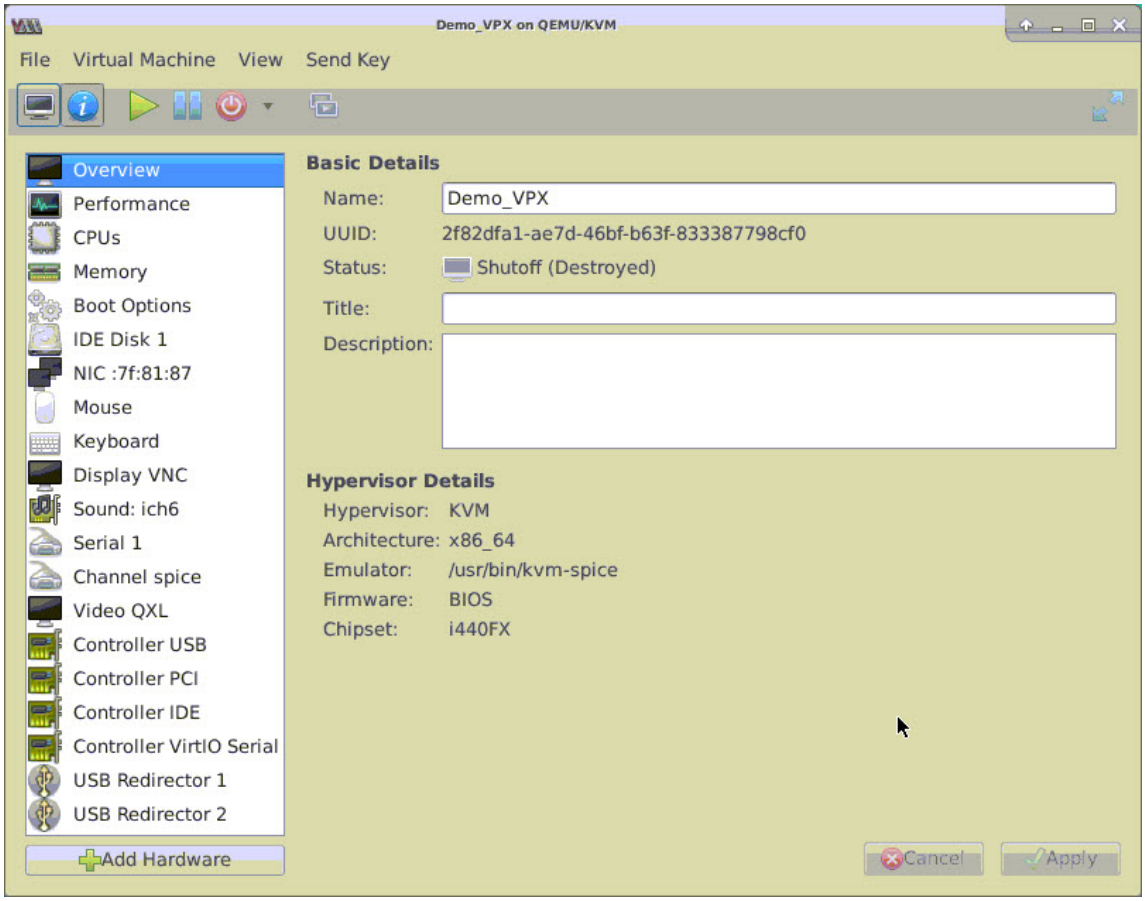
Configure a Citrix ADC VPX instance to use SR-IOV network interface

To configure the Citrix ADC VPX instance to use SR-IOV network interface by using Virtual Machine Manager, complete these steps:

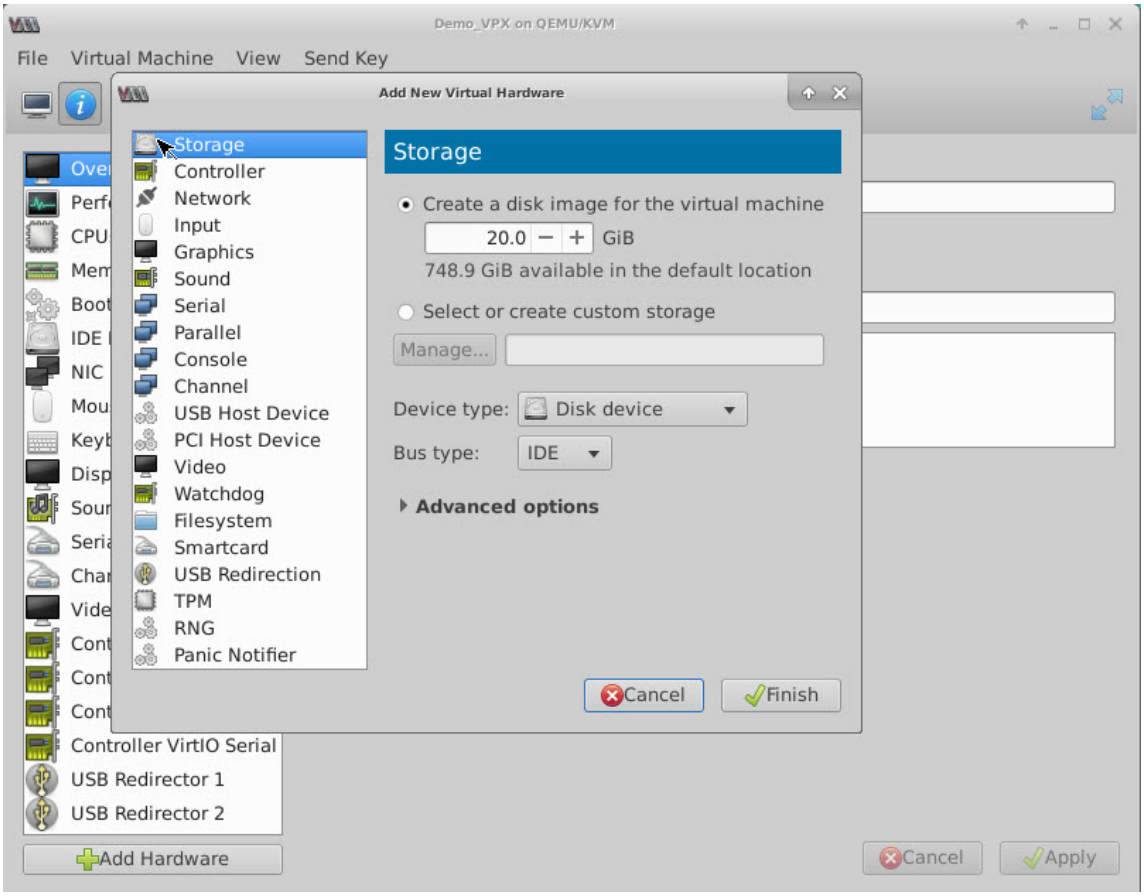
1. Power off the Citrix ADC VPX instance.
2. Select the Citrix ADC VPX instance and then select Open.



3. In the <virtual machine on KVM> window, select the **i** icon.



4. Select **Add Hardware**.



5. In the **Add New Virtual Hardware** dialog box, do the following:

   a) Select PCI Host Device.

   b) In the Host Device section, select the VF you have created and click Finish.

Figure 4: VF for Intel 82599 10G NIC

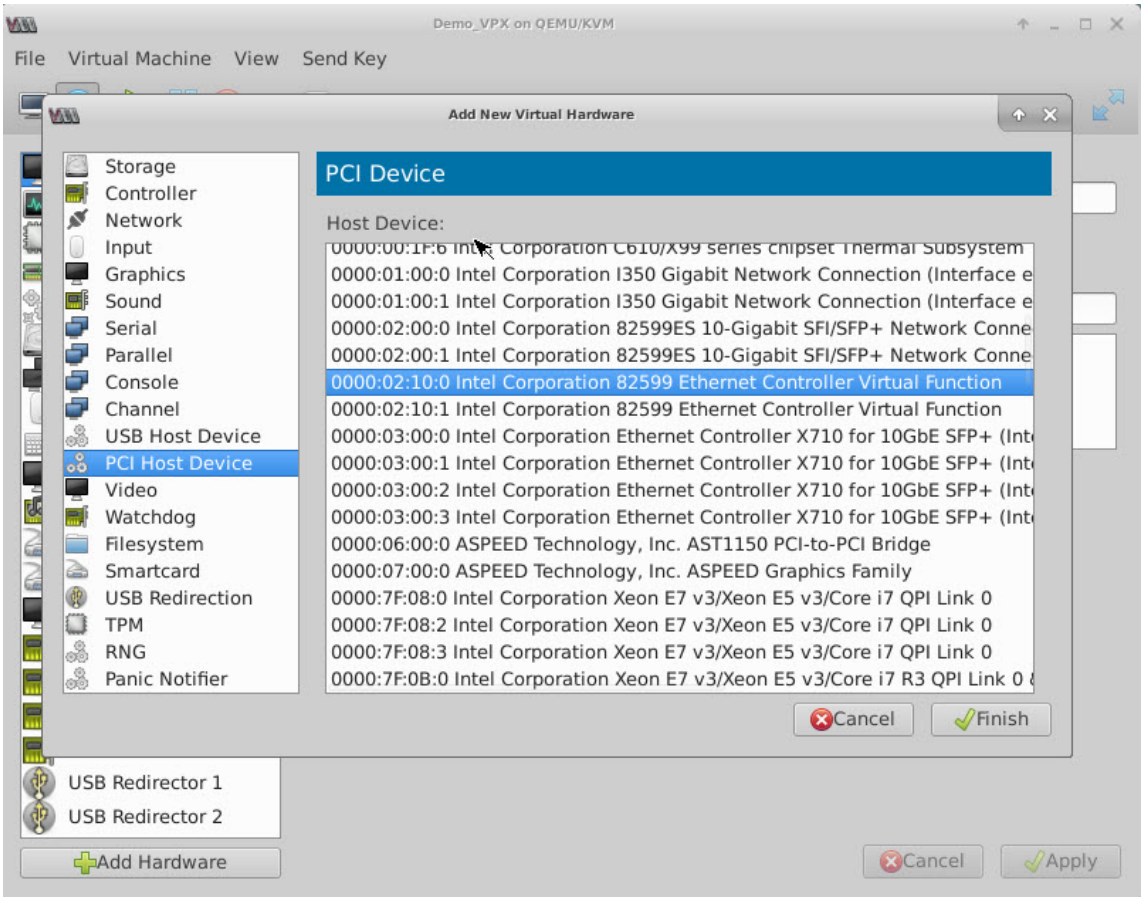


Figure 5: VF for Intel XL710 40G NIC

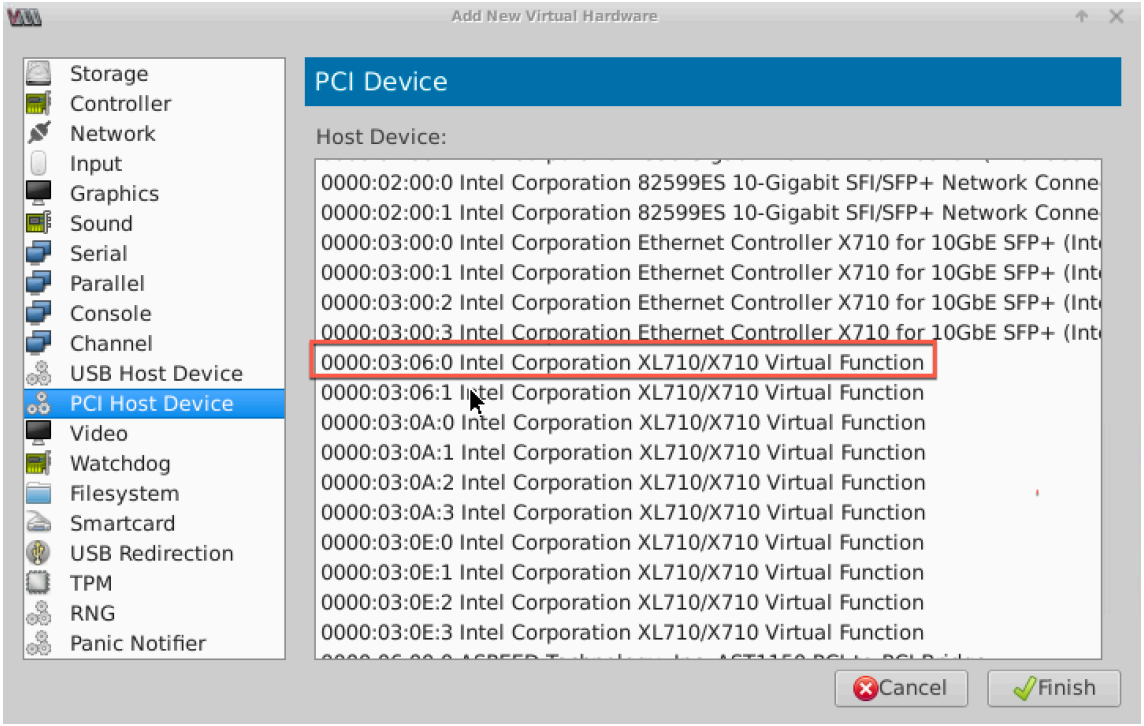
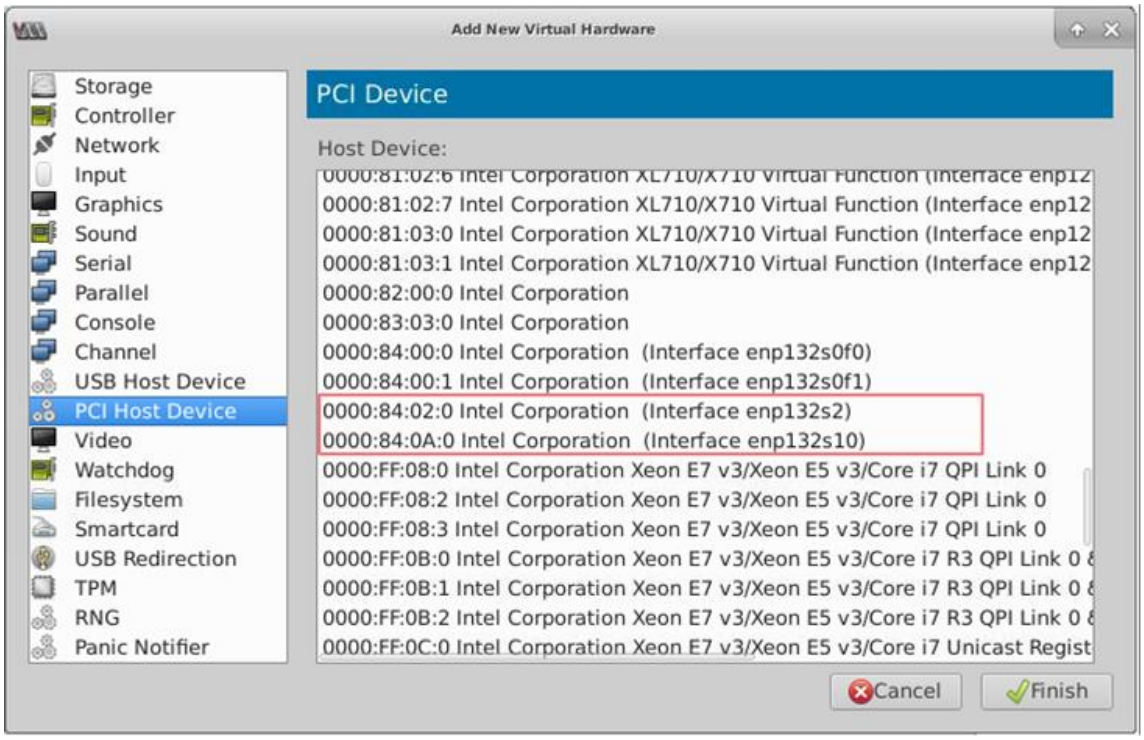


Figure 6: VF for Intel X722 10G NIC



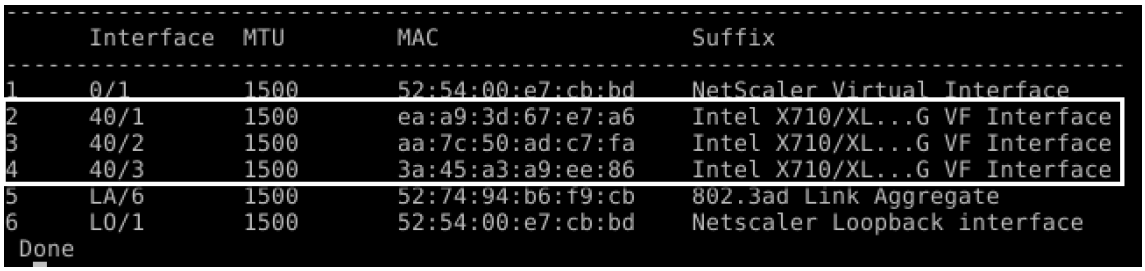
6. Repeat Step 4 and 5 to add the VFs that you have created.
7. Power on the Citrix ADC VPX instance.
8. After the Citrix ADC VPX instance powers on, use the following command to verify the configuration:

```
1 show interface summary
2 <!--NeedCopy-->
```

The output shows all the interfaces that you configured.

Figure 6: output summary for Intel 82599 NIC.

Figure 7. Output summary for Intel X710 and XL710 NICs.



Configure static LA/LACP on the SR-IOV interface

> **Important**
>
> When you are creating the SR-IOV VFs, ensure that you do not assign MAC addresses to the VFs.

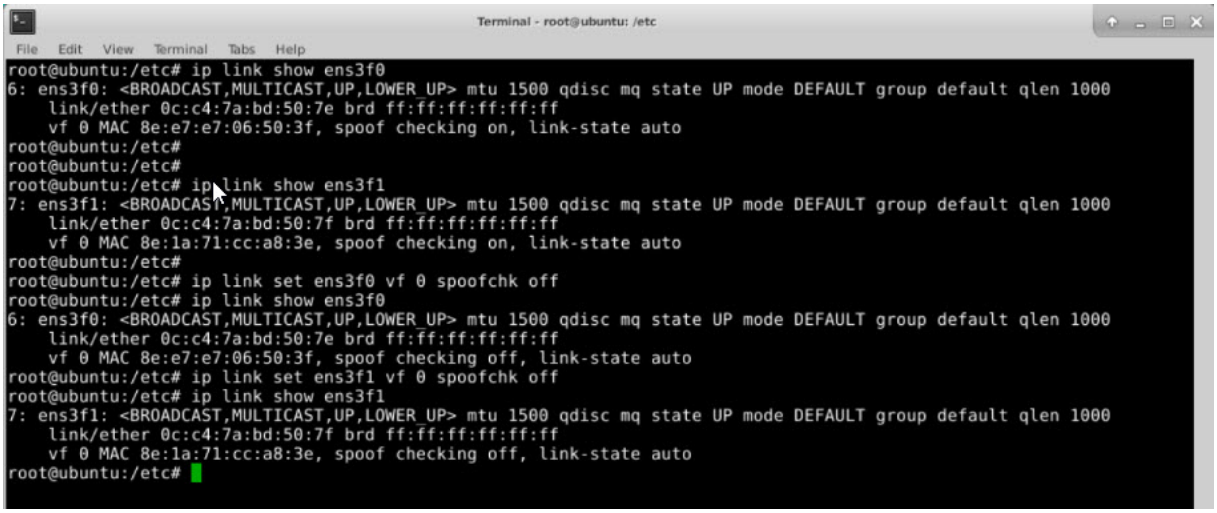
To use the SR-IOV VFs in link aggregation mode, disable spoof checking for VFs that you have created. On the KVM host, use the following command to disable spoof checking:

```
*ip link set \<interface\\_name\> vf \<VF\\_id\> spoofchk off*
```

Where:

- Interface_name – is the interface name.
- VF_id – is the Virtual Function id.

Example:



After you disable spoof checking for all the VFs that you have created. Restart the Citrix ADC VPX instance and configure link aggregation. For detailed instructions, see Configuring Link Aggregation.

Configuring VLAN on the SR-IOV Interface

You can configure VLAN on SR-IOV VFs. For detailed instructions, see Configuring a VLAN.

> **Important**
>
> Ensure that the KVM host does not contain VLAN settings for the VF interface.

Configure a Citrix ADC VPX instance to use PCI passthrough network interfaces

September 14, 2021

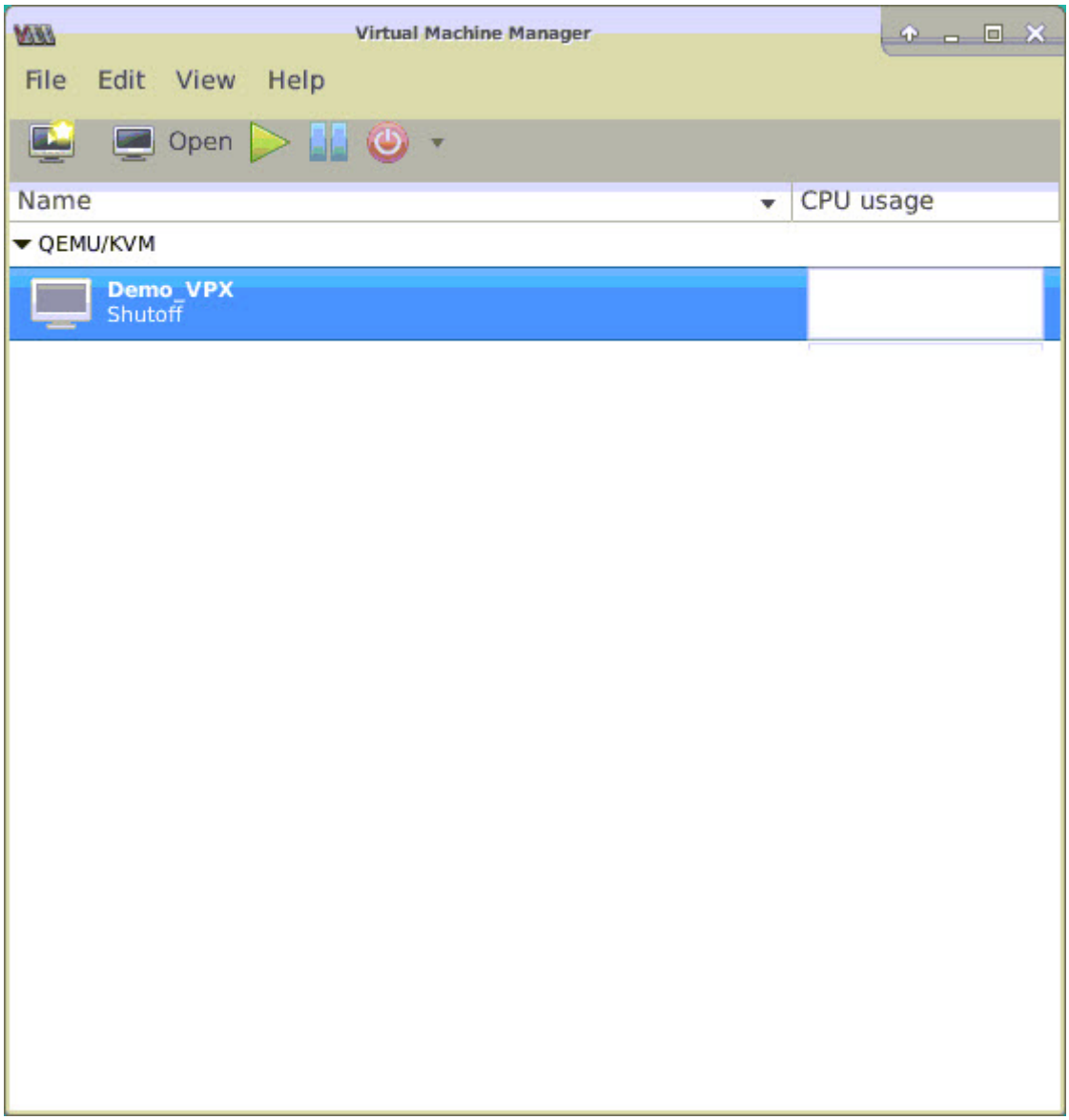
After you have installed and configured a Citrix ADC VPX instance on the Linux-KVM platform, you can use the Virtual Machine Manager to configure the virtual appliance to use PCI passthrough network interfaces.

Prerequisites

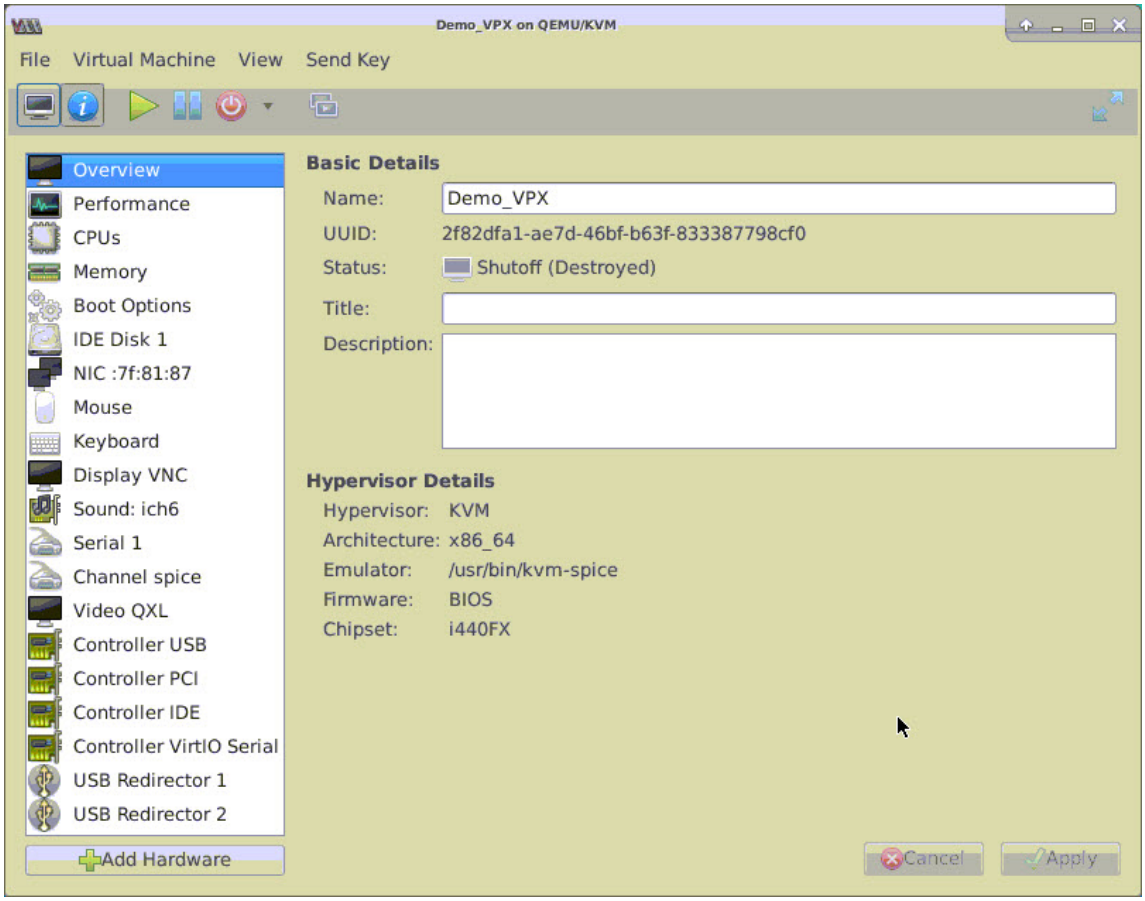
- The firmware version of the Intel XL710 NIC (NIC) on the KVM Host is 5.04.
- The KVM Host supports input–output memory management unit (IOMMU) and Intel VT-d, and they are enabled in the BIOS of the KVM Host. On the KVM Host, to enable IOMMU, add the following entry to the **/boot/grub2/grub.cfg** file: **intel_iommu=1**
- Run the following command and reboot the KVM Host: **Grub2-mkconfig –o /boot/-grub2/grub.cfg**

To configure Citrix ADC VPX instances to use PCI passthrough network interfaces by using the Virtual Machine Manager:

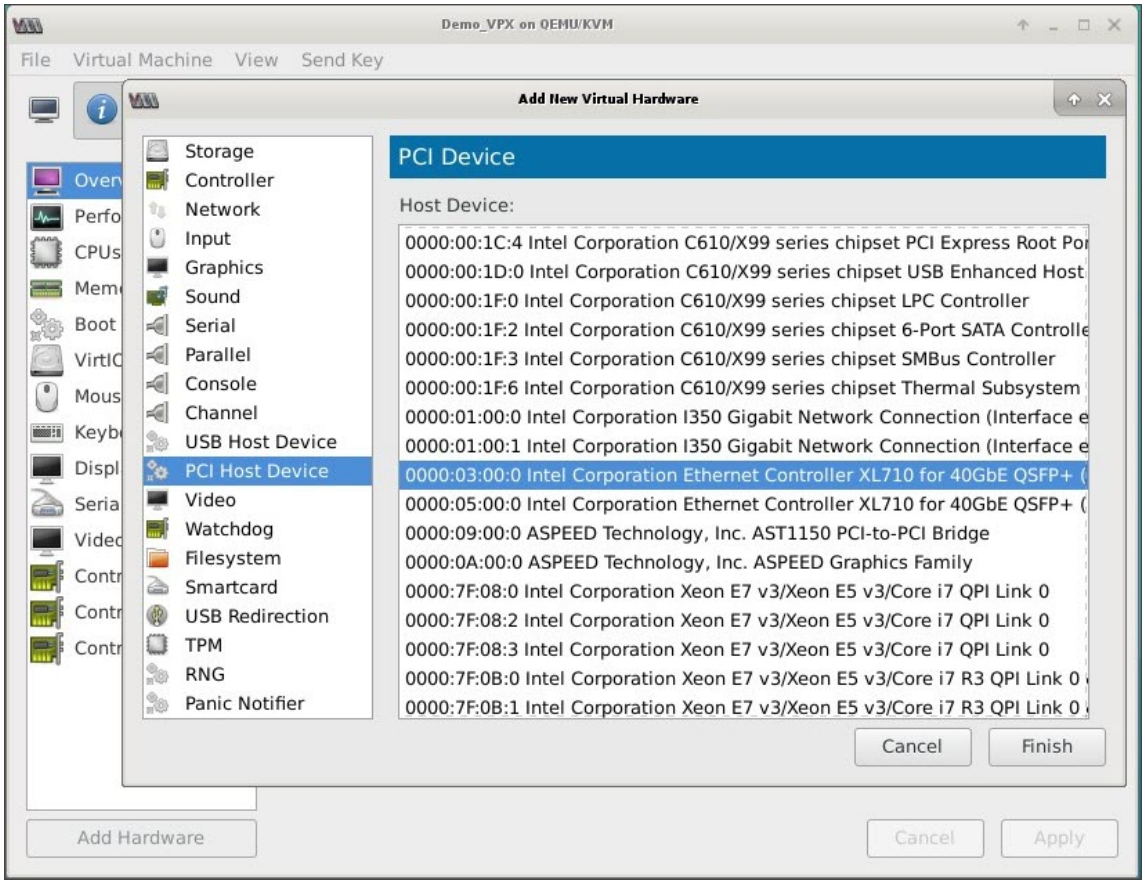
1. Power off the Citrix ADC VPX instance.
2. Select the Citrix ADC VPX instance and click **Open**.



3. In the **virtual_machine on KVM>** window, click the **i** icon.



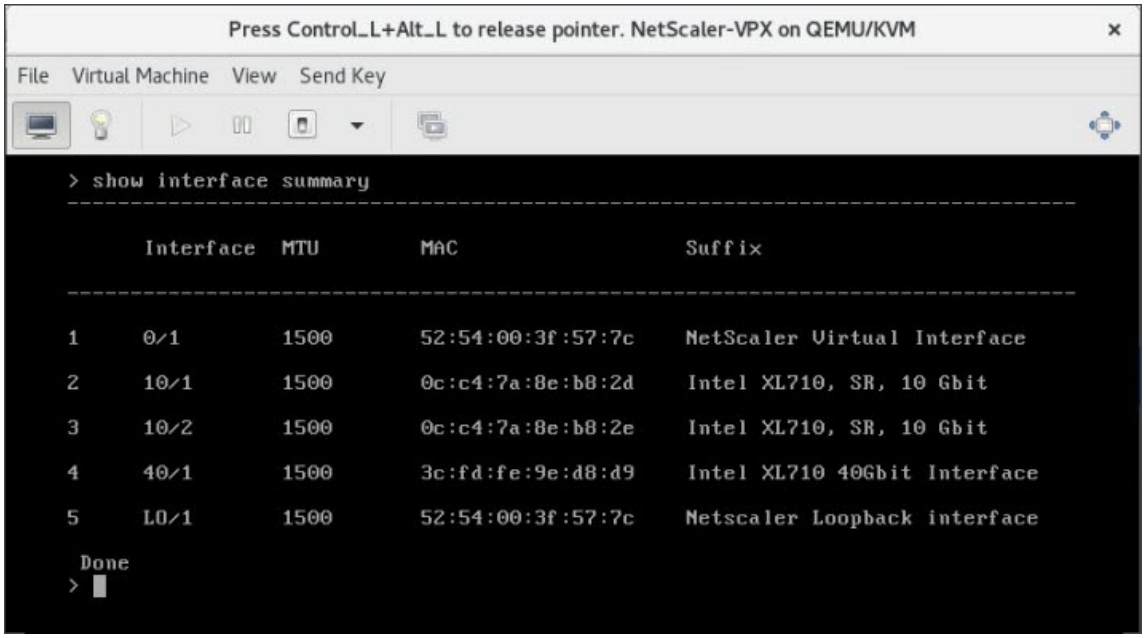
4. Click **Add Hardware**.
5. In the **Add New Virtual Hardware** dialog box, do the following:
   a. Select **PCI Host Device**.
   b. In the **Host Device** section, select the Intel XL710 physical function.
   c. Click **Finish**.



6. Repeat steps **4** and **5** to add any additional Intel XL710 physical functions.
7. Power on the Citrix ADC VPX instance.
8. Once the Citrix ADC VPX instance powers on, you can use the following command to verify the configuration:

```
COMMAND
> show interface summary
```

The output must show all the interfaces that you configured:



Provision the Citrix ADC VPX instance by using the virsh program

September 14, 2021

The `virsh` program is a command line tool for managing VM Guests. Its functionality is similar to that of Virtual Machine Manager. It enables you to change a VM Guest's status (start, stop, pause, and so on), to set up new Guests and devices, and to edit existing configurations. The `virsh` program is also useful for scripting VM Guest management operations.

To provision Citrix ADC VPX by using the `virsh` program, follow these steps:

1. Use the tar command to untar the Citrix ADC VPX package. The NSVPX-KVM-*_nc.tgz package contains the following components:

   - The Domain XML file specifying VPX attributes [NSVPX-KVM-*_nc.xml]
   - Check sum of NS-VM Disk Image [Checksum.txt]
   - NS-VM Disk Image [NSVPX-KVM-*_nc.raw]

   Example:

```
tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
NSVPX-KVM-10.1-117_nc.xml
NSVPX-KVM-10.1-117_nc.raw
checksum.txt
<!--NeedCopy-->
```


2. Copy the NSVPX-KVM-*_nc.xml XML file to a file named <DomainName>-NSVPX-KVM-*_nc.xml. The <DomainName> is also the name of the virtual machine. Example:

```
cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.
   xml
<!--NeedCopy-->
```

3. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file to specify the following parameters:

   - name— Specify the name.
   - Mac— Specify the MAC address.
     Note: The domain name and the MAC address have to be unique.
   - source file— Specify the absolute disk-image source path. The file path has to be absolute. You can specify the path of the RAW image file or a QCOW2 image file.

     If you want to specify a RAW image file, specify the disk image source path as shown in the following example:

     Example:

```
<name>NetScaler-VPX</name>
<mac address='52:54:00:29:74:b3'/>
<source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>
<!--NeedCopy-->
```

     Specify the absolute QCOW2 disk-image source path and define the driver type as **qcow2**, as shown in the following example:

     Example:

```
<name>NetScaler-VPX</name>
<mac address='52:54:00:29:74:b3'/>
<driver name ='qemu' type='qcow2'/>
<source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>*
<!--NeedCopy-->
```

4. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file to configure the networking details:

   - source dev— specify the interface.
   - mode— specify the mode. The default interface is **Macvtap Bridge**.

   Example: Mode: MacVTap Bridge Set target interface as `ethx` and mode as bridge Model type as `virtio`

```
<interface type='direct'>
    <mac address='52:54:00:29:74:b3'/>
```



```
3        <source dev='eth0' mode='bridge'/>
4        <target dev='macvtap0'/>
5        <model type='virtio'/>
6        <alias name='net0'/>
7        <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
             function='0x0'/>
8      </interface>
9  <!--NeedCopy-->
```

Here, eth0 is the physical interface attached to the VM.

5. Define the VM attributes in the <DomainName>-NSVPX-KVM-*_nc.xml file by using the following command: `virsh` define <DomainName>-NSVPX-KVM-*_nc.xml Example:

```
1 virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
2 <!--NeedCopy-->
```

Start the VM by entering the following command: `virsh` start [<DomainName> <DomainUUID>] Example:

```
1 virsh start NetScaler-VPX
2 <!--NeedCopy-->
```

Connect the Guest VM through the console `virsh` console [<DomainName> <DomainUUID> <DomainID>] Example:

```
1 virsh console NetScaler-VPX
2 <!--NeedCopy-->
```

Add more interfaces to Citrix ADC VPX instance using `virsh` program

After you have provisioned the Citrix ADC VPX on KVM, you can add additional interfaces.

To add more interfaces, follow these steps:

1. Shut down the Citrix ADC VPX instance running on the KVM.

Edit the <DomainName>-NSVPX-KVM-*_nc.xml <DomainUUID>]
file using the command: `virsh` edit
[<DomainName>

2.

3. In the <DomainName>-NSVPX-KVM-*_nc.xml file, append the following parameters:

   a) **For MacVTap**

   - Interface type— Specify the interface type as 'direct'.
   - MAC address— Specify the MAC address and make sure the MAC address is unique across the interfaces.
   - source dev— Specify the interface name.
   - mode— Specify the mode. The modes supported are - Bridge, VEPA, Private, and Pass-through
   - model type— Specify the model type as `virtio`

   Example:

   Mode: MacVTap Pass-through

   Set target interface as
   `ethx`, Mode as
   bridge, and model type as
   `virtio`

```
<interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth1' mode='passthrough'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

   Here eth1 is the physical interface attached to the VM.

   b) **For Bridge Mode**

   Note: Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.

   - Interface type— Specify the interface type as 'bridge'.
   - MAC address— Specify the MAC address and make sure the MAC address is unique across the interfaces.
   - source bridge— Specify the bridge name.

- model type— Specify the model type as `virtio`

Example: Bridge Mode

```
<interface type='bridge'>
      <mac address='52:54:00:2d:43:a4'/>
      <source bridge='br0'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

Manage the Citrix ADC VPX guest VMs

September 14, 2021

You can use the Virtual Machine Manager and the `virsh` program to perform management tasks such as starting or stopping a VM Guest, setting up new guests and devices, editing existing configurations, and connecting to the graphical console through Virtual Network Computing (VNC).

Manage the VPX guest VMs by using Virtual Machine Manager

- List the VM guests

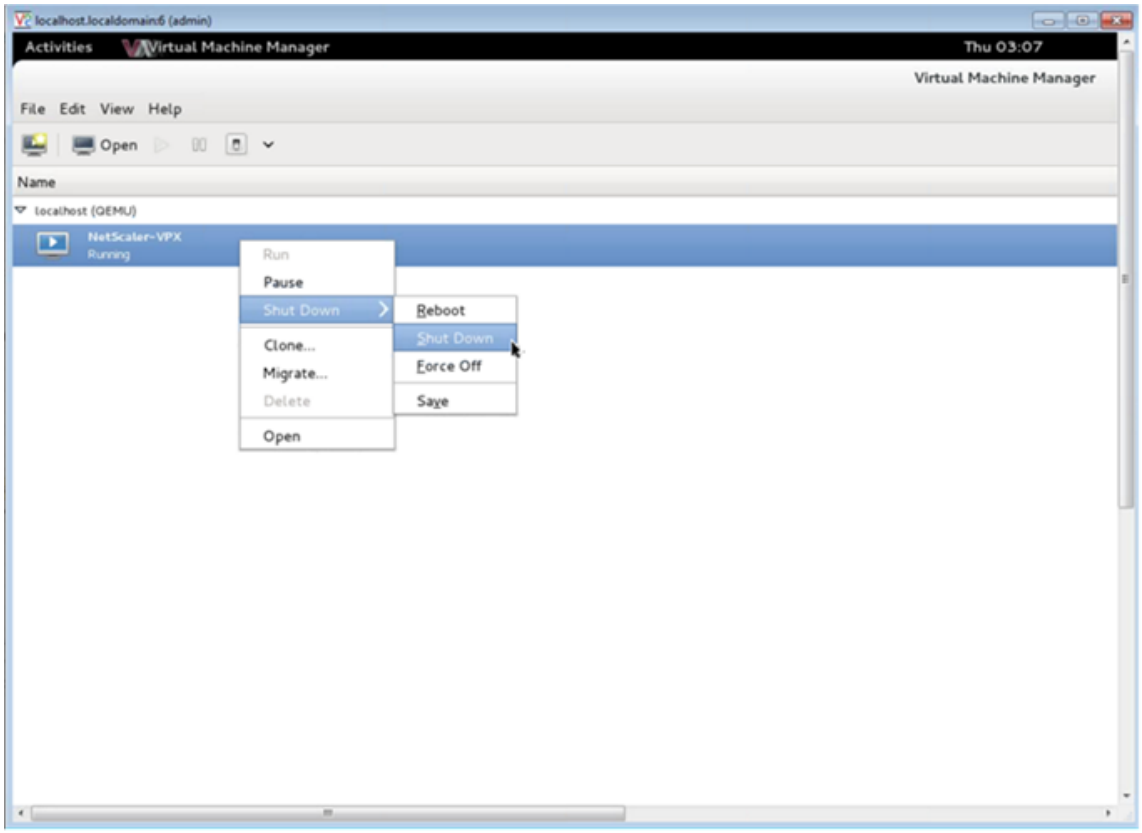
  The main Window of the Virtual Machine Manager displays a list of all the VM Guests for each VM host server it is connected to. Each VM Guest entry contains the virtual machine's name, along with its status (Running, Paused, or Shutoff) displayed as in the icon.

- Open a graphical console

  Opening a Graphical Console to a VM Guest enables you to interact with the machine like you would with a physical host through a VNC connection. To open the graphical console in the Virtual Machine Manager, right-click the VM Guest entry and select the Open option from the pop-up menu.

- Start and shut down a guest

  You can start or stop a VM Guest from the Virtual Machine Manager. To change the state of the VM, right-click the VM Guest entry and select Run or one of the Shut Down options from the pop-up menu.



- Reboot a guest

  You can reboot a VM Guest from the Virtual Machine Manager. To reboot the VM, right-click the VM Guest entry, and then select Shut Down > Reboot from the pop-up menu.

- Delete a guest

  Deleting a VM Guest removes its XML configuration by default. You can also delete a guest's storage files. Doing so completely erases the guest.

  1. In the Virtual Machine Manager, right-click the VM Guest entry.
  2. Select Delete from the pop-up menu. A confirmation window opens.
     Note: The Delete option is enabled only when the VM Guest is shut down.
  3. Click Delete.
  4. To completely erase the guest, delete the associated .raw file by selecting the Delete Associated Storage Files check box.

Manage the Citrix ADC VPX guest VMs using the `virsh` program

- List the VM Guests and their current states.

  To use `virsh` to display information about the Guests

  ```
  virsh list --all
  ```


The command output displays all domains with their states. Example output:

```
Id Name                 State
----------------------------------
  0 Domain-0           running
  1 Domain-1           paused
  2 Domain-2           inactive
  3 Domain-3           crashed
<!--NeedCopy-->
```

- Open a `virsh` console.

  Connect the Guest VM through the console

  `virsh console [<DomainID> | <DomainName> | <DomainUUID>]`

  Example:

  `virsh console NetScaler-VPX`

- Start and shut down a guest.

  Guests can be started using the DomainName or Domain-UUID.

  `virsh start [<DomainName> | <DomainUUID>]`

  Example:

  `virsh start NetScaler-VPX`

  To shut down a guest:

  `virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]`

  Example:

  `virsh shutdown NetScaler-VPX`

- Reboot a guest

  `virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]`

  Example:

  `virsh reboot NetScaler-VPX`

  Delete a guest

  To delete a Guest VM you must shut down the Guest and undefine the <DomainName>-NSVPX-KVM-*_nc.xml before you run the delete command.

```
virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
virsh undefine [<DomainName> | <DomainUUID>]
<!--NeedCopy-->
```


Example:

```
virsh shutdown NetScaler-VPX
virsh undefine NetScaler-VPX
<!--NeedCopy-->
```

> Note: The delete command doesn't remove disk image file which must be removed manually.

Provision the Citrix ADC VPX instance with SR-IOV, on OpenStack

September 14, 2021

You can deploy high-performance Citrix ADC VPX instances that use single-root I/O virtualization (SR-IOV) technology, on OpenStack.

You can deploy a Citrix ADC VPX instance that uses SR-IOV technology, on OpenStack, in three steps:

- Enable SR-IOV Virtual Functions (VFs) on the host.
- Configure and make the VFs available to OpenStack.
- Provision the Citrix ADC VPX on OpenStack.

Prerequisites

Ensure that you:

- Add the Intel 82599 NIC (NIC) to the host.
- Download and Install the latest IXGBE driver from Intel.
- Block list the IXGBEVF driver on the host. Add the following entry in the /etc/modprobe.d/blacklist.conf file: Block list `ixgbevf`

> **Note**
>
> The `ixgbe` driver version must be minimum 5.0.4.

Enable SR-IOV VFs on the host

Do one of the following steps to enable SR-IOV VFs:

- If you are using a kernel version earlier than 3.8, add the following entry to the /etc/modprobe.d/ixgbe file and restart the host: options ixgbe max_vfs=<number_of_VFs>
- If you are using kernel 3.8 version or later, create VFs by using the following command:

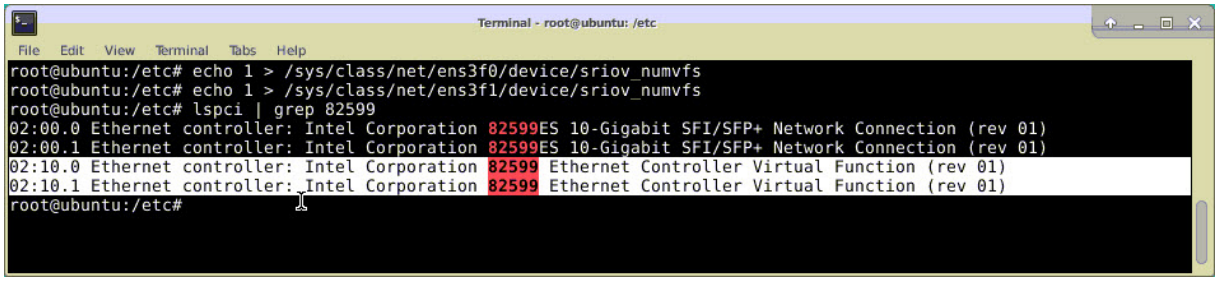

```
1     echo <number_of_VFs> > /sys/class/net/<device_name>/device/
         sriov_numvfs
2  <!--NeedCopy-->
```

Where:

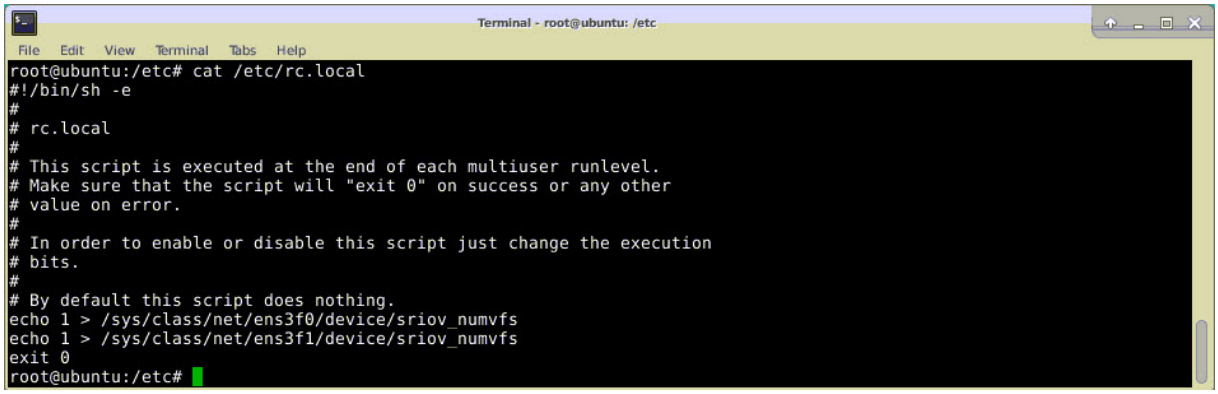
- number_of_VFs is the number of Virtual Functions that you want to create.
- device_name is the interface name.

> **Important**
>
> While you are creating the SR-IOV VFs, make sure that you do not assign MAC addresses to the VFs.

Here is an example of four VFs being created.



Make the VFs persistent, add the commands that you used to created VFs to the **rc.local** file. Here is an example showing content of rc.local file.



For more information, see this Intel SR-IOV Configuration Guide.

Configure and make the VFs available to OpenStack

Follow the steps given at the link below to configure SR-IOV on OpenStack: https://wiki.openstack.org/wiki/SR-IOV-Passthrough-For-Networking.

Provision the Citrix ADC VPX instance on OpenStack

You can provision a Citrix ADC VPX instance in an OpenStack environment by using the OpenStack CLI.

Provisioning a VPX instance, optionally involves using data from the config drive. The config drive is a special configuration drive that attaches to the instance when it boots. This configuration drive can be used to pass networking configuration information such as management IP address, network mask, and default gateway and so on to the instance before you configure the network settings for the instance.

When OpenStack provisions a VPX instance, it first detects that the instance is booting in an OpenStack environment, by reading a specific BIOS string (OpenStack Foundation) that indicates OpenStack. For Red Hat Linux distributions, the string is stored in /etc/nova/release. This is a standard mechanism that is available in all OpenStack implementations based on KVM hyper-visor platform. The drive must have a specific OpenStack label. If the config drive is detected, the instance attempts to read the following information from the file name specified in the `nova` boot command. In the procedures below, the file is called "userdata.txt."

- Management IP address
- Network mask
- Default gateway

Once the parameters are successfully read, they are populated in the NetScaler stack. This helps in managing the instance remotely. If the parameters are not read successfully or the config drive is not available, the instance transitions to the default behavior, which is:

- The instance attempts to retrieve the IP address information from DHCP.
- If DHCP fails or times-out, the instance comes up with default network configuration (192.168.100.1/16).

Provision the Citrix ADC VPX instance on OpenStack through CLI

You can provision a VPX instance in an OpenStack environment by using the OpenStack CLI. Here's the summary of the steps to provision a Citrix ADC VPX instance on OpenStack:

1. Extracting the `.qcow2` file from the .tgz file
2. Building an OpenStack image from the qcow2 image
3. Provisioning a VPX instance

To provision a VPX instance in an OpenStack environment, do the following steps.

1. Extract the. `qcow2` file from the `.tqz` file by typing the command:

```
tar xvzf <TAR file>
tar xvzf NSVPX-KVM-12.0-26.2_nc.tgz
```



```
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

2. Build an OpenStack image using the `.qcoz2` file extracted in step 1 by typing the following command:

```
glance image-create --name="<name of the OpenStack image>" --
    property hw_disk_bus=ide --is-public=true --container-format=
    bare --disk-format=qcow2< <name of the qcow2 file>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --is-public= true --container-format=bare --
    disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

The following illustration provides a sample output for the glance image-create command.

```
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 735dae4ea6e46e39ed3f0acfba02e755     |
| container_format | bare                                 |
| created_at       | 2017-02-16T10:03:29Z                 |
| disk_format      | qcow2                                |
| hw_disk_bus      | ide                                  |
| id               | aeaa13e9-b49b-411c-ab54-c61820a8e2f3 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | NSVPX-KVM-12.0-26.2                  |
| owner            | 06c41a73b32f4b48af55359fd7d3502c     |
| protected        | False                                |
| size             | 717946880                            |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-02-16T10:03:38Z                 |
| virtual_size     | None                                 |
| visibility       | private                              |
+------------------+--------------------------------------+
```

3. After an OpenStack image is created, provision the Citrix ADC VPX instance.

```
nova boot --image NSVPX-KVM-12.0-26.2 --config-drive=true --
    userdata
```



```
./userdata.txt --flavor m1. medium --nic net-id=3b258725-eaae-
455e-a5de-371d6d1f349f --nic port-id=218ba819-9f55-4991-adb6-
02086a6bdee2 NSVPX-10
<!--NeedCopy-->
```

In the preceding command, userdata.txt is the file which contains the details like, IP address, netmask, and default gateway for the VPX instance. The user data file is a user customizable file. NSVPX-KVM-12.0-26.2 is the name of the virtual appliance that you want to provision. –NIC port-id=218ba819-9f55-4991-adb6-02086a6bdee2 is the OpenStack VF.

The following illustration gives a sample output of the nova boot command.

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-SRV-ATTR:host	-
OS-EXT-SRV-ATTR:hypervisor_hostname	-
OS-EXT-SRV-ATTR:instance_name	instance-0000003c
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	-
OS-SRV-USG:terminated_at	-
accessIPv4	
accessIPv6	
adminPass	43EjPdM5shLz
config_drive	True
created	2017-02-20T11:53:37Z
flavor	m1.medium (3)
hostId	
id	6b9f6968-aab9-463c-b619-d58c73db3187
image	NSVPX-KVM-12.0-26.2 (a5478b8a-8435-48d1-b4a0-1494e2c8f8b1)
key_name	-
metadata	{}
name	NSVPX-10
os-extended-volumes:volumes_attached	[]
progress	0
security_groups	default
status	BUILD
tenant_id	06c41a73b32f4b48af55359fd7d3502c
updated	2017-02-20T11:53:38Z
user_id	418524f7101b4f0389ecbb36da9916b5

The following illustration shows a sample of the userdata.txt file. The values within the <PropertySection></PropertySection> tags are the values which are user configurable and holds the information like, IP address, netmask, and default gateway.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1">
<PlatformSection>
<Kind>NOVA</Kind>
<Version>2013.1</Version>
```



```
<Vendor>Openstack</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
   />
<Property oe:key="com.citrix.netscaler.platform" oe:value="vpx"/>
citrix.com 4
<Property oe:key="com.citrix.netscaler.orch_env"
oe:value="openstack-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip"
oe:value="10.1.0.100"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask"
oe:value="255.255.0.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway"
oe:value="10.1.0.1"/>
</PropertySection>
</Environment>
<!--NeedCopy-->
```

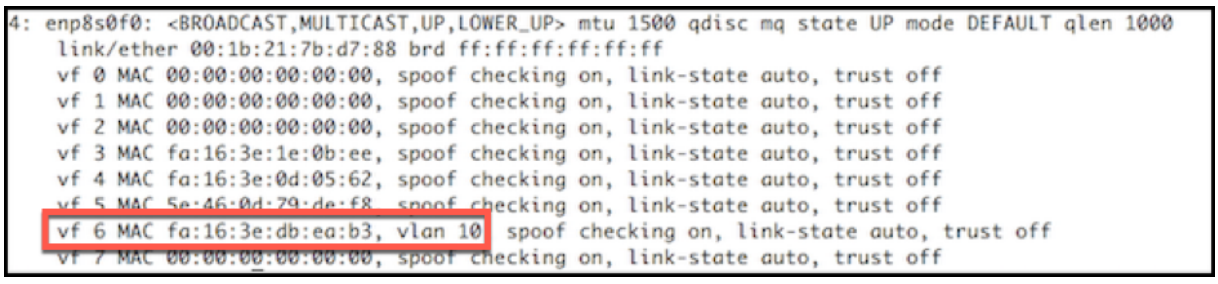
Additional supported Configurations: Creating and Deleting VLANs on SR-IOV VFs from the Host

Type the following command to create a VLAN on the SR-IOV VF:

```
ip link show enp8s0f0 vf 6 vlan 10
```

In the preceding command “enp8s0f0” is the name of the physical function.

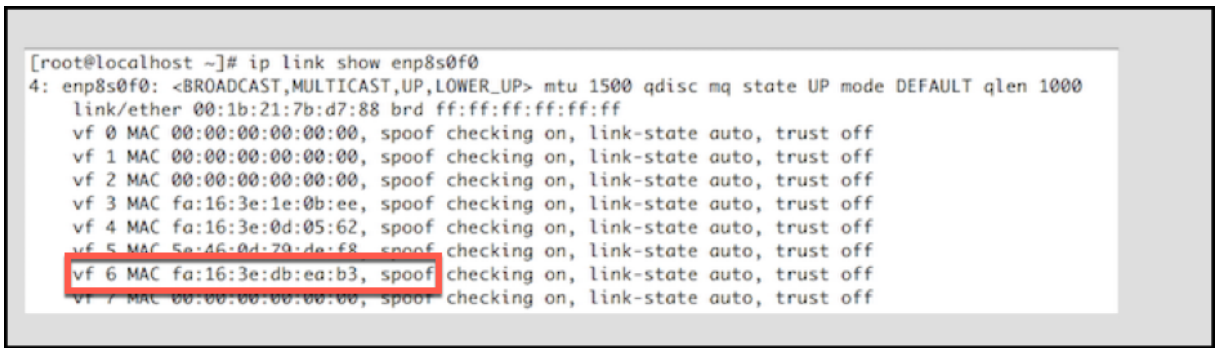
Example: VLAN 10, created on vf 6



Type the following command to delete a VLAN on the SR-IOV VF:

```
ip link show enp8s0f0 vf 6 vlan 0
```

Example: VLAN 10, removed from vf 6



These steps complete the procedure for deploying a Citrix ADC VPX instance that uses SRIOV technology, on OpenStack.

Configure a Citrix ADC VPX instance on KVM to use OVS DPDK-based host interfaces

September 14, 2021

You can configure a Citrix ADC VPX instance running on KVM (Fedora and RHOS) to use Open vSwitch (OVS) with Data Plane Development Kit (DPDK) for better network performance. This document describes how to configure the Citrix ADC VPX instance to operate on the `vhost-user` ports exposed by OVS-DPDK on the KVM host.

OVS is a multilayer virtual switch licensed under the open-source Apache 2.0 license. DPDK is a set of libraries and drivers for fast packet processing.

The following Fedora, RHOS, OVS, and DPDK versions are qualified for configuring a Citrix ADC VPX instance:

Fedora	RHOS
Fedora 25	RHOS 7.4
OVS 2.7.0	OVS 2.6.1
DPDK 16.11.12	DPDK 16.11.12

Prerequisites

Before you install DPDK, make sure the host has 1 GB huge pages.

For more information, see this DPDK system requirements documentation. Here is a summary of the steps required to configure a Citrix ADC VPX instance on KVM to use OVS DPDK-based host interfaces:

- Install DPDK.
- Build and Install OVS.
- Create an OVS bridge.
- Attach a physical interface to the OVS bridge.
- Attach `vhost-user` ports to the OVS data path.
- Provision a KVM-VPX with OVS-DPDK based `vhost-user` ports.

Install DPDK

To install DPDK, follow the instruction given at this Open vSwitch with DPDK document.

Build and install OVS

Download OVS from the OVS download page. Next, build, and install OVS by using a DPDK datapath. Follow the instructions given in the Installing Open vSwitch document.

For more detailed information, DPDK Getting Started Guide for Linux.

Create an OVS bridge

Depending on your need, type the Fedora or RHOS command to create an OVS bridge:

Fedora command:

```
1 > $OVS_DIR/utilities/ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0
     datapath_type=netdev
2 <!--NeedCopy-->
```

RHOS command:

```
1 ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0 datapath_type=netdev
2 <!--NeedCopy-->
```

Attach the physical interface to the OVS bridge

Bind the ports to DPDK and then attach them to the OVS bridge by typing the following Fedora or RHOS commands:

Fedora command:

```
1 > $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface
     dpdk0 type=dpdk  options:dpdk-devargs=0000:03:00.0
2
```



```
3 > $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface
     dpdk1 type=dpdk  options:dpdk-devargs=0000:03:00.1
4 <!--NeedCopy-->
```

RHOS command:

```
1 ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface dpdk0 type=dpdk
     options:dpdk-devargs=0000:03:00.0
2
3
4 ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface dpdk1 type=dpdk
     options:dpdk-devargs=0000:03:00.1
5 <!--NeedCopy-->
```

The `dpdk-devargs` shown as part of the options specifies the PCI BDF of the respective physical NIC.

Attach `vhost-user` ports to the OVS data path

Type the following Fedora or RHOS commands to attach `vhost-user` ports to the OVS data path:

Fedora command:

```
1 > $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user1 -- set
     Interface vhost-user1 type=dpdkvhostuser -- set Interface vhost-
     user1  mtu_request=9000
2
3 > $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user2 -- set
     Interface vhost-user2 type=dpdkvhostuser -- set Interface vhost-
     user2  mtu_request=9000
4
5 chmod g+w  /usr/local/var/run/openvswitch/vhost*
6 <!--NeedCopy-->
```

RHOS command:

```
1 ovs-vsctl add-port ovs-br0 vhost-user1 -- set Interface vhost-user1
     type=dpdkvhostuser -- set Interface vhost-user1  mtu_request=9000
2
3 ovs-vsctl add-port ovs-br0 vhost-user2 -- set Interface vhost-user2
     type=dpdkvhostuser -- set Interface vhost-user2  mtu_request=9000
4
5 chmod g+w /var/run/openvswitch/vhost*
6 <!--NeedCopy-->
```


Provision a KVM-VPX with OVS-DPDK-based vhost-user ports

You can provision a VPX instance on Fedora KVM with OVS-DPDK-based vhost-user ports only from the CLI by using the following QEMU commands:

Fedora command:

```
qemu-system-x86_64 -name KVM-VPX -cpu host -enable-kvm -m 4096M \

-object memory-backend-file,id=mem,size=4096M,mem-path=/dev/hugepages,
   share=on -numa node,memdev=mem \

-mem-prealloc -smp sockets=1,cores=2 -drive file=<absolute-path-to-disc
   -image-file>,if=none,id=drive-ide0-0-0,format=<disc-image-format> \

-device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0,
   bootindex=1 \

-netdev type=tap,id=hostnet0,script=no,downscript=no,vhost=on \

-device virtio-net-pci,netdev=hostnet0,id=net0,mac=52:54:00:3c:d1:ae,
   bus=pci.0,addr=0x3 \

-chardev socket,id=char0,path=</usr/local/var/run/openvswitch/vhost-
   user1> \

-netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce -device
   virtio-net-pci,mac=00:00:00:00:00:01,netdev=mynet1,mrg_rxbuf=on \

-chardev socket,id=char1,path=</usr/local/var/run/openvswitch/vhost-
   user2> \

-netdev type=vhost-user,id=mynet2,chardev=char1,vhostforce -device
   virtio-net

pci,mac=00:00:00:00:00:02,netdev=mynet2,mrg_rxbuf=on \

--nographic
<!--NeedCopy-->
```

For RHOS, use the following sample XML file to provision the Citrix ADC VPX instance, by using virsh.

```
<domain type='kvm'>

  <name>dpdk-vpx1</name>

```



```
<uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>

<memory unit='KiB'>16777216</memory>

<currentMemory unit='KiB'>16777216</currentMemory>

<memoryBacking>

  <hugepages>

    <page size='1048576' unit='KiB'/>

  </hugepages>

</memoryBacking>

<vcpu placement='static'>6</vcpu>

<cputune>

  <shares>4096</shares>

  <vcpupin vcpu='0' cpuset='0'/>

  <vcpupin vcpu='1' cpuset='2'/>

  <vcpupin vcpu='2' cpuset='4'/>

  <vcpupin vcpu='3' cpuset='6'/>

  <emulatorpin cpuset='0,2,4,6'/>

</cputune>

<numatune>

  <memory mode='strict' nodeset='0'/>

</numatune>

<resource>

  <partition>/machine</partition>

</resource>
```



```
50
51    <os>
52
53      <type arch='x86\_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
54
55      <boot dev='hd'/>
56
57    </os>
58
59    <features>
60
61      <acpi/>
62
63      <apic/>
64
65    </features>
66
67    <cpu mode='custom' match='minimum' check='full'>
68
69      <model fallback='allow'>Haswell-noTSX</model>
70
71      <vendor>Intel</vendor>
72
73      <topology sockets='1' cores='6' threads='1'/>
74
75      <feature policy='require' name='ss'/>
76
77      <feature policy='require' name='pcid'/>
78
79      <feature policy='require' name='hypervisor'/>
80
81      <feature policy='require' name='arat'/>
82
83  <domain type='kvm'>
84
85    <name>dpdk-vpx1</name>
86
87    <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>
88
89    <memory unit='KiB'>16777216</memory>
90
91    <currentMemory unit='KiB'>16777216</currentMemory>
92
93    <memoryBacking>
94
```



```
95      <hugepages>
96
97        <page size='1048576' unit='KiB'/>
98
99      </hugepages>
100
101   </memoryBacking>
102
103   <vcpu placement='static'>6</vcpu>
104
105   <cputune>
106
107     <shares>4096</shares>
108
109     <vcpupin vcpu='0' cpuset='0'/>
110
111     <vcpupin vcpu='1' cpuset='2'/>
112
113     <vcpupin vcpu='2' cpuset='4'/>
114
115     <vcpupin vcpu='3' cpuset='6'/>
116
117     <emulatorpin cpuset='0,2,4,6'/>
118
119   </cputune>
120
121   <numatune>
122
123     <memory mode='strict' nodeset='0'/>
124
125   </numatune>
126
127   <resource>
128
129     <partition>/machine</partition>
130
131   </resource>
132
133   <os>
134
135     <type arch='x86\_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
136
137     <boot dev='hd'/>
138
139   </os>
```



```
140
141    <features>
142
143      <acpi/>
144
145      <apic/>
146
147    </features>
148
149    <cpu mode='custom' match='minimum' check='full'>
150
151      <model fallback='allow'>Haswell-noTSX</model>
152
153      <vendor>Intel</vendor>
154
155      <topology sockets='1' cores='6' threads='1'/>
156
157      <feature policy='require' name='ss'/>
158
159      <feature policy='require' name='pcid'/>
160
161      <feature policy='require' name='hypervisor'/>
162
163      <feature policy='require' name='arat'/>
164
165      <feature policy='require' name='tsc\_adjust'/>
166
167      <feature policy='require' name='xsaveopt'/>
168
169      <feature policy='require' name='pdpe1gb'/>
170
171      <numa>
172
173        <cell id='0' cpus='0-5' memory='16777216' unit='KiB' memAccess='
             shared'/>
174
175      </numa>
176
177    </cpu>
178
179    <clock offset='utc'/>
180
181    <on\_poweroff>destroy</on\_poweroff>
182
183    <on\_reboot>restart</on\_reboot>
```



```
184
185   <on\_crash>destroy</on\_crash>
186
187   <devices>
188
189     <emulator>/usr/libexec/qemu-kvm</emulator>
190
191     <disk type='file' device='disk'>
192
193       <driver name='qemu' type='qcow2' cache='none'/>
194
195       <source file='/home/NSVPX-KVM-12.0-52.18\_nc.qcow2'/>
196
197       <target dev='vda' bus='virtio'/>
198
199       <address type='pci' domain='0x0000' bus='0x00' slot='0x07'
          function='0x0'/>
200
201     </disk>
202
203     <controller type='ide' index='0'>
204
205       <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
          function='0x1'/>
206
207     </controller>
208
209     <controller type='usb' index='0' model='piix3-uhci'>
210
211       <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
          function='0x2'/>
212
213     </controller>
214
215     <controller type='pci' index='0' model='pci-root'/>
216
217     <interface type='direct'>
218
219       <mac address='52:54:00:bb:ac:05'/>
220
221       <source dev='enp129s0f0' mode='bridge'/>
222
223       <model type='virtio'/>
224
225       <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
```



```
            function='0x0'/>
226
227       </interface>
228
229       <interface type='vhostuser'>
230
231         <mac address='52:54:00:55:55:56'/>
232
233         <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
              'client'/>
234
235         <model type='virtio'/>
236
237         <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
              function='0x0'/>
238
239       </interface>
240
241       <interface type='vhostuser'>
242
243         <mac address='52:54:00:2a:32:64'/>
244
245         <source type='unix' path='/var/run/openvswitch/vhost-user2' mode=
              'client'/>
246
247         <model type='virtio'/>
248
249         <address type='pci' domain='0x0000' bus='0x00' slot='0x05'
              function='0x0'/>
250
251       </interface>
252
253       <interface type='vhostuser'>
254
255         <mac address='52:54:00:2a:32:74'/>
256
257         <source type='unix' path='/var/run/openvswitch/vhost-user3' mode=
              'client'/>
258
259         <model type='virtio'/>
260
261         <address type='pci' domain='0x0000' bus='0x00' slot='0x06'
              function='0x0'/>
262
263       </interface>
```



```
264
265      <interface type='vhostuser'>
266
267        <mac address='52:54:00:2a:32:84'/>
268
269        <source type='unix' path='/var/run/openvswitch/vhost-user4' mode=
             'client'/>
270
271       <model type='virtio'/>
272
273        <address type='pci' domain='0x0000' bus='0x00' slot='0x09'
             function='0x0'/>
274
275      </interface>
276
277      <serial type='pty'>
278
279        <target port='0'/>
280
281      </serial>
282
283      <console type='pty'>
284
285        <target type='serial' port='0'/>
286
287      </console>
288
289      <input type='mouse' bus='ps2'/>
290
291      <input type='keyboard' bus='ps2'/>
292
293      <graphics type='vnc' port='-1' autoport='yes'>
294
295        <listen type='address'/>
296
297      </graphics>
298
299      <video>
300
301        <model type='cirrus' vram='16384' heads='1' primary='yes'/>
302
303        <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
             function='0x0'/>
304
305      </video>
```



```
306
307     <memballoon model='virtio'>
308
309       <address type='pci' domain='0x0000' bus='0x00' slot='0x08'
             function='0x0'/>
310
311     </memballoon>
312
313   </devices>
314
315 </domain
316 <!--NeedCopy-->
```

Points to note

In the XML file, the hugepage size must be 1 GB, as shown in the sample file.

```
1 <memoryBacking>
2
3     <hugepages>
4
5       <page size='1048576' unit='KiB'/>
6
7     </hugepages>
8 <!--NeedCopy-->
```

Also, in the sample file vhost-user1 is the vhost user port bound to ovs-br0.

```
1  <interface type='vhostuser'>
2
3        <mac address='52:54:00:55:55:56'/>
4
5        <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
             'client'/>
6
7        <model type='virtio'/>
8
9        <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
             function='0x0'/>
10
11     </interface>
12 <!--NeedCopy-->
```

To bring up the Citrix ADC VPX instance, start using the virsh command.

Citrix ADC VPX on AWS

October 14, 2021

You can launch a Citrix ADC VPX instance on Amazon Web Services (AWS). The Citrix ADC VPX appliance is available as an Amazon Machine Image (AMI) in AWS marketplace. A Citrix ADC VPX instance on AWS enables you to use AWS cloud computing capabilities and use Citrix ADC load balancing and traffic management features for their business needs. The VPX instance supports all the traffic management features of a physical Citrix ADC appliance, and it can be deployed as standalone instances or in HA pairs. For more information on VPX features, see the VPX data sheet.

Getting started

Before you get started with your VPX deployment, you must be familiar with the following information:

- AWS terminology
- AWS-VPX support matrix
- Limitations and usage guidelines
- Prerequisites
- How a Citrix ADC VPX instance on AWS works

Deploy a Citrix ADC VPX instance on AWS

In AWS, the following deployment types are supported for VPX instances:

- Standalone
- High availability (Acitve-Passive)
  - High availability within same zone
  - High availability across different zones using Elastic IP
  - High availability across different zones using Private IP
- Active-Active GSLB
- Autoscaling (Active-Active) using ADM

Hybrid Deployments

- Deploy Citrix ADC in AWS Outpost
- Deploy Citrix ADC in VMC in AWS

Licensing

A Citrix ADC VPX instance on AWS requires a license. The following licensing options are available for Citrix ADC VPX instances running on AWS:

- Free (unlimited)
- Hourly
- Annual
- BYOL
- Free Trial (all Citrix ADC VPX-AWS subscription offerings for 21 days free in AWS marketplace.)

Automation

- Citrix ADM: Smart Deployment
- AWS Quick Starts: Citrix ADC VPX for Web Applications on AWS
- GitHub CFTs: Citrix ADC templates and scripts for AWS deployment
- GitHub Ansible: Citrix ADC templates and scripts for AWS deployment
- GitHub Terraform: Citrix ADC templates and scripts for AWS deployment
- AWS Pattern Library (PL): Citrix ADC VPX

Blogs

- How Citrix ADC on AWS Helps Customers Deliver Applications Securely
- Application delivery in hybrid cloud with Citrix ADC and AWS
- Citrix is an AWS Networking Competency Partner
- Citrix ADC: Always ready for public clouds
- Scale out or scale in with ease in public clouds through Citrix ADC
- Citrix expands ADC deployment choice with AWS Outposts
- Using Citrix ADC with Amazon VPC ingress routing
- Citrix delivers choice, performance, and simplified deployment in AWS
- The security of Citrix Web App Firewall – now on the AWS Marketplace
- How Aria Systems uses Citrix Web App Firewall on AWS

Videos

- Simplifying public cloud Citrix ADC deployments through ADM
- Provisioning and configuring Citrix ADC VPX in AWS using ready-to-use terraform scripts
- Deploy Citrix ADC HA in AWS using CloudFormation Template
- Deploy Citrix ADC HA across Availability Zones using AWS QuickStart
- How to deploy Citrix ADC in AWS
- Citrix ADC Autoscale using ADM
- Citrix ADC supporting back-end server auto scaling in AWS or AWS Autoscaling group

Customer case studies

- Technology Solution - Xenit AB
- A better way to do business with Citrix and AWS cloud – Aria
- Discover the Citrix ADC and AWS advantage
- Rain for Rent - Customer story

Solutions

- Deploy digital advertising platform on AWS with Citrix ADC
- Enhancing Clickstream analytics in AWS using Citrix ADC

Support

- Open a Support case
- For Citrix ADC subscription offering, see Troubleshoot a VPX instance on AWS. To file a support case, find your AWS account number and support PIN code, and call Citrix support.
- For Citrix ADC Customer Licensed offering or BYOL, ensure that you have the valid support and maintenance agreement. If you do not have an agreement, contact your Citrix representative.

Additional References

- AWS On-Demand Webinar - Citrix ADC on AWS
- Deployment Guides for Citrix ADC VPX on AWS
- Creating a VPX Amazon Machine Image (AMI) in SC2S/secret region
- Citrix ADC on AWS

- Citrix ADC and AWS Validated Reference Design
- Citrix ADC VPX data sheet
- Citrix ADC in AWS Marketplace
- Citrix ADC is part of AWS networking partner solutions (load balancers)
- Citrix ADC for VMware cloud on AWS
- AWS FAQs

AWS terminology

September 14, 2021

This section describes the list of commonly used AWS terms and phrases. For more information, see AWS Glossary.

Term	Definition
Amazon Machine Image (AMI)	A machine image, which provides the information required to launch an instance, which is a virtual server in the cloud.
Elastic Block Store	Provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.
Simple Storage Service (S3)	Storage for the Internet. It is designed to make web-scale computing easier for developers.
Elastic Compute Cloud (EC2)	A web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
Elastic Load Balancing (ELB)	Distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones. This increases the fault tolerance of your applications.
Elastic network interface (ENI)	A virtual network interface that you can attach to an instance in a Virtual Private Cloud (VPC).

Term	Definition
Elastic IP (EIP) address	A static, public IPv4 address that you have allocated in Amazon EC2 or Amazon VPC and then attached to an instance. Elastic IP addresses are associated with your account, not a specific instance. They are elastic because you can easily allocate, attach, detach, and free them as your needs change.
Instance type	Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.
Identity and Access Management (IAM)	An AWS identity with permission policies that determine what the identity can and cannot do in AWS. You can use an IAM role to enable applications running on an EC2 instance to securely access your AWS resources. IAM role is required for deploying VPX instances in a high-availability setup.
Internet Gateway	Connects a network to the Internet. You can route traffic for IP addresses outside your VPC to the Internet gateway.
Key pair	A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key.
Route tables	A set of routing rules that controls the traffic leaving any subnet that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.
Security groups	A named set of allowed inbound network connections for an instance.

Term	Definition
Subnets	A segment of the IP address range of a VPC that EC2 instances can be attached to. You can create subnets to group instances according to security and operational needs.
Virtual Private Cloud (VPC)	A web service for provisioning a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define.
Auto Scaling	A web service to launch or terminate Amazon EC2 instances automatically based on user-defined policies, schedules, and health checks.
CloudFormation	A service for writing or changing templates that create and delete related AWS resources together as a unit.

VPX-AWS support matrix

October 7, 2021

The following tables list the supported VPX model and AWS regions, instance types, and services.

Table 1: Supported VPX models on AWS

Supported VPX model
Citrix ADC VPX Standard/Advanced/Premium Edition - 200 Mbps
Citrix ADC VPX Standard/Advanced/Premium Edition - 1000 Mbps
Citrix ADC VPX Standard/Advanced/Premium Edition - 3 Gbps
Citrix ADC VPX Standard/Advanced/Premium Edition - 5 Gbps
Citrix ADC VPX Standard/Advanced/Premium - 10 Mbps
Citrix ADC VPX Express - 20 Mbps
Citrix ADC VPX - Customer Licensed
Citrix ADC (formerly NetScaler) VPX FIPS - Customer Licensed

Table: 2 Supported AWS regions

Supported AWS regions
US West (Oregon) Region
US West (N. California) Region
US East (Ohio) Region
US East (N. Virginia) Region
Asia Pacific (Mumbai) Region
Asia Pacific (Seoul) Region
Canada (Central) Region
Asia Pacific (Singapore) Region
Asia Pacific (Sydney) Region
Asia Pacific (Tokyo) Region
Asia Pacific (Hong Kong) Region
Canada (Central) Region
China (Beijing) Region
China (Ningxia) region
EU (Frankfurt) Region
EU (Ireland) Region
EU (London) Region
EU (Paris) Region
EU (Milan) Region
South America (São Paulo) Region
AWS GovCloud (US-East) Region
AWS GovCloud (US-West) Region
AWS Top Secret (C2S) Region
Middle East (Bahrain) Region
Africa (Cape Town)
C2S

Table 3: Supported AWS instance types

Supported AWS instance types
t2.medium, t2.large, t2.x large, t2.2x large
m3.large, m3.x large, m3.2x large
c4.large, c4.large, c4.2x large, c4.4x large, c4.8x large
m4.large, m4.large, m4.2x large, m4.4x large, m4.10x large
m5.large, m5.x large, m5.2x large, m5.4x large, m5.12x large, m5.24x large
c5.large, c5.x large, c5.2x large, c5.4x large, c5.9x large, c5.18x large, c5.24x large
C5n.large, C5n.x large, C5n.2x large, C5n.4x large, C5n.9x large, C5n.18x large
D2.x large, D2.2x large, D2.4x large, D2.8x large
m5a.large, m5a.xlarge, m5a.2xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge

Table 4: Supported AWS Services

Supported AWS services
EC2: Launches ADC instances.
Lambda: Invokes Citrix ADC VPX NITRO APIs during provisioning of Citrix ADC VPX instances from CFT.
VPC and VPC ingress routing: VPC creates isolated networks in which ADC can be launched. VPC ingress routing is used in the firewall load balancing solution.
Route53: Distributes traffic across all the ADC VPX nodes in the Citrix ADC Autoscale solution.
ELB: Distributes traffic across all the ADC VPX nodes in the Citrix ADC Autoscale solution.
Cloudwatch: Monitors performance and system parameters for Citrix ADC VPX instance.
AWS Autoscaling: Used for back-end server autoscaling.
Cloud formation: CloudFormation templates are used to deploy Citrix ADC VPX instances.
Simple Queue Service (SQS): Monitors scale up and scale down events in back-end autoscaling.
Simple Notification Service (SNS): Monitors scale up and scale down events in back-end autoscaling.
Identity and Access Management (IAM): Provides access to AWS services and resources.
AWS Outposts: Provisions Citrix ADC VPX instances in AWS Outposts.

Citrix recommends the following AWS instance types:

- M5 and C5n series for marketplace editions or bandwidth-based pool licensing.
- C5n series for vCPU-based pool licensing.

VPX offering in AWS marketplace	AWS instance recommendation
VPX 10, VPX Express 20, VPX 200	M5.xLarge
VPX 1000, VPX 3G, VPX 5G	M5.2xLarge

Citrix recommends the following AWS instance types based on throughput.

VPX with Pooled licensing (Bandwidth licenses)	AWS instance recommendation
VPX 8G	C5n.4xLarge
VPX 10G, VPX 15G, VPX 25G	C5n.9xLarge

> Note:
>
> The VPX 25G offering doesn't give the desired 25G throughput in AWS but can give higher SSL transactions rate.

To achieve throughput more than 5G, do the following:

- Choose **Citrix ADC VPX - Customer Licensed (BYOL)** offering in AWS marketplace.
- Select **Pooled Licensing (Bandwidth licenses)** in Citrix ADC GUI or CLI.

To determine your instance based on different metrics such as packets per second, SSL transactions rate, reach out to your Citrix contact for guidance. For vCPU based Pool licensing and sizing guidance, reach out to Citrix support.

Limitations and usage guidelines

September 14, 2021

The following limitations and usage guidelines apply when deploying a Citrix ADC VPX instance on AWS:

- Before you start, read the AWS terminology section in Deploy a Citrix ADC VPX instance on AWS.
- The clustering feature is not supported for VPX.

- For the high availability setup to work effectively, associate a dedicated NAT device to management Interface or associate EIP to NSIP. For more information on NAT, in the AWS documentation, see NAT Instances.
- Data traffic and management traffic must be segregated with ENIs belonging to different subnets.
- Only the NSIP address must be present on the management ENI.
- If a NAT instance is used for security instead of assigning an EIP to the NSIP, appropriate VPC level routing changes are required. For instructions on making VPC level routing changes, in the AWS documentation, see Scenario 2: VPC with Public and Private Subnets.
- A VPX instance can be moved from one EC2 instance type to another (for example, from m3.large to an m3.xlarge).
- For storage options for VPX on AWS, Citrix recommends EBS, because it is durable and the data is available even after it is detached from the instance.
- Dynamic addition of ENIs to VPX is not supported. Restart the VPX instance to apply the update. Citrix recommends you to stop the standalone or HA instance, attach the new ENI, and then restart the instance.
- You can assign multiple IP addresses to an ENI. The maximum number of IP addresses per ENI is determined by the EC2 instance type, see the section "IP Addresses Per Network Interface Per Instance Type" in Elastic Network Interfaces. You must allocate the IP addresses in AWS before you assign them to ENIs. For more information, see Elastic Network Interfaces.
- Citrix recommends that you avoid using the enable and disable interface commands on Citrix ADC VPX interfaces.
- The Citrix ADC `set ha node \<NODE\_ID\> -haStatus STAYPRIMARY` and `set ha node \<NODE\_ID\> -haStatus STAYSECONDARY` commands are disabled by default.
- IPv6 is not supported for VPX.
- Due to AWS limitations, these features are not supported:
  - Gratuitous ARP(GARP)
  - L2 mode
  - Tagged VLAN
  - Dynamic Routing
  - virtual MAC
- For RNAT to work, ensure **Source/Destination Check** is disabled. For more information, see "Changing the Source/Destination Checking" in Elastic Network Interfaces.
- In a Citrix ADC VPX deployment on AWS, in some AWS regions, the AWS infrastructure might not be able to resolve AWS API calls. This happens if the API calls are issued through a nonmanage-

ment interface on the Citrix ADC VPX instance.
As a workaround, restrict the API calls to the management interface only. To do that, create an NSVLAN on the VPX instance and bind the management interface to the NSVLAN by using the appropriate command.
For example:

```
set ns config -nsvlan <vlan id> -ifnum 1/1 -tagged NO
save config
```

Restart the VPX instance at the prompt. For more information about configuring `nsvlan`, see Configuring NSVLAN.

- In the AWS console, the vCPU usage shown for a VPX instance under the **Monitoring** tab might be high (up to 100 percent), even when the actual usage is much lower. To see the actual vCPU usage, navigate to **View all CloudWatch metrics**. For more information, see Monitor your instances using Amazon CloudWatch.

Prerequisites

September 14, 2021

Before attempting to create a VPX instance in AWS, ensure you have the following:

- **An AWS account**: to launch a Citrix ADC VPX AMI in an Amazon Web Services (AWS) Virtual Private Cloud (VPC). You can create an AWS account for free at www.aws.amazon.com.
- **An AWS Identity and Access Management (IAM) user account**: to securely control access to AWS services and resources for your users. For more information about how to create an IAM user account, see the topic Creating IAM Users (Console).

An IAM role is mandatory for both standalone and high availability deployments. The IAM role must have the following privileges:

```
ec2:DescribeInstances
ec2:DescribeNetworkInterfaces
ec2:DetachNetworkInterface
ec2:AttachNetworkInterface
ec2:StartInstances
ec2:StopInstances
ec2:RebootInstances
ec2:DescribeAddresses
ec2:AssociateAddress
ec2:DisassociateAddress
ec2:AssignPrivateIpAddresses
ec2:UnAssignPrivateIpAddresses
autoscaling:*
```



```
14 sns:CreateTopic
15 sns:DeleteTopic
16 sns:ListTopics
17 sns:Subscribe
18 sqs:CreateQueue
19 sqs:ListQueues
20 sqs:DeleteMessage
21 sqs:GetQueueAttributes
22 sqs:SetQueueAttributes
23 iam:SimulatePrincipalPolicy
24 iam:GetRole
25 <!--NeedCopy-->
```

If you use the Citrix CloudFormation template, the IAM role is automatically created. The template does not allow selecting an already created IAM role.

> Note
>
> When you log on to the VPX instance through the GUI, a prompt to configure the required privileges for the IAM role appears. Ignore the prompt if you've already configured the privileges.

- **AWS CLI**: To use all the functionality provided by the AWS Management Console from your terminal program. For more information, see the AWS CLI user guide. You also need the AWS CLI to change the network interface type to SR-IOV.
- **Elastic Network Adapter (ENA)**: For ENA driver-enabled instance type, the firmware version must be 13.0 and above.

How a Citrix ADC VPX instance on AWS works

September 14, 2021

The Citrix ADC VPX instance is available as an AMI in AWS marketplace, and it can be launched as an EC2 instance within an AWS VPC. The Citrix ADC VPX AMI instance requires a minimum of 2 virtual CPUs and 2 GB of memory. An EC2 instance launched within an AWS VPC can also provide the multiple interfaces, multiple IP addresses per interface, and public and private IP addresses needed for VPX configuration. Each VPX instance requires at least three IP subnets:

- A management subnet
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP, MIP, and so on)

Citrix recommends three network interfaces for a standard VPX instance on AWS installation.

AWS currently makes multi-IP functionality available only to instances running within an AWS VPC. A VPX instance in a VPC can be used to load balance servers running in EC2 instances. An Amazon VPC allows you to create and control a virtual networking environment, including your own IP address range, subnets, route tables, and network gateways.

Note: By default, you can create up to 5 VPC instances per AWS region for each AWS account. You can request higher VPC limits by submitting Amazon's request form http://aws.amazon.com/contact-us/vpc-request.

Figure 1. A Sample Citrix ADC VPX Instance Deployment on AWS Architecture

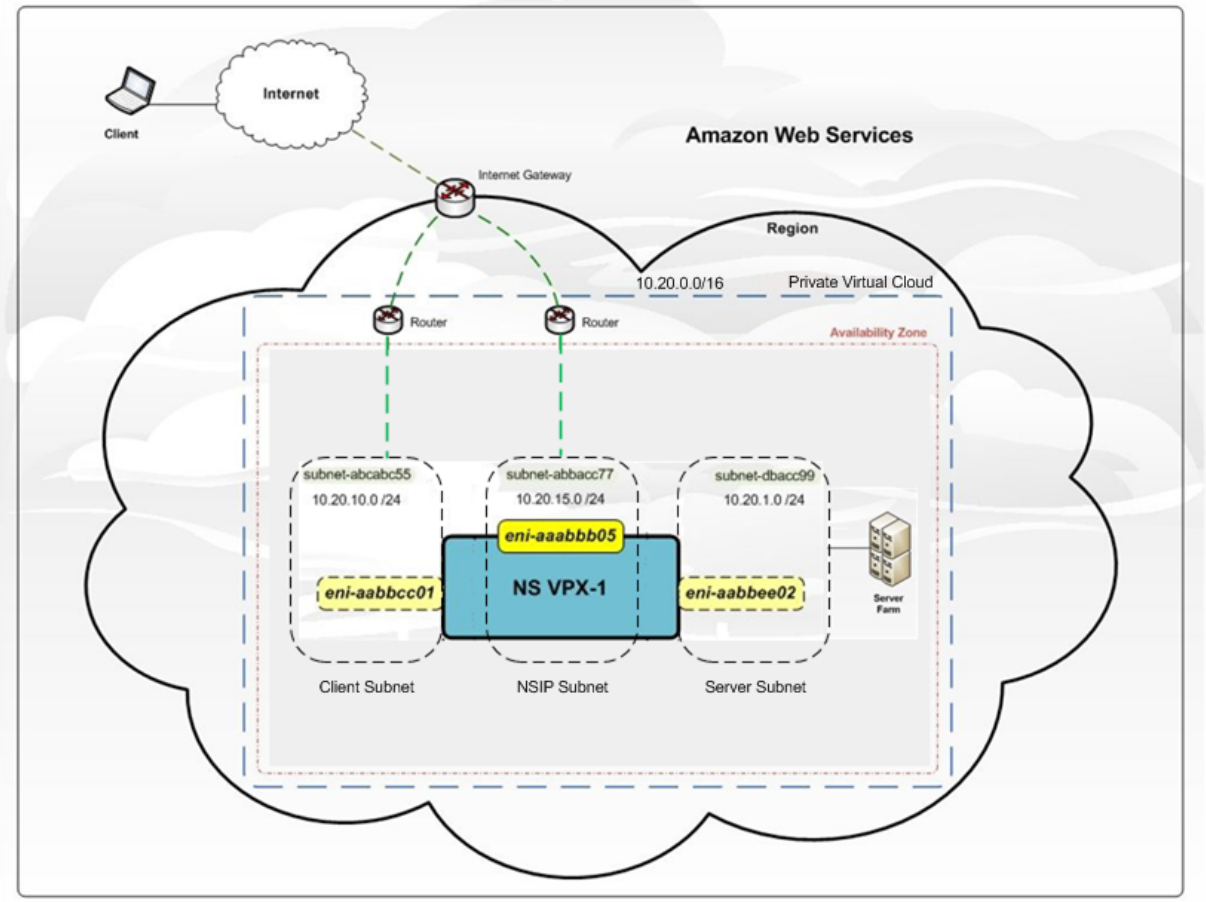


Figure 1 shows a simple topology of an AWS VPC with a Citrix ADC VPX deployment. The AWS VPC has:

1. A single Internet gateway to route traffic in and out of the VPC.
2. Network connectivity between the Internet gateway and the Internet.
3. Three subnets, one each for management, client, and server.
4. Network connectivity between the Internet gateway and the two subnets (management and client).
5. A standalone Citrix ADC VPX instance deployed within the VPC. The VPX instance has three ENIs,

one attached to each subnet.

Deploy a Citrix ADC VPX standalone instance on AWS

September 14, 2021

You can deploy a Citrix ADC VPX standalone instance on AWS by using the following options:

- AWS web console
- Citrix-authored CloudFormation template
- AWS CLI

This topic describes the procedure for deploying a Citrix ADC VPX instance on AWS.

Before you start your deployment, read the following topics:

- Prerequisites
- Limitation and usage guidelines

Deploy a Citrix ADC VPX instance on AWS by using the AWS web console

You can deploy a Citrix ADC VPX instance on AWS through the AWS web console. The deployment process includes the following steps:

1. Create a Key Pair
2. Create a Virtual Private Cloud (VPC)
3. Add more subnets
4. Create security groups and security rules
5. Add route tables
6. Create an internet gateway
7. Create a Citrix ADC VPX instance
8. Create and attach more network interfaces
9. Attach elastic IPs to the management NIC
10. Connect to the VPX instance

Step 1: Create a key pair.

Amazon EC2 uses a key pair to encrypt and decrypt logon information. To log on to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

When you review and launch an instance by using the AWS Launch Instance wizard, you are prompted to use an existing key pair or create a new key pair. More more information about how to create a key pair, see Amazon EC2 Key Pairs.

Step 2: Create a VPC.

A Citrix ADC VPC instance is deployed inside an AWS VPC. A VPC allows you to define the virtual network dedicated to your AWS account. For more information about AWS VPC, see Getting Started With Amazon VPC.

While creating a VPC for your Citrix ADC VPX instance, keep the following points in mind.

- Use the VPC with a Single Public Subnet Only option to create an AWS VPC in an AWS availability zone.
- Citrix recommends that you create at least **three subnets**, of the following types:
  - One subnet for management traffic. You place the management IP(NSIP) on this subnet. By default elastic network interface (ENI) eth0 is used for management IP.
  - One or more subnets for client-access (user-to-Citrix ADC VPX) traffic, through which clients connect to one or more virtual IP (VIP) addresses assigned to Citrix ADC load balancing virtual servers.
  - One or more subnets for the server-access (VPX-to-server) traffic, through which your servers connect to VPX-owned subnet IP (SNIP) addresses. For more information about Citrix ADC load balancing and virtual servers, virtual IP addresses (VIPs), and subnet IP addresses (SNIPs), see:
  - All subnets must be in the same availability zone.

Step 3: Add subnets.

When you used the VPC wizard, only one subnet was created. Depending on your requirement, you might want to create more subnets. For more information about how to create more subnets, see Adding a Subnet to Your VPC.

Step 4: Create security groups and security rules.

To control inbound and outbound traffic, create security groups and add rules to the groups. For more information how to create groups and add rules, see Security Groups for Your VPC.

For Citrix ADC VPX instances, the EC2 wizard gives default security groups, which are generated by AWS Marketplace and is based on recommended settings by Citrix. However, you can create more security groups based on your requirements.

> **Note**
>
> Port 22, 80, 443 to be opened on the Security group for SSH, HTTP, and HTTPS access respectively.

Step 5: Add route tables.

Route table contains a set of rules, called routes, that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table. For more information about how to create a route table, see Route Tables.

Step 6: Create an internet gateway.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

Create an internet gateway for internet traffic. For more information about how to create an Internet Gateway, see the section Attaching an Internet Gateway.

Step 7: Create a Citrix ADC VPX instance by using the AWS EC2 service.

To create a Citrix ADC VPX instance by using the AWS EC2 service, complete the following steps.

1. From the AWS dashboard, go to **Compute > EC2 > Launch Instance > AWS Marketplace**.

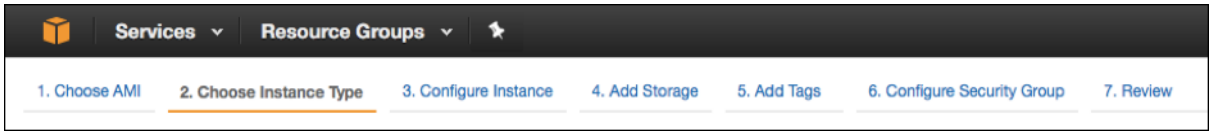
   Before you click **Launch Instance**, ensure your region is correct by checking the note that appears under **Launch Instance**.

2. In the Search AWS Marketplace bar, search with the keyword Citrix ADC VPX.
3. Select the version you want to deploy and then click **Select**. For the Citrix ADC VPX version, you have the following options:

   - A licensed version
   - Citrix ADC VPX Express appliance (This is a free virtual appliance, which is available from Citrix ADC 12.0 56.20.)
   - Bring your own device

The Launch Instance wizard starts. Follow the wizard to create an instance. The wizard prompts you to:

- Choose Instance Type
- Configure Instance
- Add Storage
- Add Tags
- Configure Security Group
- Review



Step 8: Create and attach more network interfaces.

Create two more network interfaces for VIP and SNIP. For more information about how to create more network interfaces, see the Creating a Network Interface section.

After you've created the network interfaces, you must attach them to the VPX instance. Before attaching the interface, shut down the VPX instance, attach the interface, and power on the instance. For more information about how to attach network interfaces, see the Attaching a Network Interface When Launching an Instance section.

Step 9: Allocate and associate elastic IPs.

If you assign a public IP address to an EC2 instance, it remains assigned only until the instance is stopped. After that, the address is released back to the pool. When you restart the instance, a new public IP address is assigned.

In contrast, an elastic IP (EIP) address remains assigned until the address is disassociated from an instance.

Allocate and associate an elastic IP for the management NIC. For more information about how to allocate and associate elastic IP addresses, see these topics:

- Allocating an Elastic IP Address
- Associating an Elastic IP Address with a Running Instance

These steps complete the procedure to create a Citrix ADC VPX instance on AWS. It can take a few minutes for the instance to be ready. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column on the Instances page.

Step 10: Connect to the VPX instance.

After you've created the VPX instance, you connect the instance by using the GUI and an SSH client.

- GUI

The following are the default administrator credentials to access a Citrix ADC VPX instance

User name: `nsroot`

Password: The default password for the ns root account is set to the AWS instance-ID of the Citrix ADC VPX instance. On your first logon, you are prompted to change the password for security reasons. After changing the password, you must save the configuration. If the configuration is not saved and the instance restarts, you must log on with the default password. Change the password again at the prompt.

- SSH client

From the AWS management console, select the Citrix ADC VPX instance and click **Connect**. Follow the instructions given on the **Connect to Your Instance** page.

For more information about how to deploy a Citrix ADC VPX standalone instance on AWS by using the AWS web console, see:

- Scenario: standalone instance
- How to configure a Citrix NetScaler VPX instance on AWS by using Citrix CloudFormation template

Configure a Citrix ADC VPX instance by using the Citrix CloudFormation template

You can use the Citrix-provided CloudFormation template to automate VPX instance launch. The template provides functionality to launch a single Citrix ADC VPX instance, or to create a high availability environment with a pair of Citrix ADC VPX instances.

You can launch the template from AWS Marketplace or GitHub.

The CloudFormation template requires an existing VPC environment, and it launches a VPX instance with three elastic network interfaces (ENIs). Before you start the CloudFormation template, ensure that you complete the following requirements:

- An AWS virtual private cloud (VPC)
- Three subnets within the VPC: one for management, one for client traffic, and one for back-end servers
- An EC2 key pair to enable SSH access to the instance
- A security group with UDP 3003, TCP 3009–3010, HTTP, SSH ports open

See the "Deploy a Citrix ADC VPX Instance on AWS by Using the AWS Web Console" section or AWS documentation for more information about how to complete the prerequisites.

Watch this video to learn about how to configure and launch a Citrix ADC VPX standalone instance by using the Citrix CloudFormation template available in the AWS Marketplace.

Further, you configure and launch a Citrix ADC VPX Express standalone instance by using the Citrix CloudFormation template available in GitHub:

https://github.com/citrix/citrix-adc-aws-cloudformation/tree/master/templates/standalone/

An IAM role is not mandatory for a standalone deployment. However, Citrix recommends that you create and attach an IAM role with the required privileges to the instance, for future need. The IAM role ensures that the standalone instance is easily converted to a high availability node with SR-IOV, when required.

For more information about the required privileges, see Configuring Citrix ADC VPX instances to Use SR-IOV Network Interface.

> Note
> If you deploy a Citrix ADC VPX instance on AWS by using the AWS web console, the CloudWatch

> service is enabled by default. If you deploy a Citrix ADC VPX instance by using the Citrix CloudFormation template, the default option is "Yes." If you want to disable the CloudWatch service, select "No." For more information, see Monitor your instances using Amazon CloudWatch

Configure a Citrix ADC VPX instance by using the AWS CLI

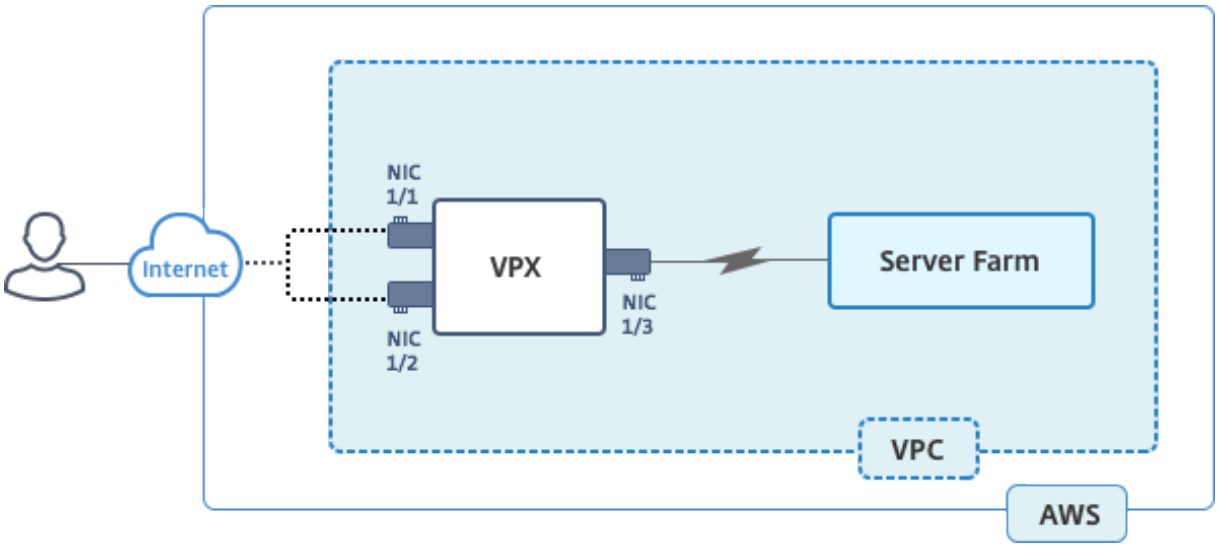
You can use the AWS CLI to launch instances. For more information, see the AWS Command Line Interface Documentation.

Scenario: standalone instance

September 14, 2021

This scenario illustrates how to deploy a Citrix ADC VPX standalone EC2 instance in AWS by using the AWS GUI. Create a standalone VPX instance with three NICs. The instance, which is configured as a load balancing virtual server, communicates with back-end servers (the server farm). For this configuration, set up the required communication routes between the instance and the back-end servers, and between the instance and the external hosts on the public internet.

For more details about the procedure for deploying a VPX instance, see Deploy a Citrix ADC VPX standalone instance on AWS.

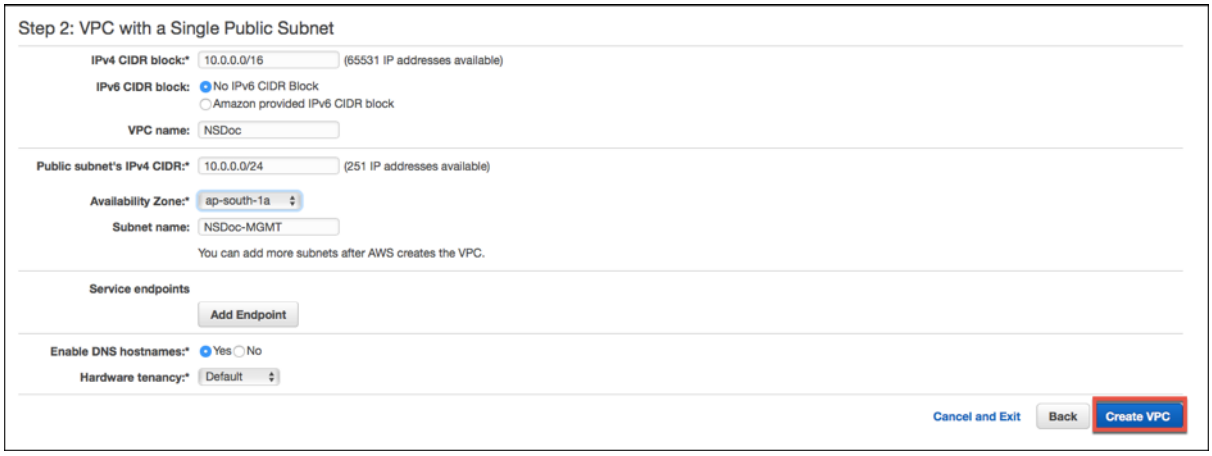


Create three NICs. Each NIC can be configured with a pair of IP addresses (public and private). The NICs serve the following purposes.

NIC	Purpose	Associated with
eth0	Serves management traffic (NSIP)	A public IP address and a private IP address
eth1	Serves client-side traffic (VIP)	A public IP address and a private IP address
eth2	Communicates with back-end servers (SNIP)	A public IP address (Private IP address not mandatory)

Step 1: Create a VPC.

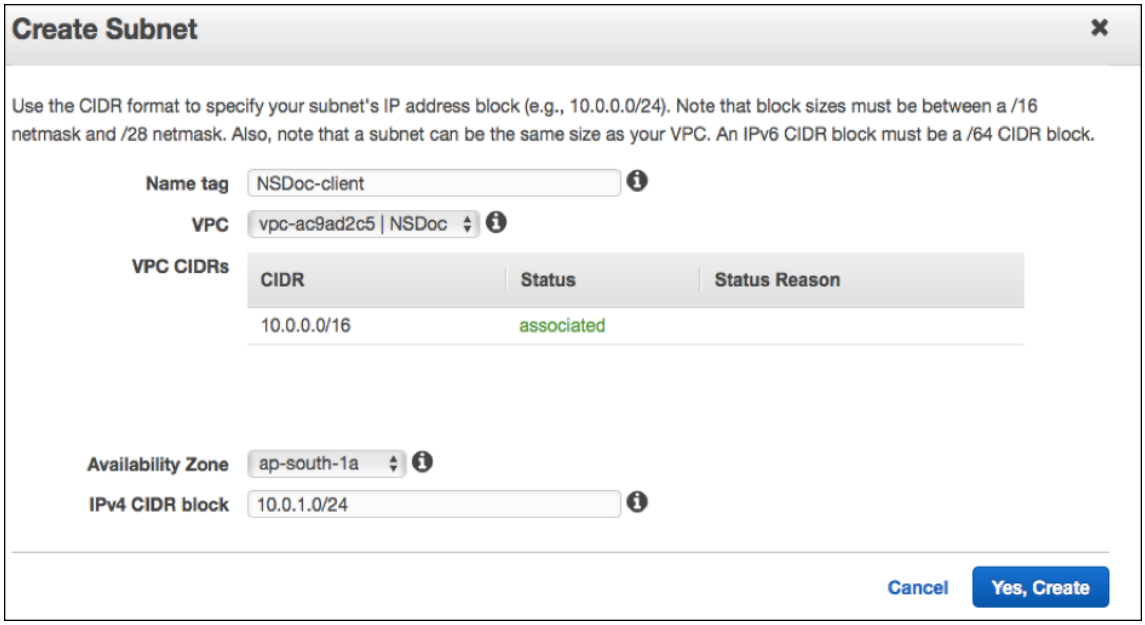
1. Log on to the AWS web console and navigate to **Networking & Content Delivery** > **VPC**. Click **Start VPC Wizard**.
2. Select **VPC with a Single Public Subnet** and click **Select**.
3. Set the IP CIDR Block to 10.0.0.0/16, for this scenario.
4. Give a name for the VPC.
5. Set the public subnet to 10.0.0.0/24. (This is the management network).
6. Select an availability zone.
7. Give a name for the subnet.
8. Click Create **VPC**.



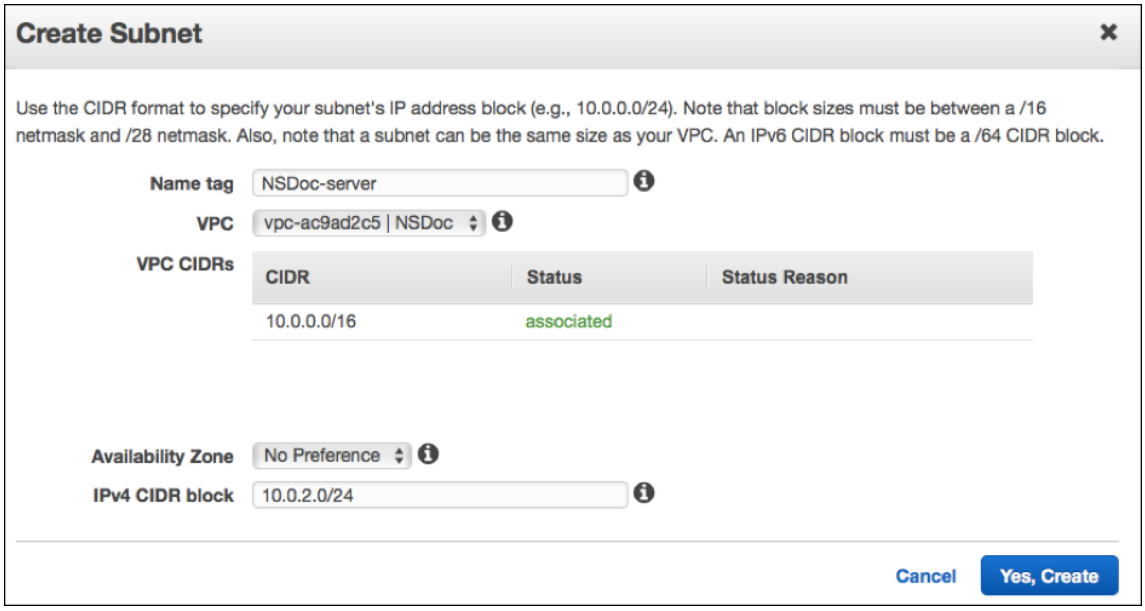
Step 2: Create extra subnets.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Subnets, Create Subnet after you enter the following details.
   - Name tag: Provide a name for your subnet.

- VPC: Choose the VPC for which you're creating the subnet.
- Availability Zone: Choose the availability zone in which you created the VPC in step 1.
- IPv4 CIDR block: Specify an IPv4 CIDR block for your subnet. For this scenario, choose 10.0.1.0/24.



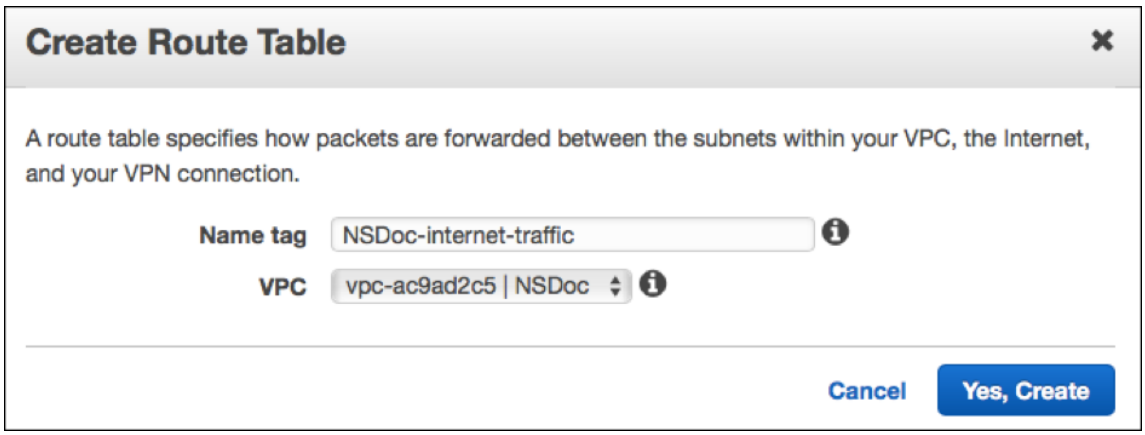
3. Repeat the steps to create one more subnet for back-end servers.



Step 3: Create a route table.

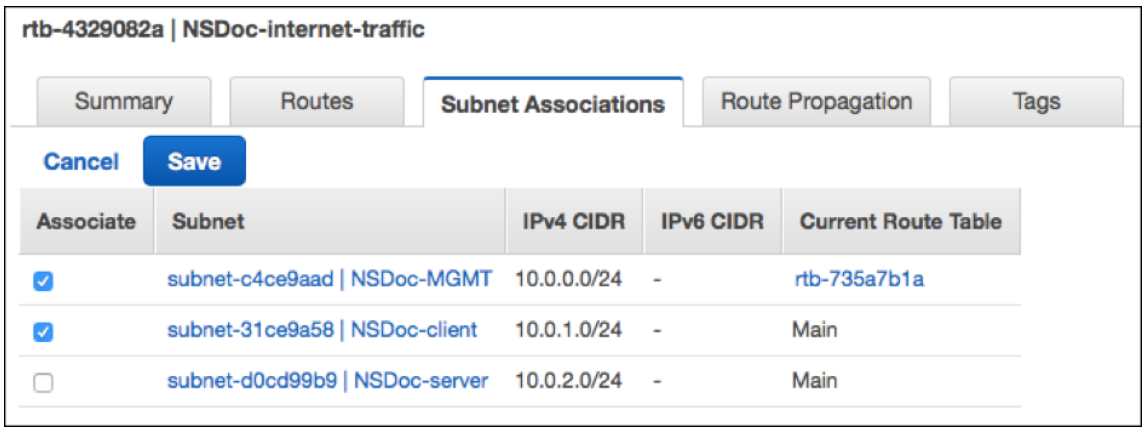
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route Tables** > **Create Route Table.**
3. In the Create Route Table window, add a name and select the VPC that you created in step 1.

4. Click **Yes, Create**.

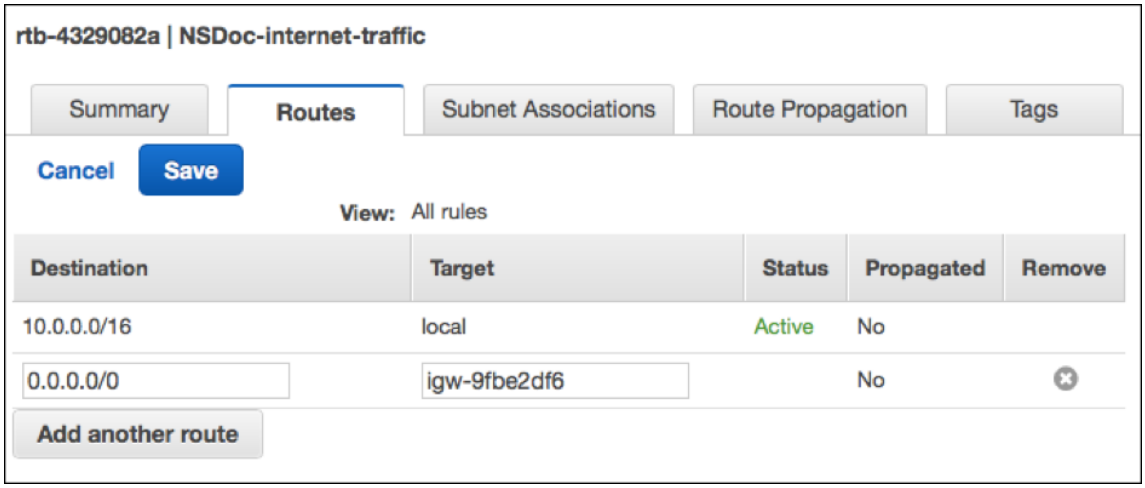


   The route table is assigned to all the subnets that you created for this VPC, so that routing of traffic from an instance in one subnet can reach an instance in another subnet.

5. Click Subnet Associations, and then click Edit.
6. Click the management and client subnet and click Save. This creates a route table for internet traffic only.



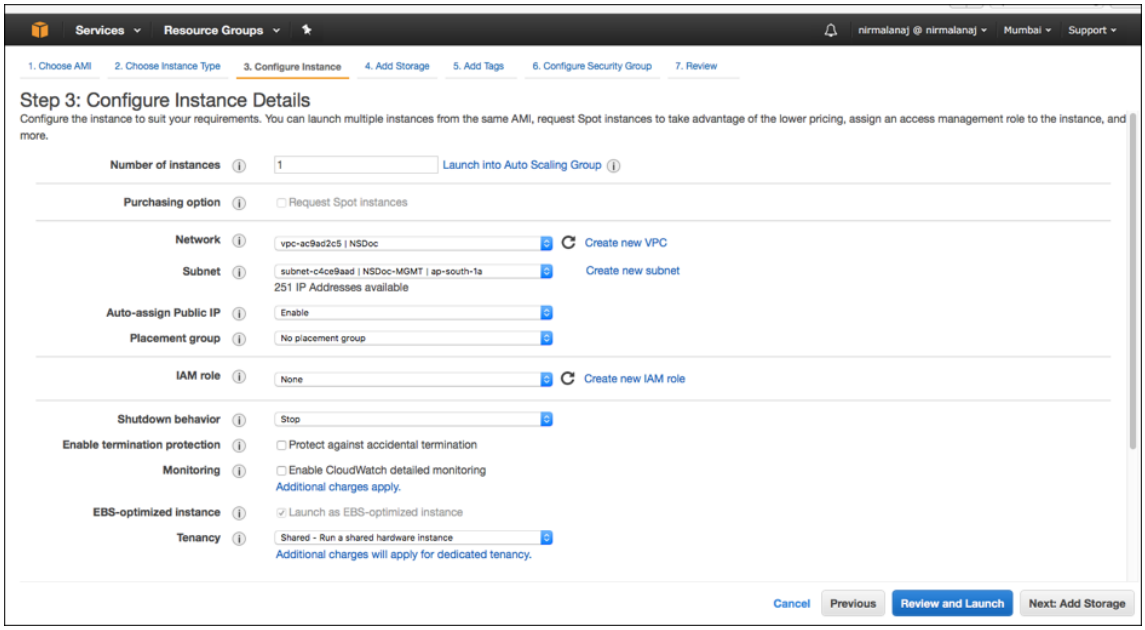
7. Click **Routes > Edit > Add another route**.
8. In the Destination field add 0.0.0.0/0, and click the Target field to select igw-<xxxx> the Internet Gateway that the VPC Wizard created automatically.
9. Click Save.



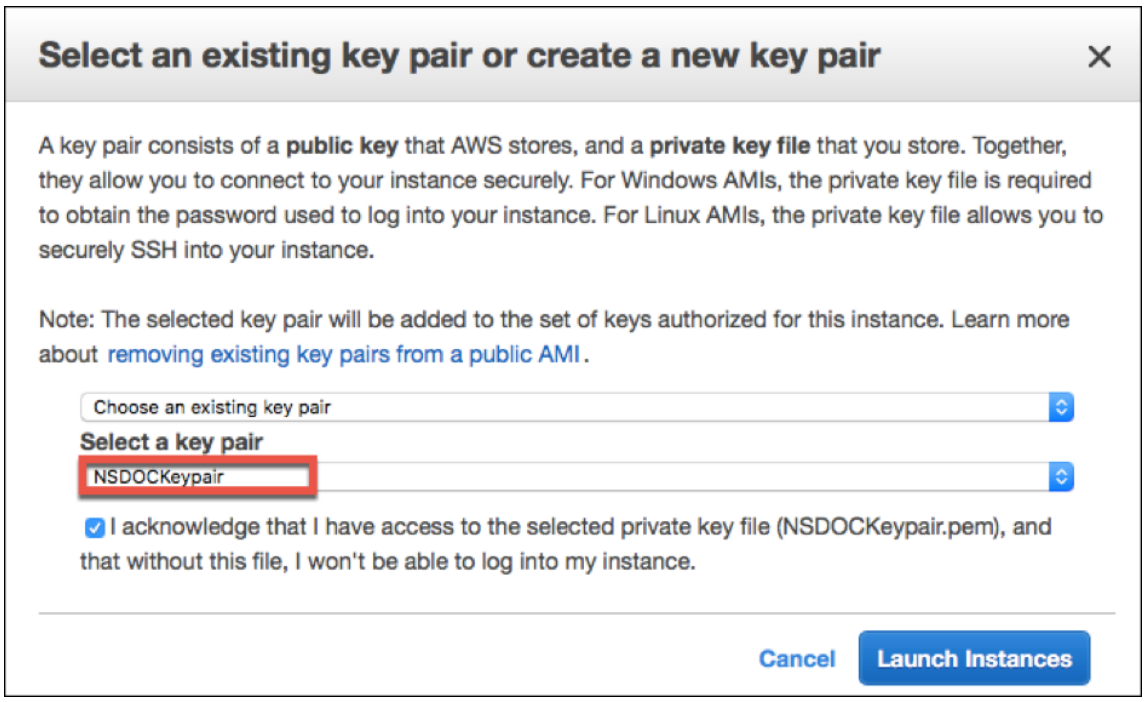
10. Follow the steps to create a route table for server-side traffic.

Step 4: Create a Citrix ADC VPX instance.

1. Log on the AWS management console and click **EC2** under **Compute**.
2. Click AWS Marketplace. In the Search AWS Marketplace bar, type Citrix ADC VPX and press Enter. The available Citrix ADC VPX editions are displayed.
3. Click **Select** to choose the desired Citrix ADC VPX edition. The EC2 instance wizard starts.
4. In the **Choose Instance Type** page, select **m4. Xlarge** (recommended) and click **Next: Configure Instance Details**.
5. In the Configure Instance Details page, select the following, and then click Next: Add Storage.
   - Number of instances: 1
   - Network: the VPC that created in Step 1
   - Subnet: the management subnet
   - Auto-assign Public IP: Enable



6. In the Add Storage page, select the default option, and click Next: Add Tags.
7. In the Add Tags page, add a name for the instance, and click Next: Configure Security Group.
8. In the Configure Security Group page, select the default option (which is generated by AWS Marketplace and is based on recommended settings by Citrix Systems) and then click **Review and Launch > Launch**.
9. You are prompted to select an existing key pair or create and new key pair. From the Select a key pair drop-down list, select the key pair that you created as a prerequisite (See the Prerequisite section.)
10. Check the box to acknowledge the key pair and click Launch Instances.



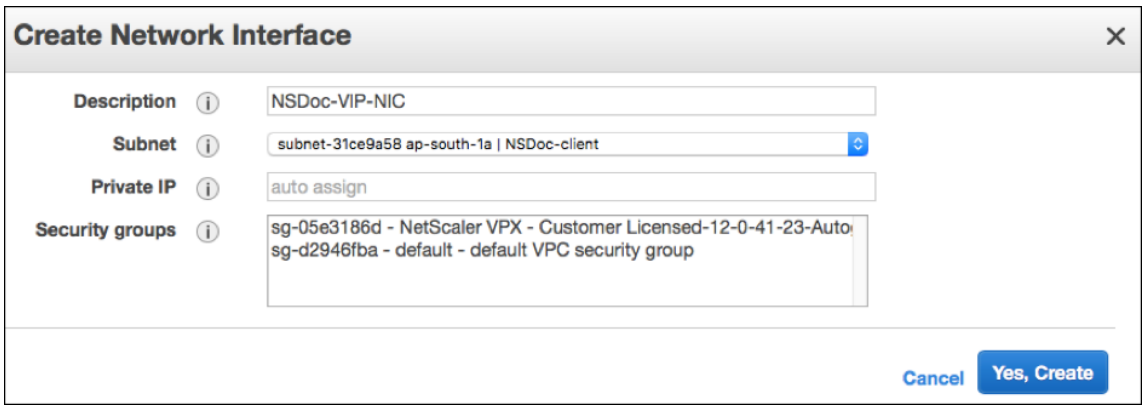
Launch Instance Wizard displays the Launch Status, and the instance appears in the list of instances when it is fully launched.

The check instance, go the AWS console click EC2 > Running Instances. Select the instance and add a name. Make sure the Instance State is running and Status Checks is complete.

Step 5: Create and attach more network interfaces.

When you created the VPC, only one network interface associated with it. Now add two more network interfaces to the VPC, for the VIP and SNIP.

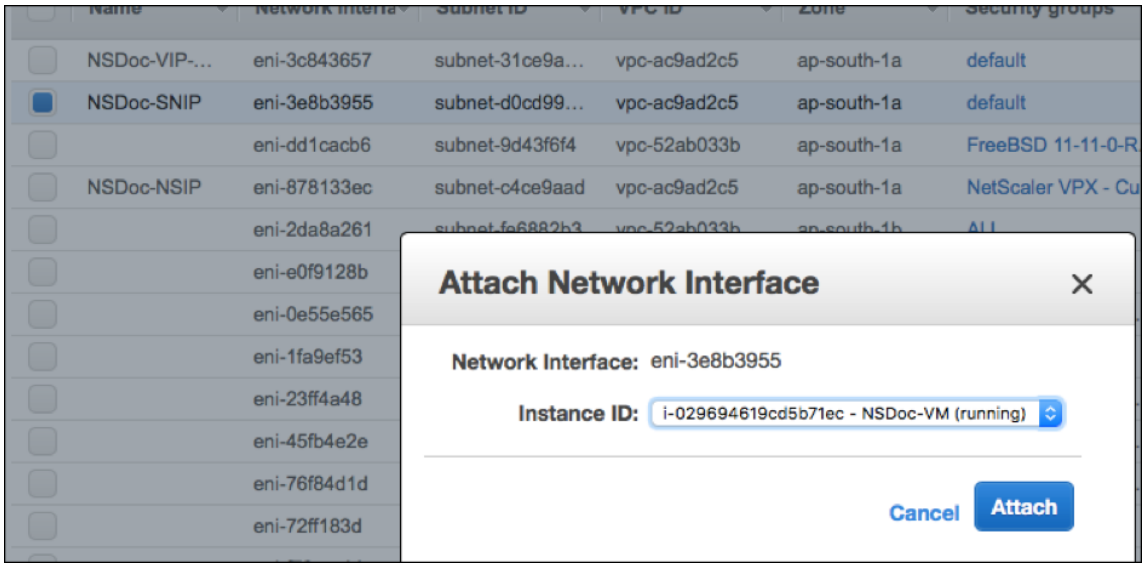
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Choose Create Network Interface.
4. For Description, enter a descriptive name.
5. For Subnet, select the subnet that you created previously for the VIP.
6. For Private IP, leave the default option.
7. For Security groups, select the group.
8. Click **Yes, Create**.



9. After the network interface is created, add a name to the interface.
10. Repeat the steps to create a network interface for server-side traffic.

Attach the network interfaces:

1. In the navigation pane, choose Network Interfaces.
2. Select the network interface and choose Attach.
3. In the Attach Network Interface dialog box, select the instance and choose Attach.



Step 6: Attach an elastic IP to the NSIP.

1. From the AWS management console, go to **NETWORK & SECURITY > Elastic IPs**.
2. Check for available free EIP to attach. If none, click **Allocate new address**.
3. Select the newly allocated IP address and choose **Actions > Associate address**.
4. Click the **Network interface** radio button.
5. From the Network interface drop-down list, select the management NIC.

6. From the **Private IP** drop-down menu, select the AWS-generated IP address.
7. Select the **Reassociation** check box.
8. Click **Associate**.

Access the VPX instance:

After you've configured a standalone Citrix ADC VPX instance with three NICs, log on to the VPX instance to complete the Citrix ADC-side configuration. Use of the following options:

- GUI: Type the public IP of the management NIC in the browser. Log on by using `nsroot` as the user name and the instance ID (i-0c1ffe1d987817522) as the password.

> Note
>
> On your first logon, you are prompted to change the password for security reasons. After changing the password, you must save the configuration. If the configuration is not saved and the instance restarts, you must log on with the default password. Change the password again at the prompt and save the configuration.

- SSH: Open an SSH client and type:

```
ssh -i \<location of your private key\> ns root@\<public DNS of the instance \>
```

To find the public DNS, click the instance, and click **Connect**.

Related information:

- To configure the Citrix ADC-owned IP addresses (NSIP, VIP, and SNIP), see Configuring Citrix ADC-Owned IP Addresses.
- You've configured a BYOL version of the Citrix ADC VPX appliance, for more information see the VPX Licensing Guide at http://support.citrix.com/article/CTX122426

Download a Citrix ADC VPX license

September 14, 2021

After the launch of Citrix ADC VPX-customer licensed instance from the AWS marketplace, a license is required. For more information on VPX licensing, see Licensing overview.

You have to:

1. Use the licensing portal within the Citrix website to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance activate automatically.

If you use a Citrix ADC VPX instance with a model number higher than VPX 5000, the network throughput might not be the same as specified by the instance's license. However, other features, such as SSL throughput and SSL transactions per second, might improve.

5 Gbps network bandwidth is observed in the `c4.8xlarge` instance type.

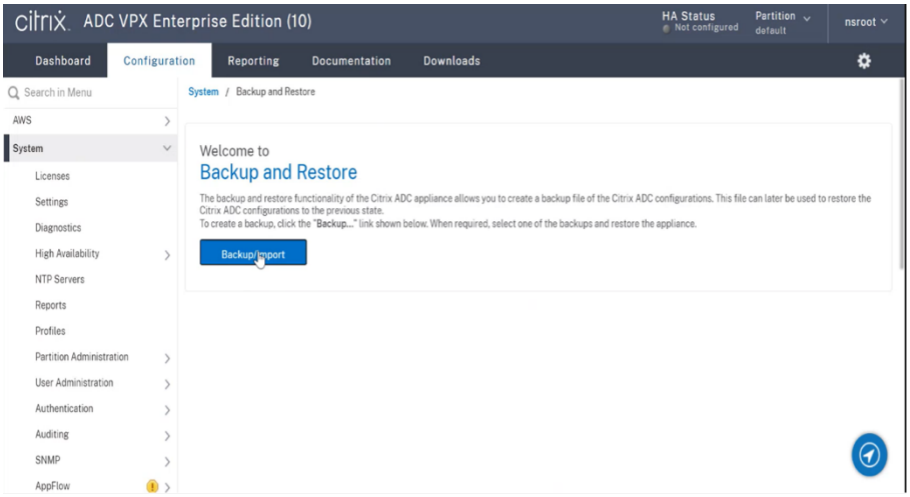
How to migrate the AWS subscription to BYOL

This section describes the procedure to migrate from AWS subscription to Bring your own license (BYOL), and conversely.

Do the following steps to migrate an AWS subscription to BYOL:

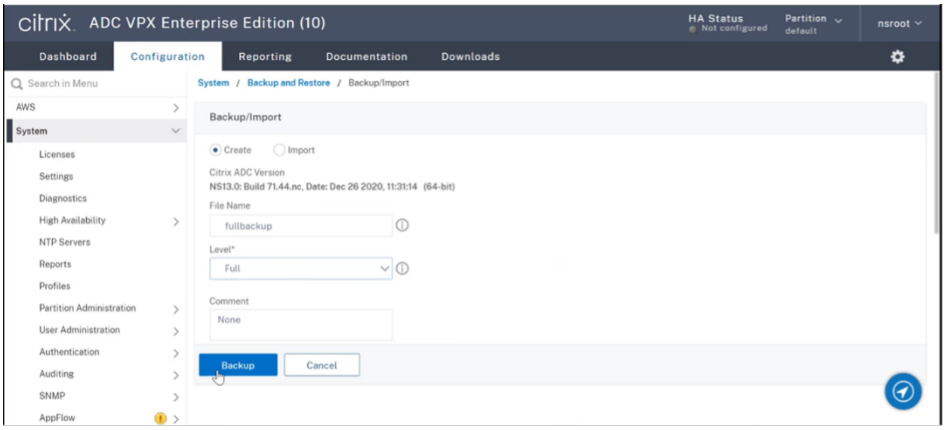
> Note
>
> The **Step 2** and **Step 3** are done on the Citrix ADC VPX instance, and all other steps are done on the AWS portal.

1. Create a BYOL EC2 instance using Citrix ADC VPX - Customer Licensed in the same availability zone as the old EC2 instance that has the same security group, IAM role, and subnet. The new EC2 instance must have only one ENI interface.
2. To back up the data on the old EC2 instance using the Citrix ADC GUI, follow these steps.
    a) Navigate to **System > Backup and Restore**.
    b) In the **Welcome** page, click **Backup/Import** to start the process.

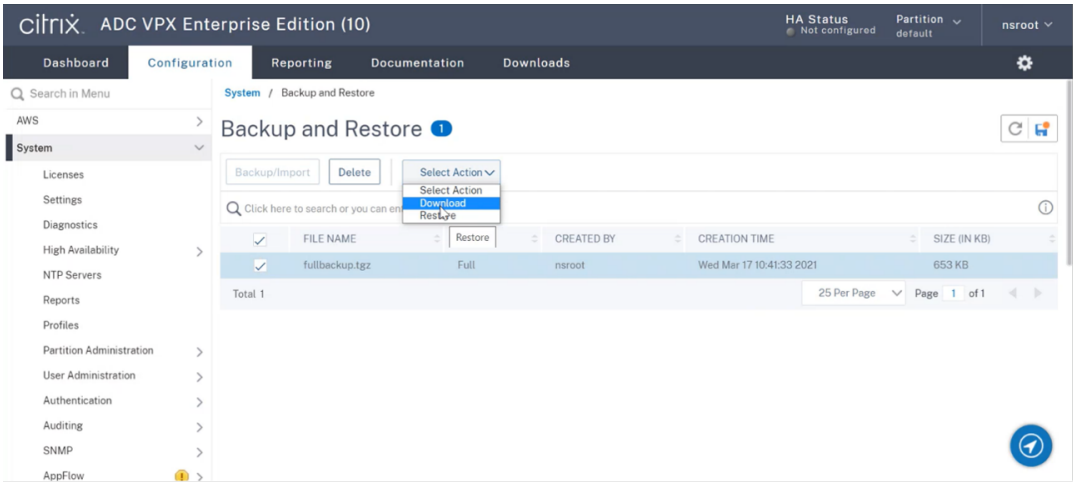


c) In the **Backup/Import** page, fill in the following details:

- **Name** – Name of the backup file.
- **Level** – Select the backup level as **Full**.
- **Comment** – Provide a brief description of the backup.



d) Click **Backup**. Once the backup is complete, you can select the file and download it to your local machine.

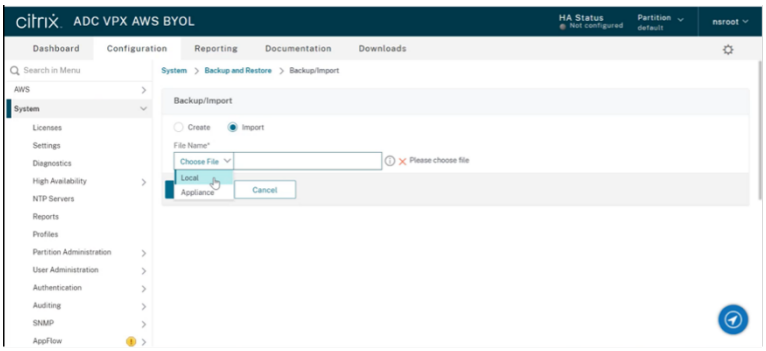


3. To restore the data on the new EC2 instance using the Citrix ADC GUI, follow these steps:

   a) Navigate to **System > Backup and Restore**.

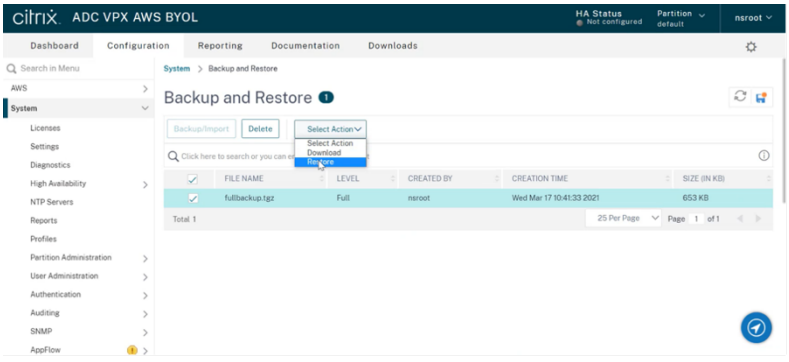
   b) Click **Backup/Import** to start the process.

   c) Select the **Import** option and upload the backup file.

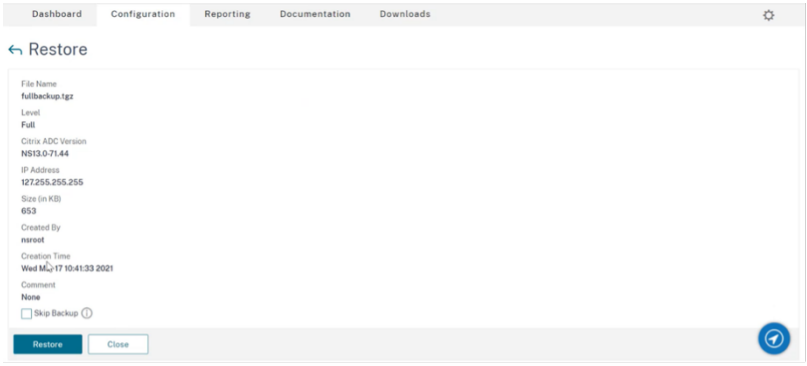


   d) Select the file.

   e) From the **Select Action** drop-down menu, select **Restore**.



   f) On the **Restore** page, verify the file details, and click **Restore**.



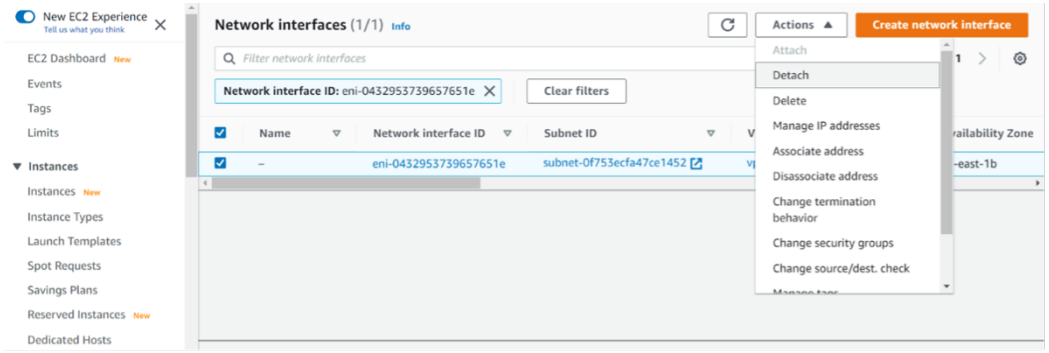
g) After the restore, reboot the EC2 instance.

4. Move all interfaces (except the management interface to which the NSIP address is bound) from the old EC2 instance to the new EC2 instance. To move a network interface from one EC2 instance to another, follow these steps:

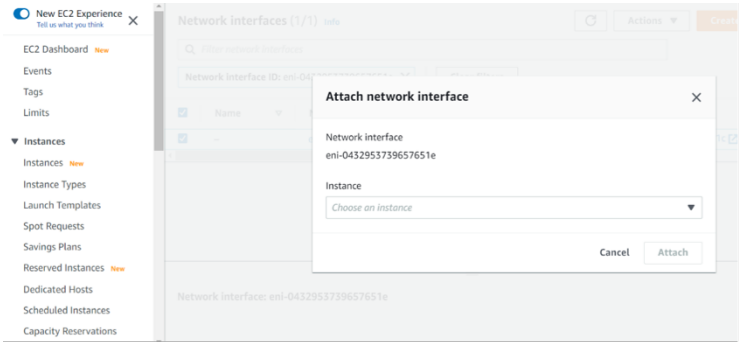
   a) In the **AWS Portal**, stop both the old and new EC2 instances.

   b) Navigate to **Network Interfaces**, and select the network interface attached to the old EC2 instance.

   c) Detach the EC2 instance by clicking **Actions > Detach**.



   d) Attach the network interface to the new EC2 instance by clicking **Actions > Attach**. Enter the EC2 instance name to which the network interface must be attached.



   e) Do the **Step 1** to **Step 4** for all other interfaces that are attached. Make sure to follow the sequence and maintain the interface order. That is, first detach interface 2 and attach it,

and then detach interface 3 and attach it, and so on.

5. You can't detach the management interface from an old EC2 instance. So, move all the secondary IP addresses (if any) on the management interface (primary network interface) of the old EC2 instance to the new EC2 instance. To move an IP address from one interface to another, follow these steps:

   a) In the **AWS Portal**, make sure that both the old and new EC2 instances are in **Stop** state.

   b) Navigate to **Network Interfaces**, and select the management network interface attached to the old EC2 instance.

   c) Click **Actions > Manage IP Address**, and make note of all the secondary IP addresses assigned (if any).

   d) Navigate to the management network interface or primary interface of the new EC2 instance.

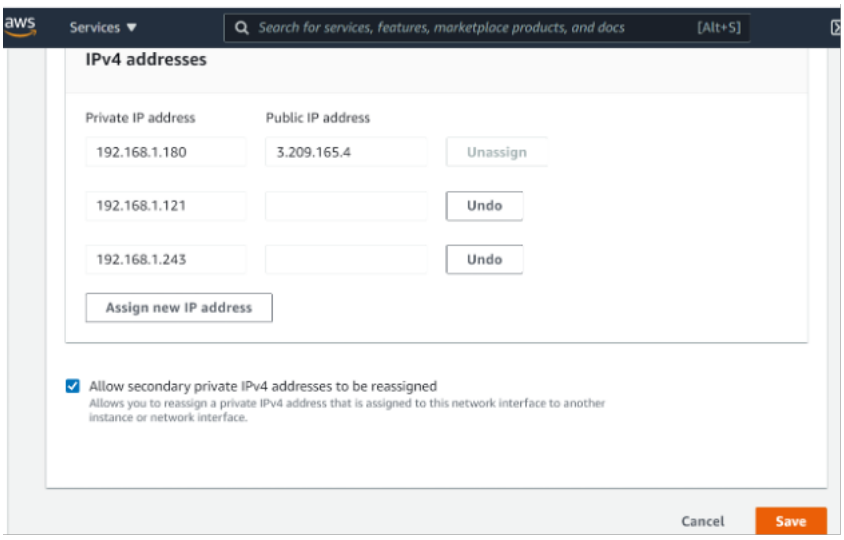
   e) Click **Actions > Manage IP Addresses**.

   f) Under **IPv4 Addresses**, click **Assign new IP address**.

   g) Enter the IP addresses, which are noted in the **Step 3**.

   h) Select **Allow secondary private IP addresses to be reassigned** check box.

   i) Click **Save**.



6. Start the new EC2 instance and verify the configuration. After all the configuration is moved, you can delete or keep the old EC2 instance as per your requirement.
7. If any EIP address is attached to the NSIP address of the old EC2 instance, move the old instance NSIP address to the new instance NSIP address.
8. If you want to revert to the old instance, then follow the same steps in the opposite way between the old and new instance.

9. After you move from subscription instance to BYOL instance, a license is required. To install a license follow these steps:
    - Use the licensing portal in the Citrix website to generate a valid license.
    - Upload the license to the instance. For more information, see VPX ADC - Install a new license.

> Note
>
> When you move BYOL instance to subscription instance (paid marketplace instance), you need not install the license. The correct feature set and performance is automatically activated.

Limitations

The management interface can't be moved to the new EC2 instance. So Citrix recommends you manually configure the management interface. For more information, see **Step 5** in the preceding procedure. A new EC2 instance is created with the exact replica of the old EC2 instance but only the NSIP address has a new IP address.

Load balancing servers in different availability zones

September 14, 2021

A VPX instance can be used to load balance servers running in the same availability zone, or in:

- A different availability zone (AZ) in the same AWS VPC
- A different AWS region
- AWS EC2 in a VPC

To enable a VPX instance to load balance servers running outside the AWS VPC that the VPX instance is in, configure the instance to use EIPs to route traffic through the Internet gateway, as follows:

1. Configure a SNIP on the Citrix ADC VPX instance by using the Citrix ADC CLI or the GUI.
2. Enable traffic to be routed out of the AZ, by creating a public facing subnet for the server-side traffic.
3. Add an Internet gateway route to the routing table, using the AWS GUI console.
4. Associate the routing table you updated with the server-side subnet.
5. Associate an EIP with the server-side private IP address that is mapped to a Citrix ADC SNIP address.

How high availability on AWS works

September 14, 2021

You can configure two Citrix ADC VPX instances on AWS as a high availability (HA) active-passive pair. When you configure one instance as the primary node and the other as the secondary node, the primary node accepts connections and manages servers. The secondary node monitors the primary. If for any reason, the primary node is unable to accept connections, the secondary node takes over.

In AWS, the following deployment types are supported for VPX instances:

- High availability within same zone
- High availability across different zones

> **Note**
>
> For high availability to work, ensure both the Citrix ADC VPX instances are attached with IAM roles and assigned with the Elastic IP (EIP) address to the NSIP. You need not assign an EIP on NSIP if the NSIP can reach internet through the NAT instance.

High availability within the same zones

In a high-availability deployment within the same zones, both VPX instances must have similar networking configurations.

Follow these two rules:

Rule 1. Any NIC on one VPX instance must be in the same subnet as the corresponding NIC in the other VPX. Both instances must have:

- Management interface on the same subnet (referred as management subnet)
- Client interface on the same subnet (referred as client subnet)
- Server interface on the same subnet (referred as server subnet)

Rule 2. Sequence of mgmt NIC, client NIC, and server NIC on both instances must be the same.
For example, the following scenario is not supported.

VPX instance 1

NIC 0: management
NIC 1: client
NIC 2: Server

VPX instance 2

NIC 0: management

NIC 1: server

NIC 2: client

In this scenario, NIC 1 of instance 1 is in client subnet while NIC 1 of instance 2 is in server subnet. For HA to work, NIC 1 of both the instances must be either in the client subnet or in the server subnet.

From 13.0 41.xx, high availability can be achieved by migrating secondary private IP addresses attached to the NICs (client and server-side NICs) of the primary HA node to the secondary HA node after failover. In this deployment:

- Both the VPX instances have the same number of NICs and subnet mapping according to NIC enumeration.
- Each VPX NIC has one extra private IP address, except the first NIC - which corresponds to the management IP address. The extra private IP address appears as the primary private IP address in the AWS web console. In our document, we refer to this extra IP address as the dummy IP address).
- The dummy IP addresses must be not configured on the Citrix ADC instance as VIP and SNIP.
- Other secondary private IP addresses must be created, as required, and configured as VIP and SNIP.
- On failover, the new primary node looks for configured SNIPs and VIPs and moves them from NICs attached to the previous primary to corresponding NICs on the new primary.
- Citrix ADC instances require IAM permissions for HA to work. Add the following IAM privileges to the IAM policy added to each instance.

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeNetworkInterfaces"
"ec2:AssignPrivateIpAddresses"
```

> Note: `unassignPrivateIpAddress` is not required.

This method is faster than the legacy method. In the older method, HA depends on the migration of AWS elastic network interfaces of the primary node to the secondary node.

For a legacy method, the following policies are required:

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

For more information, see Deploy a high availability pair on AWS.

High availability across different zones

You can configure two Citrix ADC VPX instances on two different subnets or two different AWS availability zones, as a high availability active-passive pair in Independent Network Configuration (INC) mode. Upon failover, the EIP (Elastic IP) of the VIP of the primary instance migrates to the secondary, which takes over as the new primary. In the failover process, the AWS API:

- Checks the virtual servers that have `IPSets` attached to them.
- Finds the IP address that has an associated public IP, from the two IP addresses the virtual server is listening on. One that is directly attached to the virtual server, and one that is attached through the IP set.
- Reassociates the public IP (EIP) to the private IP belonging to the new primary VIP.

For HA across different zones, the following policies are required:

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

For more information, see High availability across AWS availability zones.

Before you start your deployment

Before you start any HA deployment on AWS, read the following document:

- Prerequisites
- Limitations and usage guidelines
- Deploy a Citrix ADC VPX instance on AWS
- High Availability

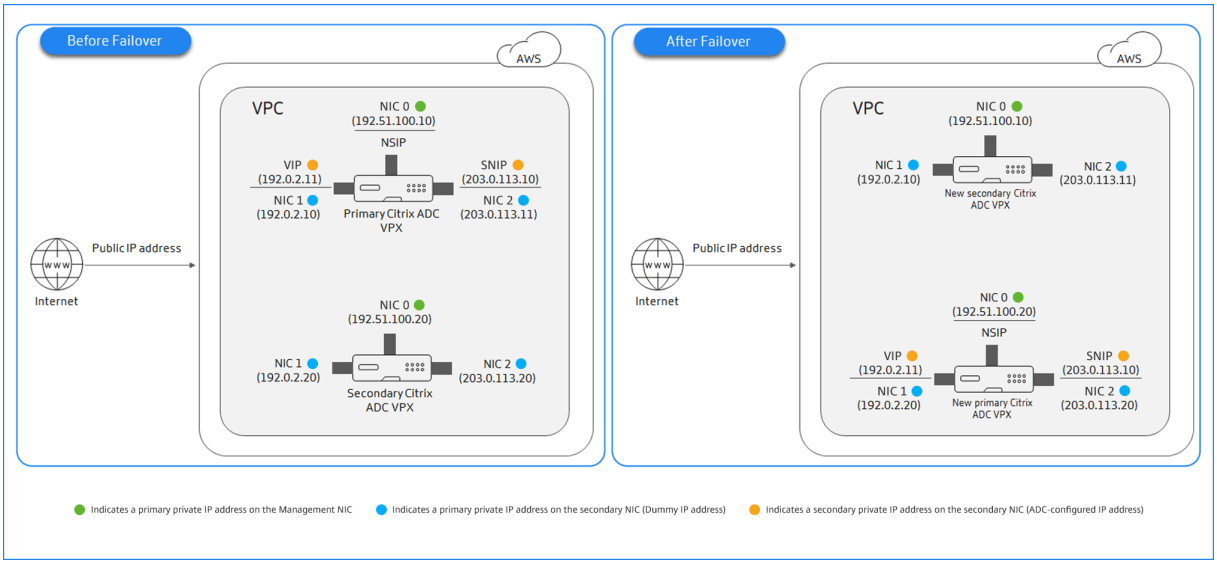
Deploy a high availability pair on AWS

October 27, 2021

You can configure two Citrix ADC VPX instances on AWS as a high-availability (HA) pair, in the same AWS zone where both VPX instances are on the same subnet. HA is achieved by migrating secondary private IP addresses attached to the NICs (client and server-side NICs) of the primary HA node to the secondary HA node after failover. All the Elastic IP addresses associated with the secondary private IP addresses are also migrated.

The following illustration depicts an HA failover scenario by migrating secondary private IP addresses.

Figure 1. A Citrix ADC VPX HA Pair on AWS, using private IP migration



Before you start your document, read the following docs:

- Prerequisites
- Limitations and usage guidelines
- Deploy a Citrix ADC VPX instance on AWS
- High Availability

How to deploy a VPX HA pair in the same zone

Here is the summary of the steps to deploy a VPX HA pair in the same zone:

1. Create two VPX instances on AWS, each with three NICs
2. Assign AWS secondary private IP address to VIP and SNIP of primary node
3. Configure VIP and SNIP on primary node using AWS secondary private IP addresses
4. Configure HA on both nodes

Step 1. Create two VPX instances (primary and secondary nodes) by using the same VPC, each with three NICs (Ethernet 0, Ethernet 1, Ethernet 2)

Follow the steps given in Deploy a Citrix ADC VPX instance on AWS by using the AWS web console.

Step 2. On the primary node, assign secondary private IP addresses for Ethernet 1 (client IP or VIP) and Ethernet 2 (back-end server IP or SNIP)

The AWS console automatically assigns primary private IP addresses to the configured NICs. Assign more private IP addresses to VIP and SNIP, known as secondary private IP addresses.

To assign a secondary private IPv4 address to a network interface, follow these steps:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces, and then select the network interface attached to the instance.
3. Choose Actions, Manage IP Addresses.
4. Under IPv4 Addresses, choose Assign new IP.
5. Enter a specific IPv4 address that's within the subnet range for the instance, or leave the field blank to let Amazon select an IP address for you.
6. (Optional) Choose Allow reassignment to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
7. Choose Yes, Update.

Under the instance description, the assigned secondary private IP addresses appear.

Step 3. Configure VIP and SNIP on the primary node, using secondary private IP addresses

Access the primary node using SSH. Open an ssh client and type:

```
ssh -i <location of your private key> nsroot@<public DNS of the
   instance>
<!--NeedCopy-->
```

Next, configure VIP and SNIP.

For VIP, type:

```
add ns ip <IPAddress> <netmask> -type <type>
<!--NeedCopy-->
```

For SNIP, type:

```
add ns ip <IPAddress> <netmask> -type SNIP
<!--NeedCopy-->
```

Type `save config` to save.

To see the configured IP addresses, type the following command:

```
show ns ip
<!--NeedCopy-->
```

For more information, see the following topics:

- Configuring and Managing Virtual IP (VIP) Addresses
- Configuring the NSIP address

Step 4: Configure HA on both instances

On the primary node, open a Shell client and type the following command:

```
add ha node <id> <private IP address of the management NIC of the
   secondary node>
<!--NeedCopy-->
```

On the secondary node, type the following command:

```
add ha node <id> < private IP address of the management NIC of the
   primary node >
<!--NeedCopy-->
```

Type `save config` to save the configuration.

To see the configured HA nodes, type `show ha node`.

Upon failover, the secondary private IP addresses configured as VIP and SNIP on the previous primary node are migrated to the new primary node.

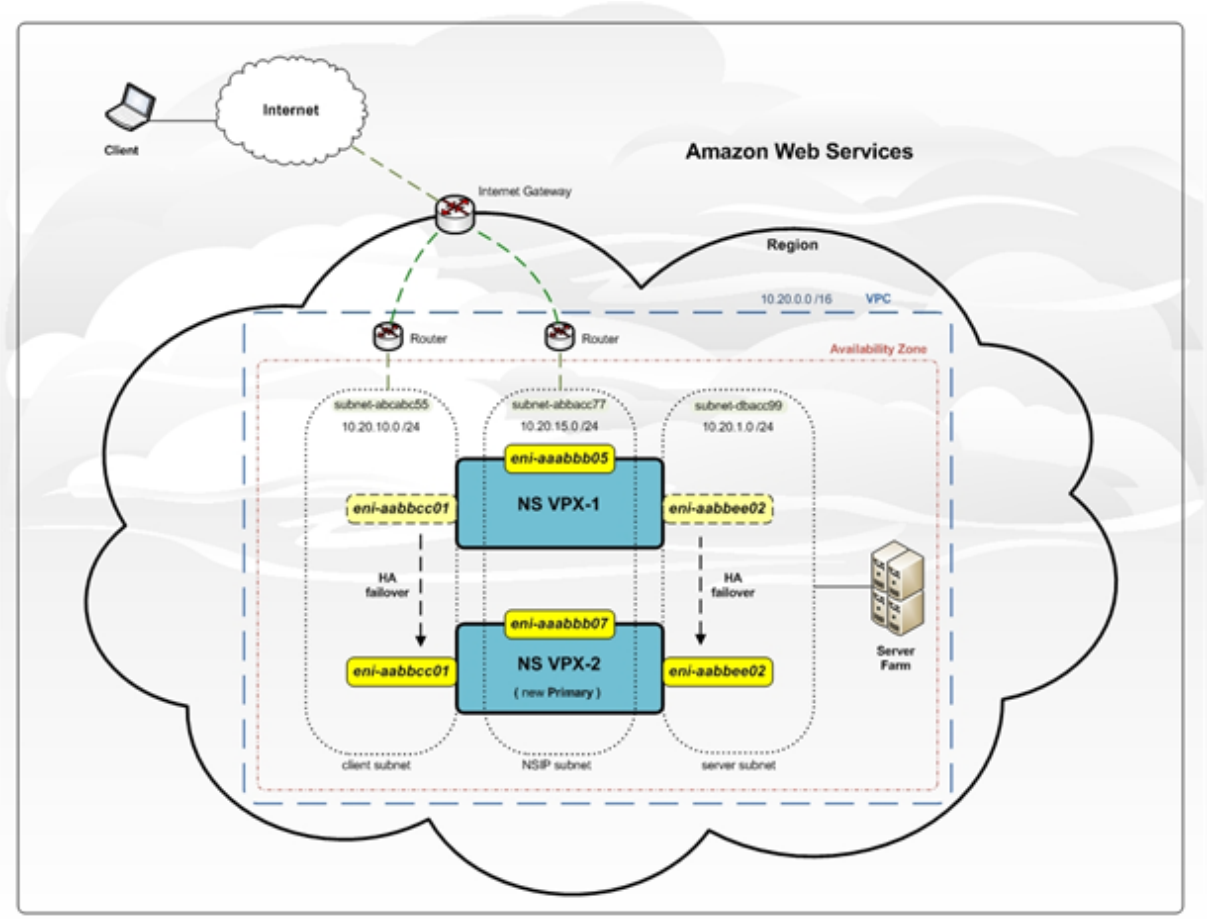
To force a failover on a node, type `force HA`failover.

Legacy method for deploying a VPX HA pair

Before 13.0 41.x release, HA within the same zone was achieved through AWS elastic network interface (ENI) migration. However, this method is slowly deprecated.

The following figure shows an example of the HA deployment architecture for Citrix ADC VPX instances on AWS.

Figure 1. A Citrix ADC VPX HA Pair on AWS, using ENI migration



You can deploy two VPX instances on AWS as an HA pair by using one of the following options:

- Create the instances with IAM Role manually by using the AWS Management Console and then configure HA on them.
- Or automate the high availability deployment by using the Citrix CloudFormation template.

The CloudFormation template significantly decreases the number of steps involved for creating an HA pair, and it automatically creates an IAM Role. This section shows how to deploy a Citrix ADC VPX HA (active-passive) pair by using the Citrix CloudFormation template.

Keep the following points in mind while deploying two Citrix ADC VPX instances as an HA pair.

Points to note

- HA on AWS requires the primary node to have at least two ENIs (one for management and the other for data traffic), and the secondary node to have one management ENI. However, for security purposes, create three ENIs on the primary node, because this setup allows you to segregate the private and public network (recommended).
- The secondary node always has one ENI interface (for management) and the primary node can have up to four ENIs.

- The NSIP addresses for each VPX instance in a high availability pair must be configured on the default ENI of the instance.
- Amazon does not allow any broadcast/multicast packets in AWS. As a result, in a HA setup, data-plane ENIs are migrated from the primary to the secondary VPX instance when the primary VPX instance fails.
- Because the default (management) ENI cannot be moved to another VPX instance, do not use the default ENI for client and server traffic (data-plane traffic).
- The message AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF success output 0 in the /var/log/ns.log indicates that the two data ENIs have successfully attached to the secondary instance (the new primary).
- Failover might take up to 20 seconds due to the AWS detach/attach ENI mechanism.
- Upon failover, the failed instance always restarts.
- The heartbeat packets are received only on the management interface.
- The configuration file of the primary and secondary VPX instances is synchronized, including the `nsroot` password. The `nsroot` password of the secondary node is set to that of the primary node after the HA configuration synchronization.
- To have access to the AWS API servers, either the VPX instance must have a public IP address assigned or routing must be set up correctly at VPC subnet level pointing to the internet gateway of the VPC.
- Nameservers/DNS servers are configured at VPC level using DHCP options.
- The Citrix CloudFormation template does not create an HA setup between different availability zones.
- The Citrix CloudFormation template does not create an INC mode.
- The AWS debug messages are available in the log file, /var/log/ns.log, on the VPX instance.

Deploy a high availability pair by using the Citrix CloudFormation template

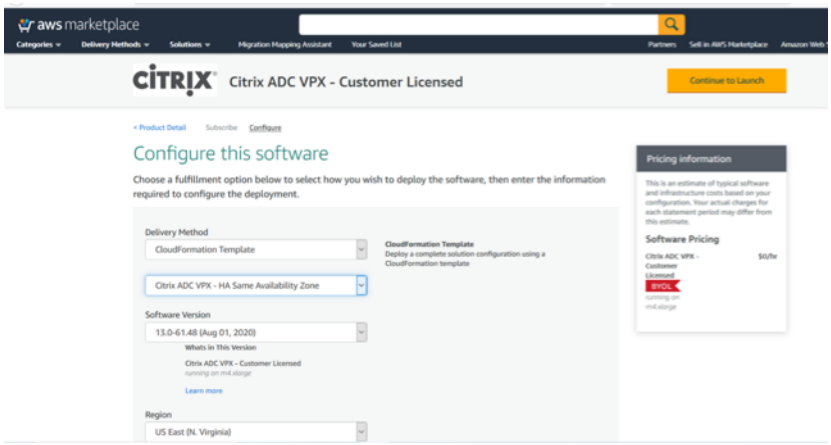
Before starting the CloudFormation template, ensure that you complete the following requirements:

- A VPC
- Three subnets within the VPC
- A security group with UDP 3003, TCP 3009–3010, HTTP, SSH ports open
- A key pair
- Create an internet gateway
- Edit route tables for client and management networks to point to the internet gateway

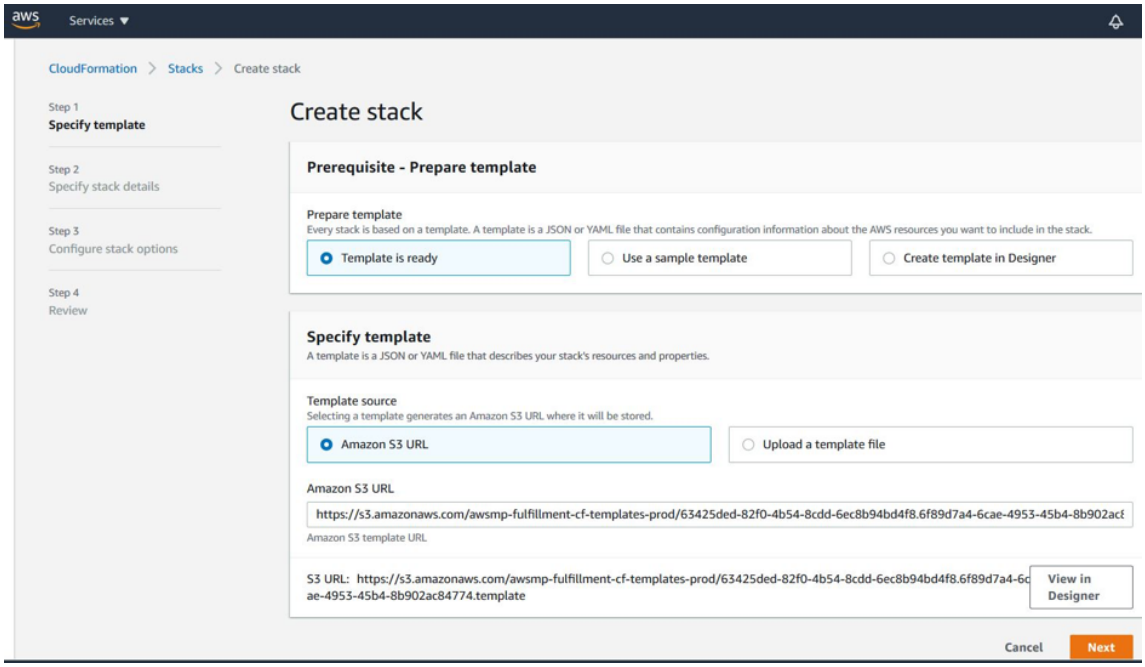
> **Note**
>
> The Citrix CloudFormation template automatically creates an IAM Role. Existing IAM Roles do not appear in the template.

To launch the Citrix CloudFormation template:

1. Log on to the AWS marketplace by using your AWS credentials.
2. In the search field, type **Citrix ADC VPX** to search for the Citrix ADC AMI, and click **Go**.
3. On the search result page, click the desired Citrix ADC VPX offering.
4. Click the **Pricing** tab, to go to **Pricing Information**.
5. Select the region and **Fulfillment Option** as **Citrix ADC VPX – Customer Licensed**.
6. Click **Continue to Subscribe**.
7. Check the details in the **Subscribe** page and click **Continue to Configuration**.
8. Select **Delivery Method** as **CloudFormation Template**.
9. Select the required CloudFormation template.
10. Select **Software Version** and **Region**, and click **Continue to Launch**.

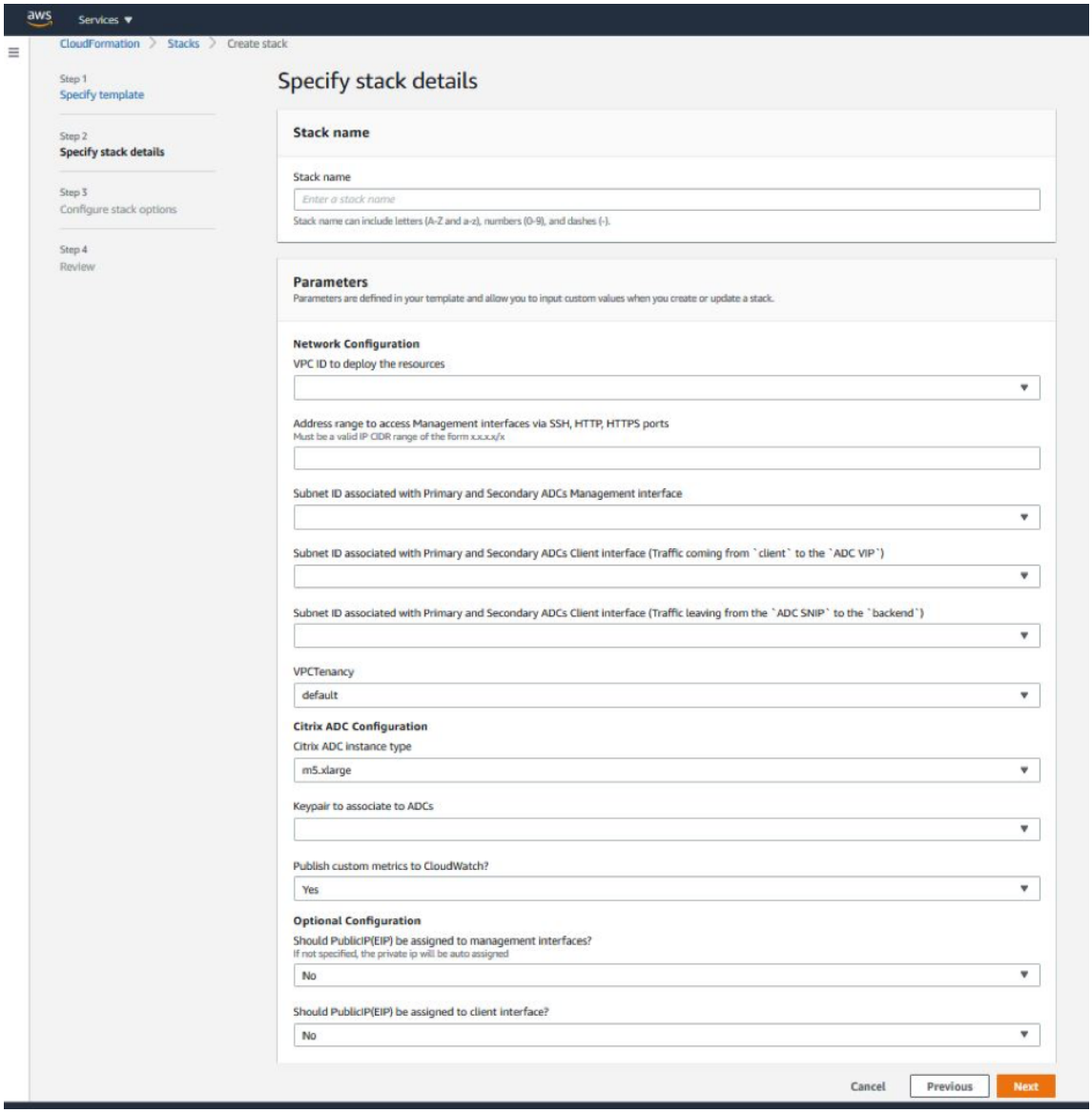


11. Under **Choose Action**, select **Launch CloudFormation**, and click **Launch**.
12. The **Create stack** page appears, and click **Next**.

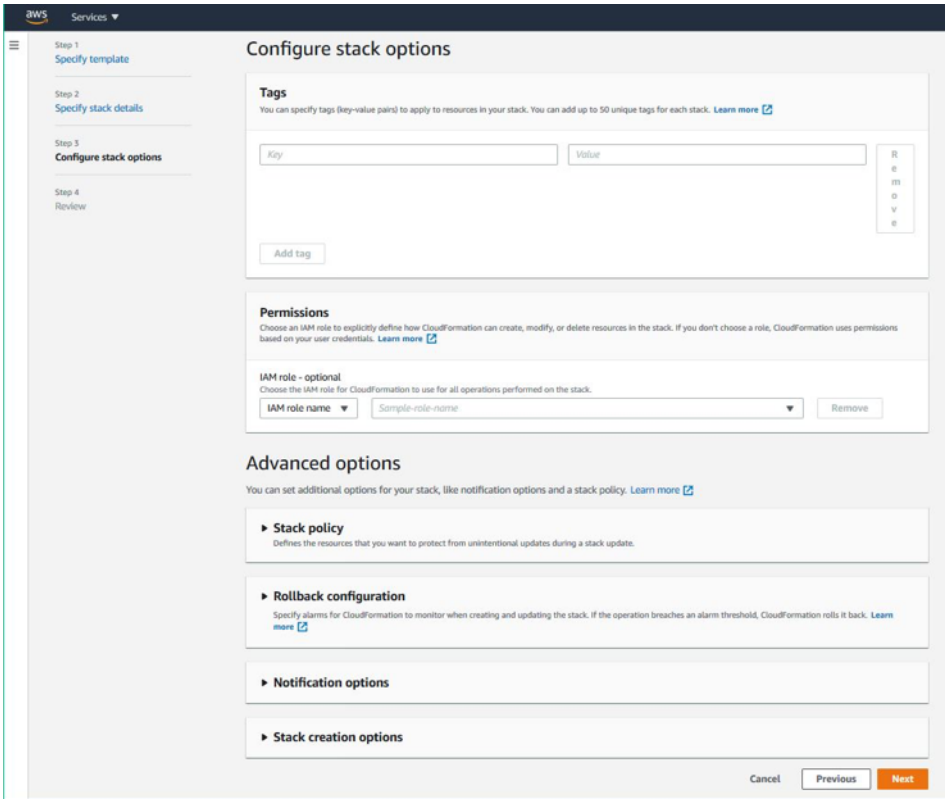


13. The **Specify stack details** page appears. Enter the following details.

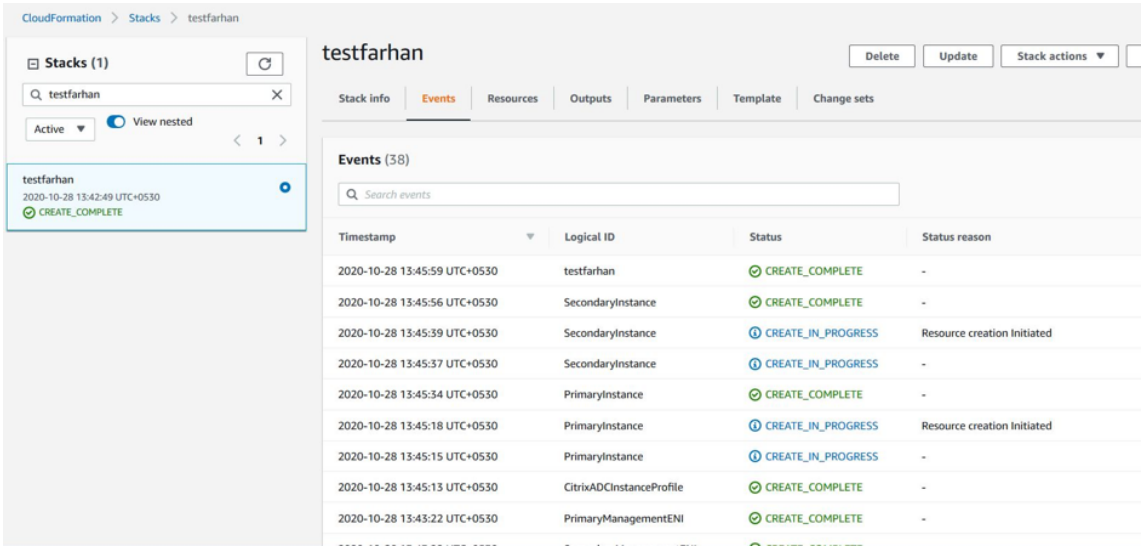
    - Type a **Stack name**. The name must be within 25 characters.
    - Under **Network Configuration**, perform the following:
        - Select **Management Subnetwork**, **Client Subnetwork**, and **Server Subnetwork**. Ensure that you select the correct subnetworks you created within the VPC that you selected under VPC ID.
        - Add **Primary Management IP**, **Secondary Management IP**, **Client IP**, and **Server IP**. The IP addresses must belong to the same subnets of the respective subnetworks. Alternatively, you can let the template assign the IP addresses automatically.
        - Select **default** for **VPCTenancy**.
    - Under **Citrix ADC Configuration**, perform the following:
        - Select **m5.xlarge** for **Instance type**.
        - Select the key pair that you've already created from the menu for **Key Pair**.
        - By default, the **Publish custom metrics to CloudWatch?** option is set to **Yes**. If you want to disable this option, select **No**.
          For more information about CloudWatch metrics, see Monitor your instances using Amazon CloudWatch.
    - Under **Optional Configuration**, perform the following:
        - By default, the **Should publicIP(EIP) be assigned to management interfaces?** option is set to **No**.
        - By default, the **Should publicIP(EIP) be assigned to client interface?** option is set to **No**.



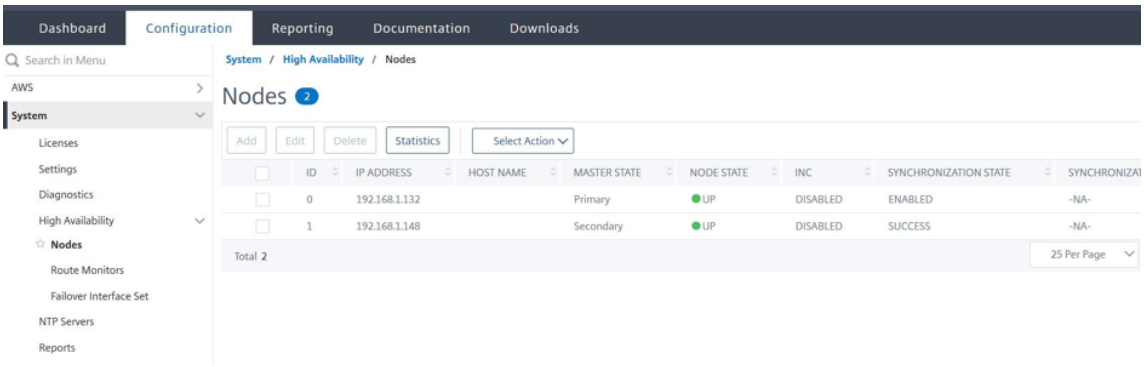
14. Click **Next**.
15. The **Configure stack options** page appears. This is an optional page.



16. Click **Next**.
17. The **Options** page appears. (This is an optional page.). Click **Next**.
18. The **Review** page appears. Take a moment to review the settings and make any changes, if necessary.
19. Select the **I acknowledge that AWS CloudFormation might create IAM resources.** check box, and then click **Create stack**.
20. The **CREATE-IN-PROGRESS** status appears. Wait until the status is **CREATE-COMPLETE**. If the status does not change to **COMPLETE**, check the **Events** tab for the reason of failure, and recreate the instance with proper configurations.



21. After an IAM resource is created, navigate to **EC2 Management Console > Instances**. You find two VPX instances created with IAM role. The primary and secondary nodes are created each with three private IP addresses and three network interfaces.
22. Log on to the primary node with user name `nsroot` and the instance ID as the password. From the GUI, navigate to **System > High Availability > Nodes**. The Citrix ADC VPX is already configured in HA pair by the CloudFormation template.
23. The Citrix ADC VPX HA pair appears.



Monitor your instances using Amazon CloudWatch

You can use the Amazon CloudWatch service to monitor a set of Citrix ADC VPX metrics such as CPU and memory utilization, and throughput. CloudWatch monitors resources and applications that run on AWS, in real time. You can access the Amazon CloudWatch dashboard by using the AWS Management console. For more information, see Amazon CloudWatch.

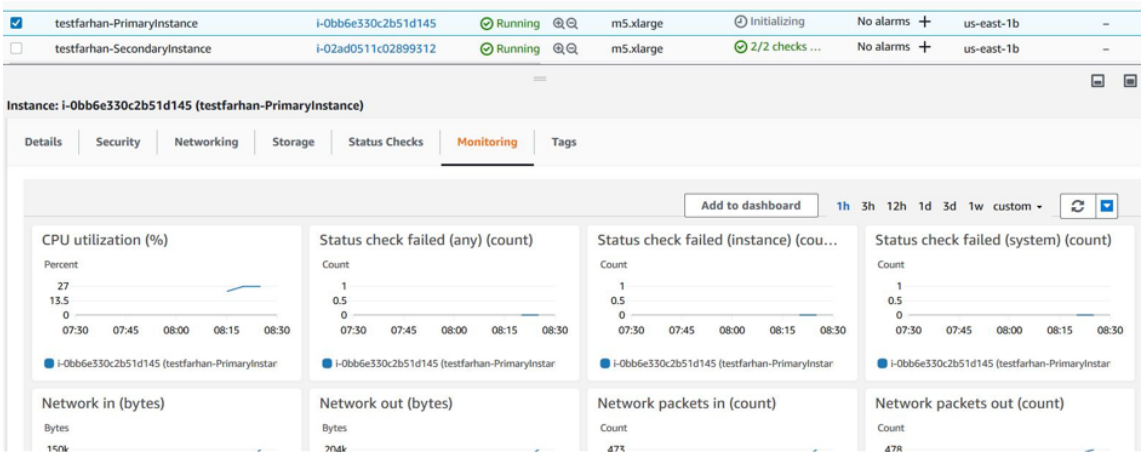
Points to note

- If you deploy a Citrix ADC VPX instance on AWS by using the AWS web console, the CloudWatch service is enabled by default.
- If you deploy a Citrix ADC VPX instance by using the Citrix CloudFormation template, the default option is “Yes.” If you want to disable the CloudWatch service, select “No.”
- Metrics are available for CPU (management and packet CPU usage), memory, and throughput (inbound and outbound).

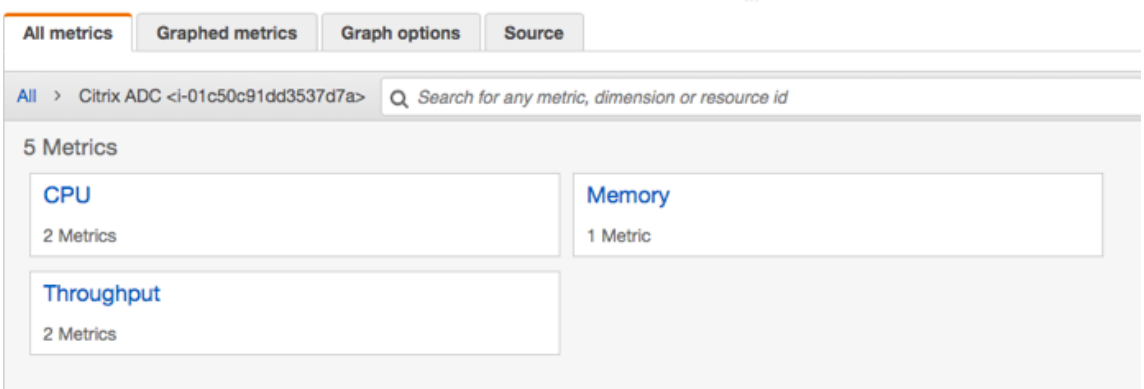
How to view CloudWatch metrics

To view CloudWatch metrics for your instance, follow these steps:

1. Log on to **AWS Management console > EC2 > Instances**.
2. Select the instance.
3. Click **Monitoring**.
4. Click **View all CloudWatch metrics**.



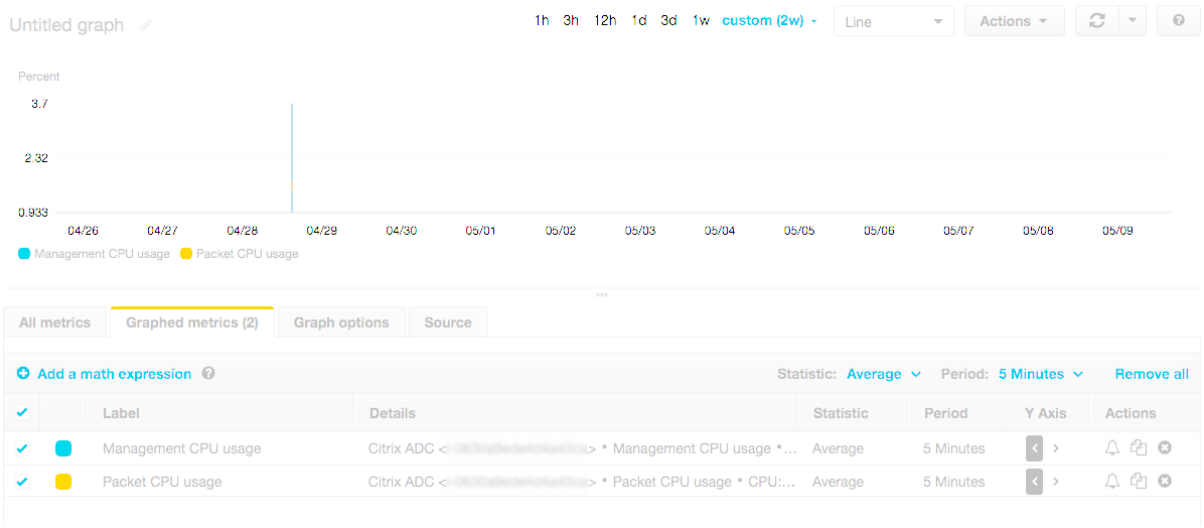
5. Under All metrics, click your instance ID.



6. Click the metrics you want to view, set the duration (by minutes, hours, days, weeks, months).

7. Click **Graphed metrics** to view the statistics of usage. Use the **Graph options** to customize your graph.

Figure. Graphed metrics for CPU usage



Configuring SR-IOV on a high availability setup

Support for SR-IOV interfaces in a high availability setup is available from Citrix ADC release 12.0 57.19 onwards. For more information about how to configure SR-IOV, see Configuring Citrix ADC VPX instances to Use SR-IOV Network Interface.

Related resources

How high availability on AWS works

High availability across AWS availability zones

September 14, 2021

You can configure two Citrix ADC VPX instances on two different subnets or two different AWS availability zones, as a high availability active-passive pair in Independent Network Configuration (INC) mode. If for any reason, the primary node is unable to accept connections, the secondary node takes over.

For more information about high availability, see High availability. For more information about INC, see Configuring high availability nodes in different subnets.

Points to note

- Read the following documents before you start your deployment:
  - AWS terminology
  - Prerequisites
  - Limitations and usage guidelines
- The VPX high availability pair can either reside in the same availability zone in a different subnet or in two different AWS availability zones.
- Citrix recommends that you use different subnets for management (NSIP), client traffic (VIP), and back-end server (SNIP).
- High availability must be set in Independent Network Configuration (INC) mode for a failover to work.
- The two instances must have port 3003 open for UDP traffic as that is used for heartbeats.
- The management subnets of both the nodes must have access to internet or to AWS API server through internal NAT so that the rest APIs are functional.
- IAM role must have E2 permission for the public IP or elastic IP (EIP) migration and EC2 Route Table permissions for the private IP migration.

You can deploy high availability across AWS availability zones in the following ways:

- Using elastic IP addresses
- Using private IP addresses

How high availability across AWS availability zones works

Upon failover, the EIP of the VIP of the primary instance migrates to the secondary, which takes over as the new primary. In the failover process, AWS API

1. Checks the virtual servers that have `IPSets` attached to them.
2. Finds the IP address that has an associated public IP, from the two IP addresses the virtual server is listening on. One that is directly attached to the virtual server, and one that is attached through the IP set.
3. Reassociates the public IP (EIP) to the private IP belonging to the new primary VIP.

> **Note**
>
> To protect your network from attacks such as denial-of-service (DoS), when using an EIP, you can create security groups in AWS to restrict the IP access. For high availability, you can switch from EIP to a private IP movement solution as per your deployments.

Deploy a VPX high-availability pair with elastic IP addresses across different AWS zones

September 14, 2021

You can configure two Citrix ADC VPX instances on two different subnets or two different AWS availability zones using elastic IP addresses in the INC mode.

For more information about high availability, see High availability. For more information about INC, see Configuring high availability nodes in different subnets.

How to deploy a VPX high-availability pair with elastic IP addresses across different AWS zones

The following is the summary of steps for deploying a VPX pair on two different subnets or two different AWS availability zones.

1. Create an Amazon virtual private cloud.
2. Deploy two VPX instances in two different availability zones or in the same zone but in different subnets.
3. Configure high availability
    a) Set up high availability in INC mode in both the instances.
    b) Add an IP set in both the instances.
    c) Bind the IP set in both the instances to the VIP.
    d) Add a virtual server in the primary instance.

For steps 1 and 2, use the AWS console. For steps 3, use the Citrix ADC VPX GUI or the CLI.

Step 1. Create an Amazon virtual private cloud (VPC).

Step 2. Deploy two VPX instance in two different availability zones or in the same zone but in different subnets. Attach an EIP to the VIP of the primary VPX.

For more information about how to create a VPC and deploy a VPX instance on AWS, see Deploy a Citrix ADC VPX standalone instance on AWS and Scenario: standalone instance

Step 3. Configure high availability. You can use the Citrix ADC VPX CLI or the GUI to set up high availability.

Configure high availability by using the CLI

1. Set up high availability in INC mode in both the instances.

    On the primary node:


```
add ha node 1 <sec_ip> -inc ENABLED
```

On the secondary node:

```
add ha node 1 <prim_ip> -inc ENABLED
```

<sec_ip> refers to the private IP address of the management NIC of the secondary node

<prim_ip> refers to the private IP address of the management NIC of the primary node

2. Add the IP set in both the instances.

   Type the following command on both the instances.

   ```
   add ipset <ipsetname>
   ```

3. Bind the IP set to the VIP set on both the instances.

   Type the following command on both the instances:

   ```
   add ns ip <secondary vip> <subnet> -type VIP
   ```

   ```
   bind ipset <ipsetname> <secondary VIP>
   ```

   > **Note**
   >
   > You can bind the IP set to the primary VIP or to the secondary VIP. However, if you bind the IP set to the primary VIP, use the secondary VIP to add to the virtual server, and conversely.

4. Add a virtual server on the primary instance.

   Type the following command:

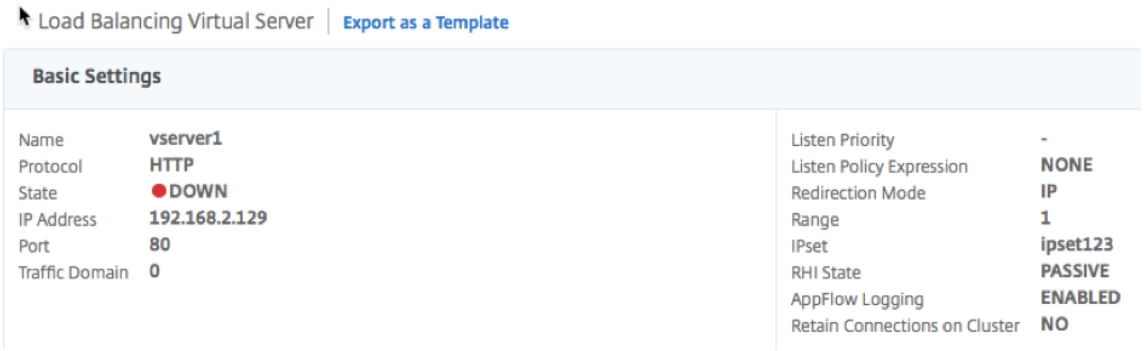
   ```
   add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
   > -ipset \<ipset_name>
   ```

Configure high availability by using the GUI

1. Set up high availability in INC mode on both the instances
2. Log on to the primary node with user name nsroot and instance ID as password.
3. From the GUI, go to **Configuration** > **System** > **High Availability**. Click **Add**.
4. At the **Remote Node IP address** field, add the private IP address of the management NIC of the secondary node.
5. Select **Turn on NIC (Independent Network Configuration)** mode on self-node.
6. Under **Remote System Login Credential**, add the user name and password for the secondary node and click **Create**.
7. Repeat the steps in the secondary node.

8. Add IP set and bind IP set to the VIP set on both the instances.
9. From the GUI, navigate to **System** > **Network** > **IPs** > **Add**.
10. Add the required values for IP Address, Netmask, IP Type (virtual IP) and click **Create**.
11. Navigate to **System** > **Network** > **IP Sets** > **Add**. Add an IP set name and click **Insert**.
12. From the IPV4s page, select the virtual IP and click **Insert**. Click **Create** to create the IP set.
13. Add a virtual server in the primary instance

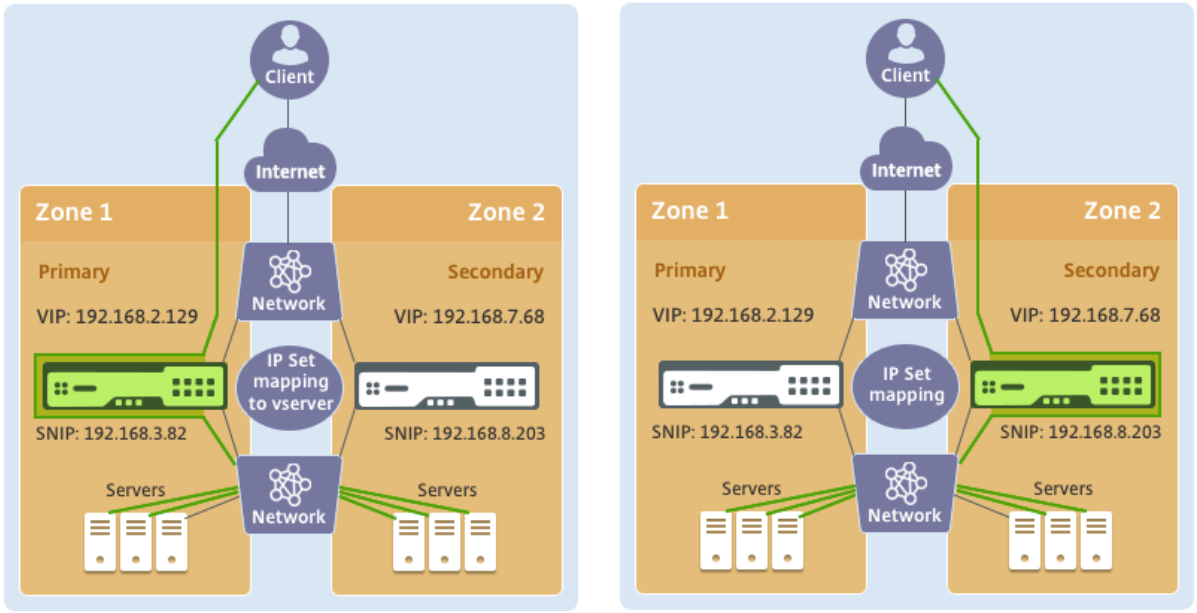
    From the GUI, go to **Configuration** > **Traffic Management** > **Virtual Servers** > **Add**.



Scenario

In this scenario, a single VPC is created. In that VPC, two VPX instances are created in two availability zones. Each instance has three subnets - one for management, one for client, and one for back-end server. An EIP is attached to the VIP of the primary node.

Diagram: This diagram illustrates the Citrix ADC VPX high availability setup in INC mode, on AWS



For this scenario, use CLI to configure high availability.

1. Set up high availability in INC mode on both the instances.

   Type the following commands on the primary and the secondary nodes.

   On primary:

   ```
   add ha node 1 192.168.6.82 -inc enabled
   ```

   Here, 192.168.6.82 refers to the private IP address of the management NIC of the secondary node.

   On secondary:

   ```
   add ha node 1 192.168.1.108 -inc enabled
   ```

   Here, 192.168.1.108 refers to the private IP address of the management NIC of the primary node.

2. Add an IP set and bind the IP set to the VIP on both the instances

   On primary:

   ```
   add ipset ipset123
   ```

   ```
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   ```

   ```
   bindipset ipset123 192.168.7.68
   ```

   On secondary:

   ```
   add ipset ipset123
   ```

   ```
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   ```



```
bind ipset ipset123 192.168.7.68
```

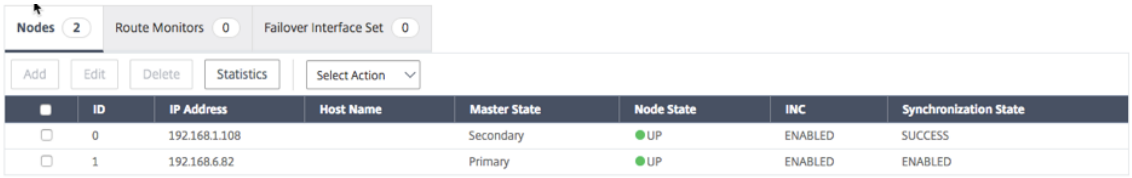
3. Add a virtual server on the primary instance.

   The following command:

   ```
   add lbvserver vserver1 http 192.168.2.129 80 -ipset ipset123
   ```

4. Save the configuration.

5. After a forced failover, the secondary becomes the new primary.



Deploy a VPX high-availability pair with private IP addresses across different AWS zones

September 17, 2021

You can configure two Citrix ADC VPX instances on two different subnets or two different AWS availability zones using private IP addresses in the INC mode. This solution can be easily integrated with the existing multizone VPX high-availability pair with elastic IP addresses. Therefore, you can use both the solutions together.

For more information about high availability, see High availability. For more information about INC, see Configuring high availability nodes in different subnets.

> Note:
>
> This deployment is supported from Citrix ADC release 13.0 build 67.39 onwards. This deployment is compatible with AWS Transit Gateway and VPC peering.

Prerequisites

Ensure that the IAM role associated with your AWS account has the following IAM permissions:


```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:AssociateAddress",
                "ec2:DisassociateAddress",
                "ec2:DescribeRouteTables",
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "iam:SimulatePrincipalPolicy",
                "iam:GetRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
         }

    ]
 }


<!--NeedCopy-->
```

How to deploy a VPX high-availability pair with private IP addresses across different AWS zones

The following is the summary of steps for deploying a VPX pair on two different subnets or two different AWS availability zones using private IP addresses.

1. Create an Amazon virtual private cloud.
2. Deploy two VPX instances in two different availability zones.
3. Configure high availability
    a) Set up high availability in INC mode in both the instances.
    b) Add the respective route tables in the VPC that points to the client interface.
    c) Add a virtual server in the primary instance.

For steps 1 and 2, use the AWS console. For step 3, use the Citrix ADC VPX GUI or the CLI.

Step 1. Create an Amazon virtual private cloud (VPC).

Step 2. Deploy two VPX instance in two different availability zones with the same number of ENI (Network Interface).

For more information about how to create a VPC and deploy a VPX instance on AWS, see Deploy a Citrix ADC VPX standalone instance on AWS and Scenario: standalone instance

Step 3. Configure the ADC VIP addresses by choosing a subnet that does not overlap with the Amazon VPC subnets. If your VPC is 192.168.0.0/16, then to configure ADC VIP addresses, you can choose any subnet from these IP address ranges:

- 0.0.0.0 - 192.167.0.0
- 192.169.0.0 - 254.255.255.0

In this example, the chosen 10.10.10.0/24 subnet and created VIPs in this subnet. You can choose any subnet other than the VPC subnet (192.168.0.0/16).

Step 4. Add a route that points to the client interface (VIP) of the primary node from the VPC route table.

From the AWS CLI, type the following command:

```
aws ec2 create-route --route-table-id rtb-2272532 --destination-cidr-
    block 10.10.10.0/24 --gateway-id <eni-client-primary>
<!--NeedCopy-->
```

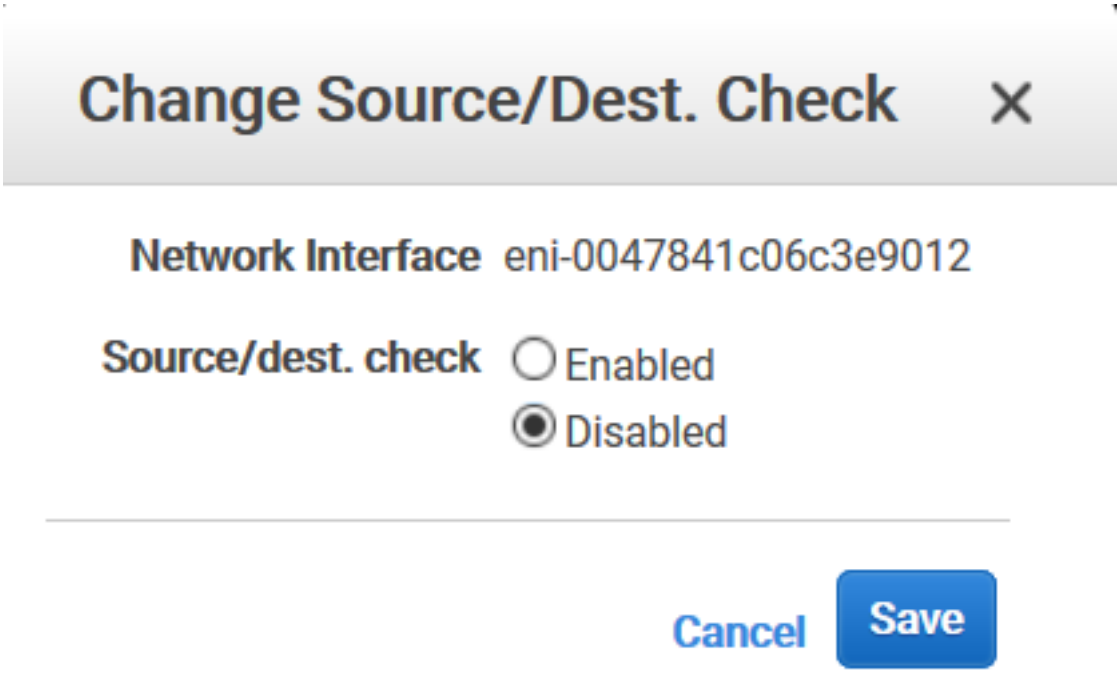
From the AWS GUI, perform the following steps to add a route:

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Route Tables**, and select the route table.
3. Choose **Actions**, and click **Edit routes**.
4. To add a route, choose **Add route**. For **Destination**, enter the destination CIDR block, a single IP address, or the ID of a prefix list. For gateway ID, select the ENI of a client interface of the primary node.

> Note
>
> You must disable **Source/Dest Check** on the client ENI of the primary instance.

To disable the source/destination checking for a network interface using the console, perform the following steps:

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface of a primary client interface, and choose **Actions**, and click **Change Source/Dest. Check**.
4. In the dialog box, choose **Disabled**, click **Save**.



Step 5. Configure high availability. You can use the Citrix ADC VPX CLI or the GUI to set up high availability.

Configure high availability by using the CLI

1. Set up high availability in INC mode in both the instances.

   On the primary node:

   ```
   1 add ha node 1 \<sec\_ip\> -inc ENABLED
   2 <!--NeedCopy-->
   ```

   On the secondary node:


```
1 add ha node 1 \<prim\_ip\> -inc ENABLED
2 <!--NeedCopy-->
```

<sec_ip> refers to the private IP address of the management NIC of the secondary node.

<prim_ip> refers to the private IP address of the management NIC of the primary node.

2. Add a virtual server on the primary instance. You must add it from the chosen subnet, for example, 10.10.10.0/24.

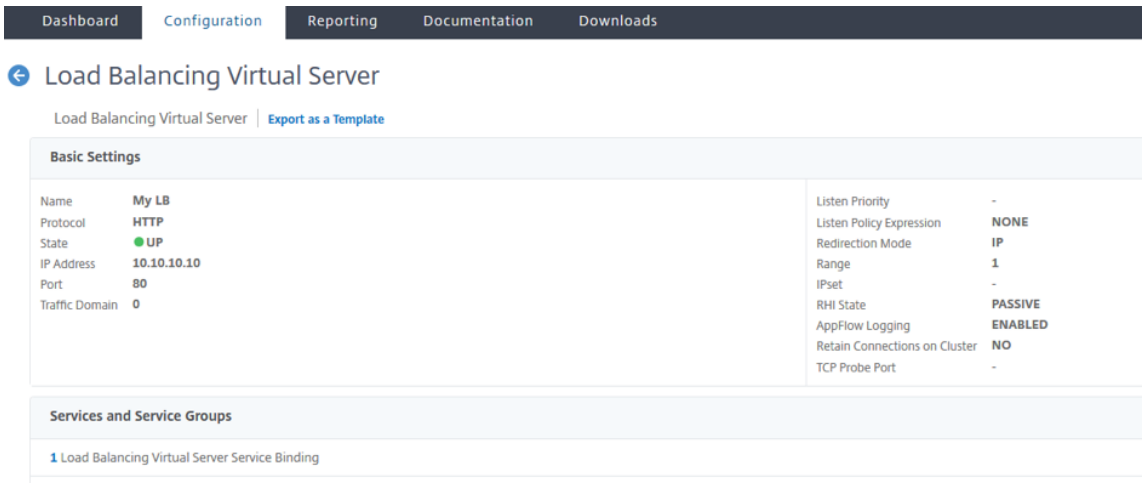
   Type the following command:

```
1 add \<server\_type\> vserver \<vserver\_name\> \<protocol\> \<
     primary\_vip\> \<port\>
2 <!--NeedCopy-->
```

Configure high availability by using the GUI

1. Set up high availability in INC mode on both the instances
2. Log on to the primary node with user name `nsroot` and instance ID as password.
3. Navigate to **Configuration** > **System** > **High Availability**, and click **Add**.
4. At the **Remote Node IP address** field, add the private IP address of the management NIC of the secondary node.
5. Select **Turn on NIC (Independent Network Configuration)** mode on self-node.
6. Under **Remote System Login Credential**, add the user name and password for the secondary node and click **Create**.
7. Repeat the steps in the secondary node.
8. Add a virtual server in the primary instance

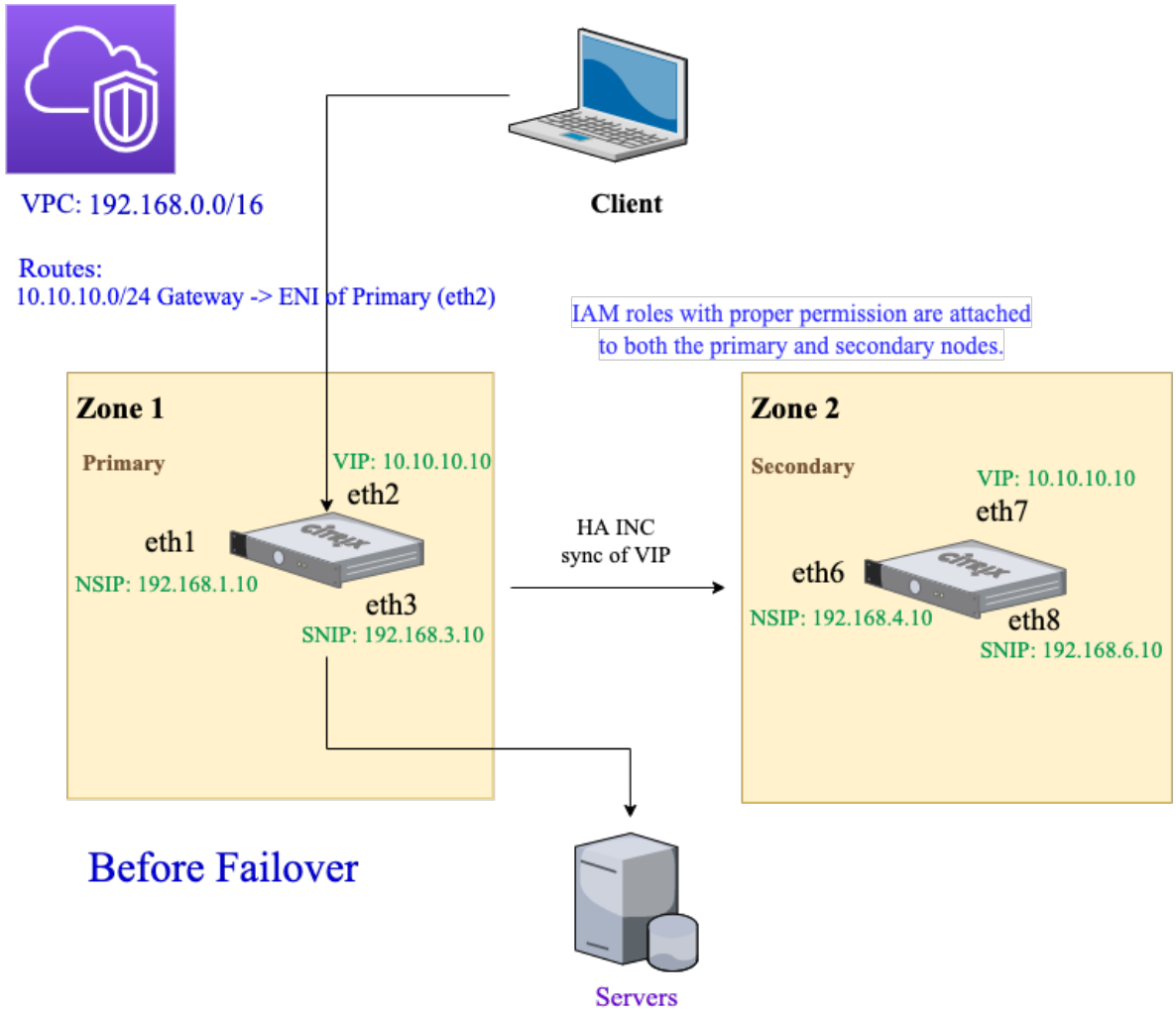
   Navigate to **Configuration** > **Traffic Management** > **Virtual Servers** > **Add**.

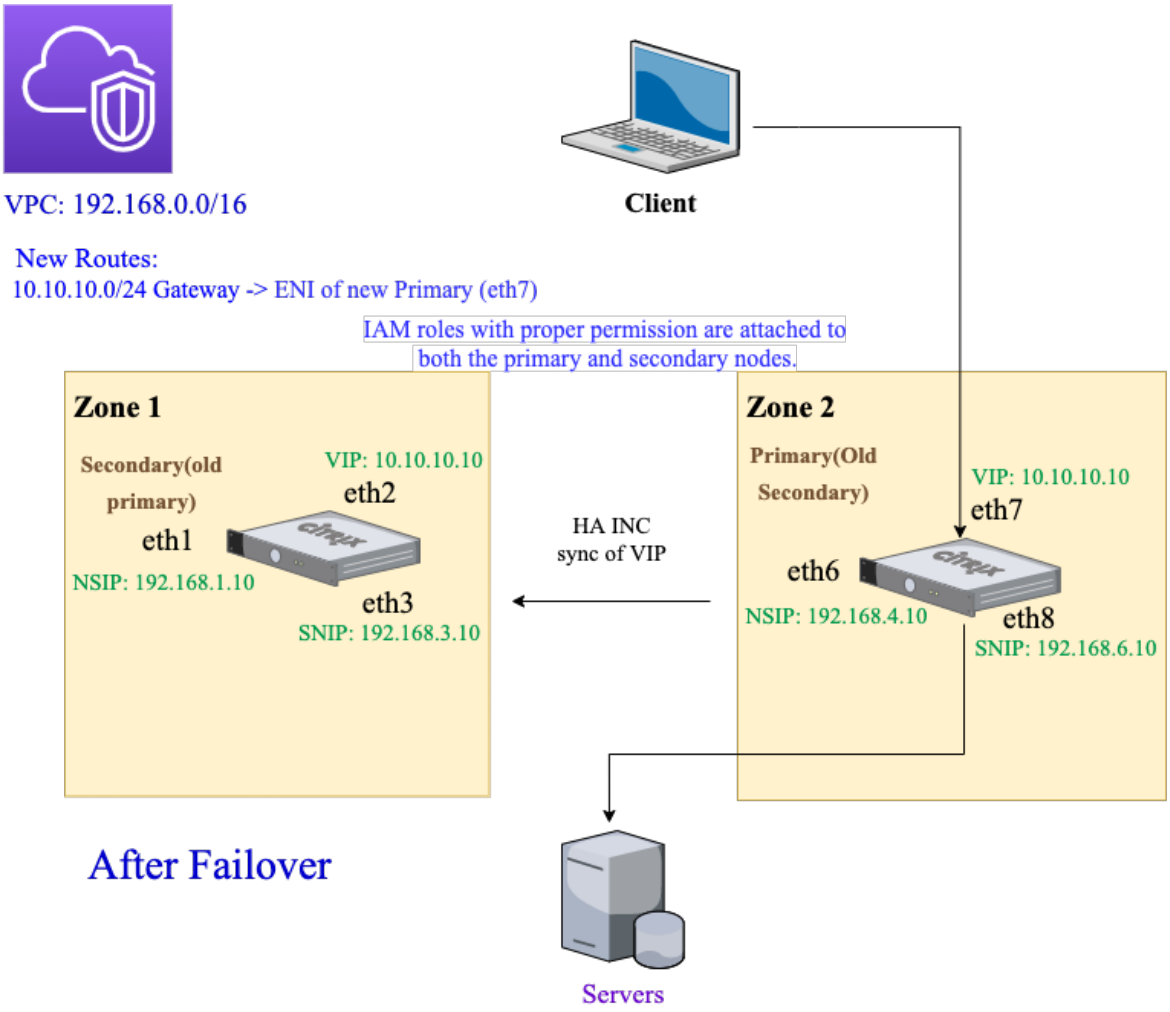


Scenario

In this scenario, a single VPC is created. In that VPC, two VPX instances are created in two availability zones. Each instance has three subnets - one for management, one for client, and one for back-end server.

The following diagrams illustrate the Citrix ADC VPX high availability setup in INC mode, on AWS. The custom subnet 10.10.10.10, which is not part of the VPC is used as VIP. Therefore, the 10.10.10.10 subnet can be used across availability zones.





For this scenario, use CLI to configure high availability.

1. Set up high availability in INC mode on both the instances.

   Type the following commands on the primary and the secondary nodes.

   On the primary node:

   ```
   add ha node 1 192.168.4.10 -inc enabled
   <!--NeedCopy-->
   ```

   Here, 192.168.4.10 refers to the private IP address of the management NIC of the secondary node.

   On the secondary node:

   ```
   add ha node 1 192.168.1.10 -inc enabled
   <!--NeedCopy-->
   ```

   Here, 192.168.1.10 refers to the private IP address of the management NIC of the primary node.

2. Add a virtual server on the primary instance.

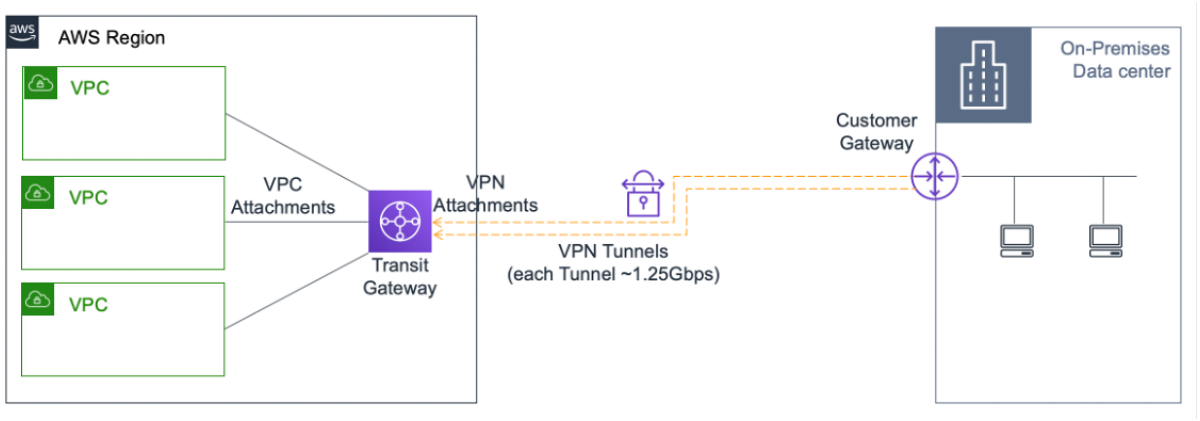
   Type the following command:

```
add lbvserver vserver1 http 10.10.10.10 80
<!--NeedCopy-->
```

3. Save the configuration.
4. After a forced failover:
   - The secondary instance becomes the new primary instance.
   - The VPC route pointing to the primary ENI migrates to the secondary client ENI.
   - Client traffic resumes to the new primary instance.

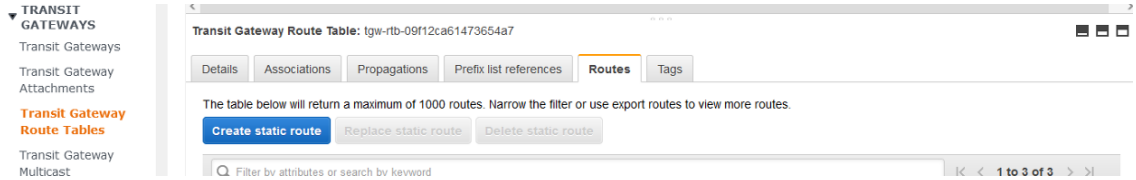
AWS Transit Gateway configuration for HA private IP solution

You need AWS Transit Gateway to make the private VIP subnet routable within the internal network, across AWS VPCs, regions, and On-premises networks. The VPC must connect to AWS Transit Gateway. A static route for the VIP subnet or IP pool inside the AWS Transit Gateway route table is created and pointed towards the VPC.

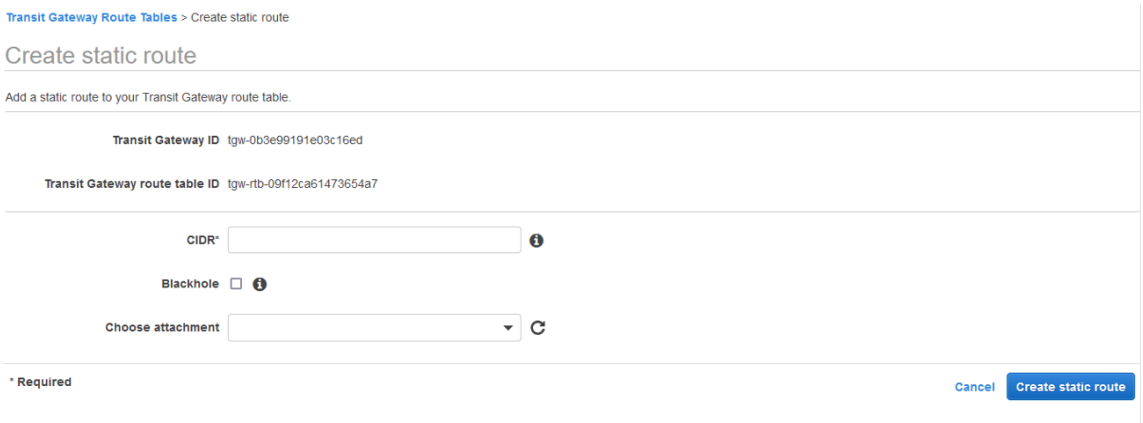


To configure AWS Transit Gateway, follow these steps:

1. Open the Amazon VPC console.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Choose the **Routes** tab, and click **Create static route**.



4. Create a static route where CIDR points to your private VIPS subnet and attachment points to the VPC having ADC VPX.



5. Click **Create static route**, then choose **Close**.

Deploy a Citrix ADC VPX instance on AWS Outposts

September 14, 2021

AWS Outposts is a pool of AWS compute and storage capacity deployed at your site. Outposts provides AWS infrastructure and services in your on-premises location. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can use the same Citrix ADC VPX instances, AWS APIs, tools, and infrastructure across on-premises and the AWS cloud for a consistent hybrid experience.

You can create subnets on your Outposts and specify them when you create AWS resources such as EC2 instances, EBS volumes, ECS clusters, and RDS instances. Instances in the Outposts subnets communicate with other instances in the AWS Region using private IP addresses, all within the same Amazon Virtual Private Cloud (VPC).

For more information, see the AWS Outposts user guide.

How AWS Outposts works

AWS Outposts is designed to operate with a constant and consistent connection between your Outposts and an AWS Region. To achieve this connection to the Region, and to the local workloads in your on-premises environment, you must connect your Outpost to your on-premises network. Your on-premises network must provide WAN access back to the Region and to the internet. The internet must also provide LAN or WAN access to the local network where your on-premises workloads or applications reside.

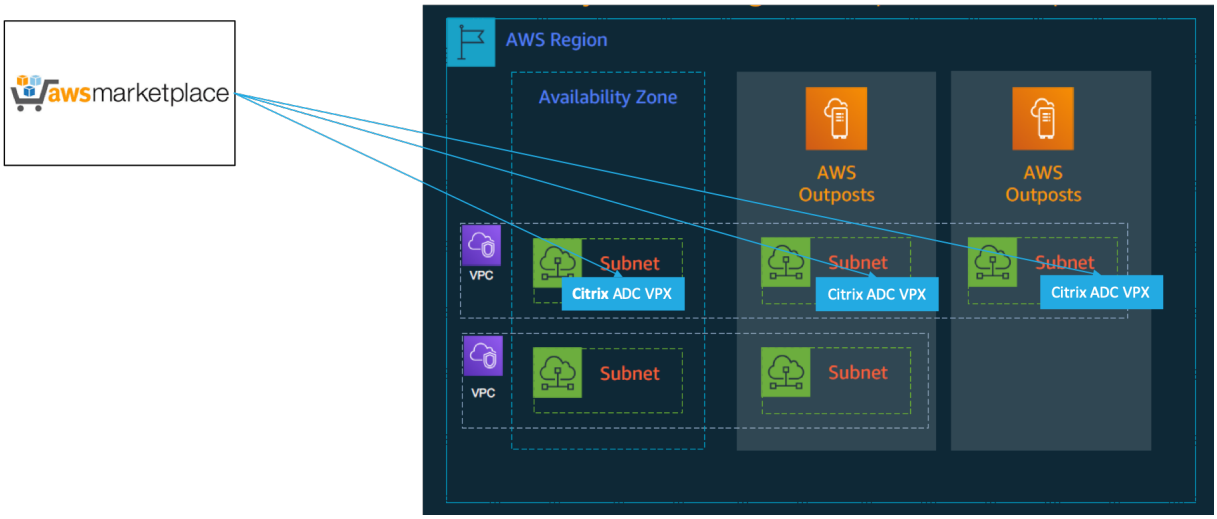
Prerequisite

- You must install an AWS Outposts at your site.
- The AWS Outposts' compute and storage capacity must be available for use.

For more information on how to place an order for AWS Outposts, see the following AWS documentation:
https://aws.amazon.com/blogs/aws/aws-outposts-now-available-order-your-racks-today/

Deploy a Citrix ADC VPX instance on AWS Outposts by using the AWS web console

The following figure depicts a simple deployment of Citrix ADC VPX instances on the Outposts. The Citrix ADC AMI present in the AWS Marketplace is also deployed in the Outposts.



Log in to the AWS web console and complete the following steps to deploy ADC VPX EC2 instances on your AWS Outposts.

1. Create a key pair.
2. Create a Virtual Private Cloud (VPC).
3. Add more subnets.
4. Create security groups and security rules.
5. Add route tables.
6. Create an internet gateway.
7. Create an ADC VPX instance by using the AWS EC2 service.
   From the AWS dashboard, navigate to **Compute > EC2 > Launch Instance > AWS Marketplace**.
8. Create and attach more network interfaces.
9. Attach elastic IPs to the management NIC.
10. Connect to the VPX instance.

For detailed instructions on each of the steps, see Deploy a Citrix ADC VPX instance on AWS by using the AWS web console.

For high availability within same availability zone deployment, see Deploy a high availability pair on AWS.

Add back-end AWS Autoscaling service

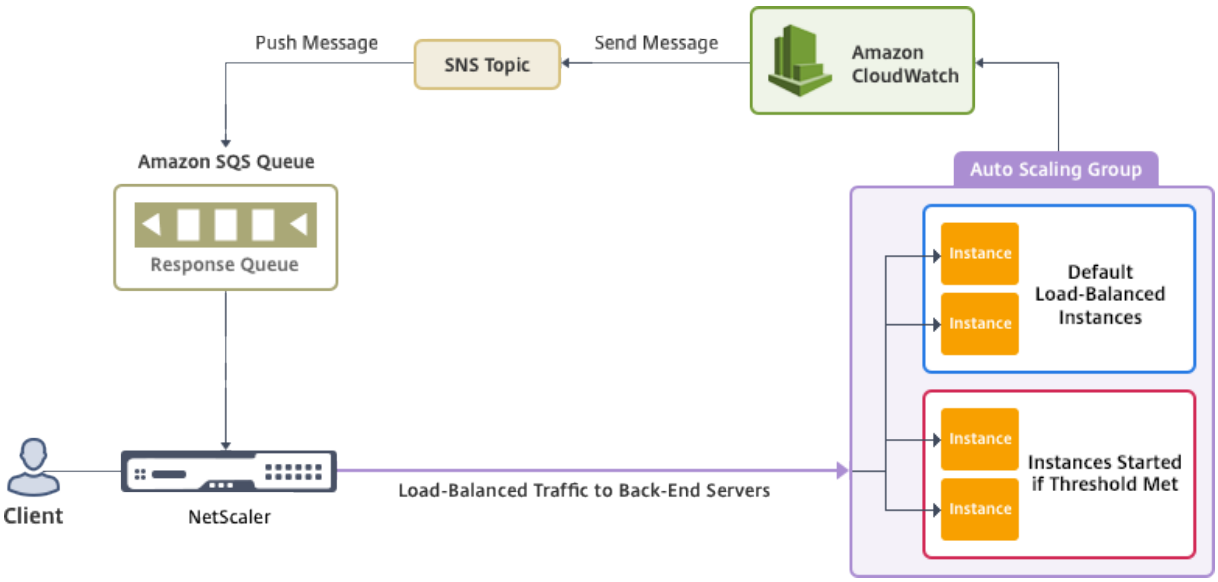
September 14, 2021

Efficient hosting of applications in a cloud involves easy and cost-effective management of resources depending on the application demand. To meet increasing demand, you have to scale network resources upward. Whether demand subsides, you need to scale down to avoid the unnecessary cost of idle resources. To minimize the cost of running the application by deploying only as many instances as are necessary during any given time, you constantly have to monitor traffic, memory and CPU use, and so on. However, monitoring traffic manually is cumbersome. For the application environment to scale up or down dynamically, you must automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

Integrated with the AWS Auto Scaling service, the Citrix ADC VPX instance provides the following advantages:

- **Load balance and management**: Auto configures servers to scale up and scale down, depending on demand. The VPX instance auto detects Autoscale groups in the back-end subnet and allows a user to select the Autoscale groups to balance the load. All of this is done by auto configuring the virtual and subnet IP addresses on the VPX instance.
- **High availability**: Detects Autoscale groups that span multiple availability zones and load-balance servers.
- **Better network availability**: The VPX instance supports:
  - Back-end servers on different VPCs, by using VPC peering
  - Back-end servers on same placement groups
  - Back-end servers on different availability zones
- **Graceful connection termination**: Removes Autoscale servers gracefully, avoiding loss of client connections when scale-down activity occurs, by using the Graceful Timeout feature.

Diagram: AWS Autoscaling service with a Citrix ADC VPX Instance



This diagram illustrates how the AWS Autoscaling service is compatible with a Citrix ADC VPX instance (Load balancing virtual server). For more information, see the following AWS topics.

- Autoscaling groups
- CloudWatch
- Simple Notification Service (SNS)
- Simple Queue Service (Amazon SQS)

Before you begin

Before you start using Autoscaling with your Citrix ADC VPX instance, you must complete the following tasks.

1. Read the following topics:
   - Prerequisites
   - Limitation and usage guidelines
2. Create a Citrix ADC VPX instance on AWS according to your requirement.
   - For more information about how to create a Citrix ADC VPX standalone instance, see Deploy a Citrix ADC VPX standalone instance on AWS and Scenario: standalone instance
   - For more information about how to deploy VPX instances in HA mode, see Deploy a high availability pair on AWS.

> **Note**
>
> Citrix recommends the CloudFormation template for creating Citrix ADC VPX instances on AWS.

> Citrix recommends you create three interfaces: one for management (NSIP), one for client-facing LB virtual server (VIP), and one for subnet IP (NSIP).

3. Create an AWS Autoscale group. If you don't have an existing Autoscaling configuration, you must:

   a) Create a Launch Configuration

   b) Create an Autoscaling Group

   c) Verify the Autoscaling Group

   For more information, see http://docs.aws.amazon.com/autoscaling/latest/userguide/GettingStartedTutorial.html.

4. In the AWS Autoscale group, you must specify at least one scale-down policy. The Citrix ADC VPX instance supports only the Step scaling policy. The Simple scaling policy and Target tracking scaling policy are not supported for Autoscale group.

Add the AWS Autoscaling service to a Citrix ADC VPX instance

You can add the Autoscaling service to a VPX instance with a single click by using the GUI. Complete these steps to add the Autoscaling service to the VPX instance:

1. Log on to the VPX instance by using your credentials for `nsroot`.
2. When you log on to the Citrix ADC VPX instance for the first time, you see the default Cloud Profile page. Select the AWS Autoscaling group from the drop-down menu and click **Create** to create a cloud profile. Click **Skip** if you want to create the cloud profile later.

   Points to keep in mind while creating a Cloud Profile: By default the CloudFormation Template creates and attaches the below IAM Role.

```
{


    "Version": "2012-10-17",

    "Statement": \[

     {


            "Action": \[

                "ec2:DescribeInstances",

                "ec2:DescribeNetworkInterfaces",
```



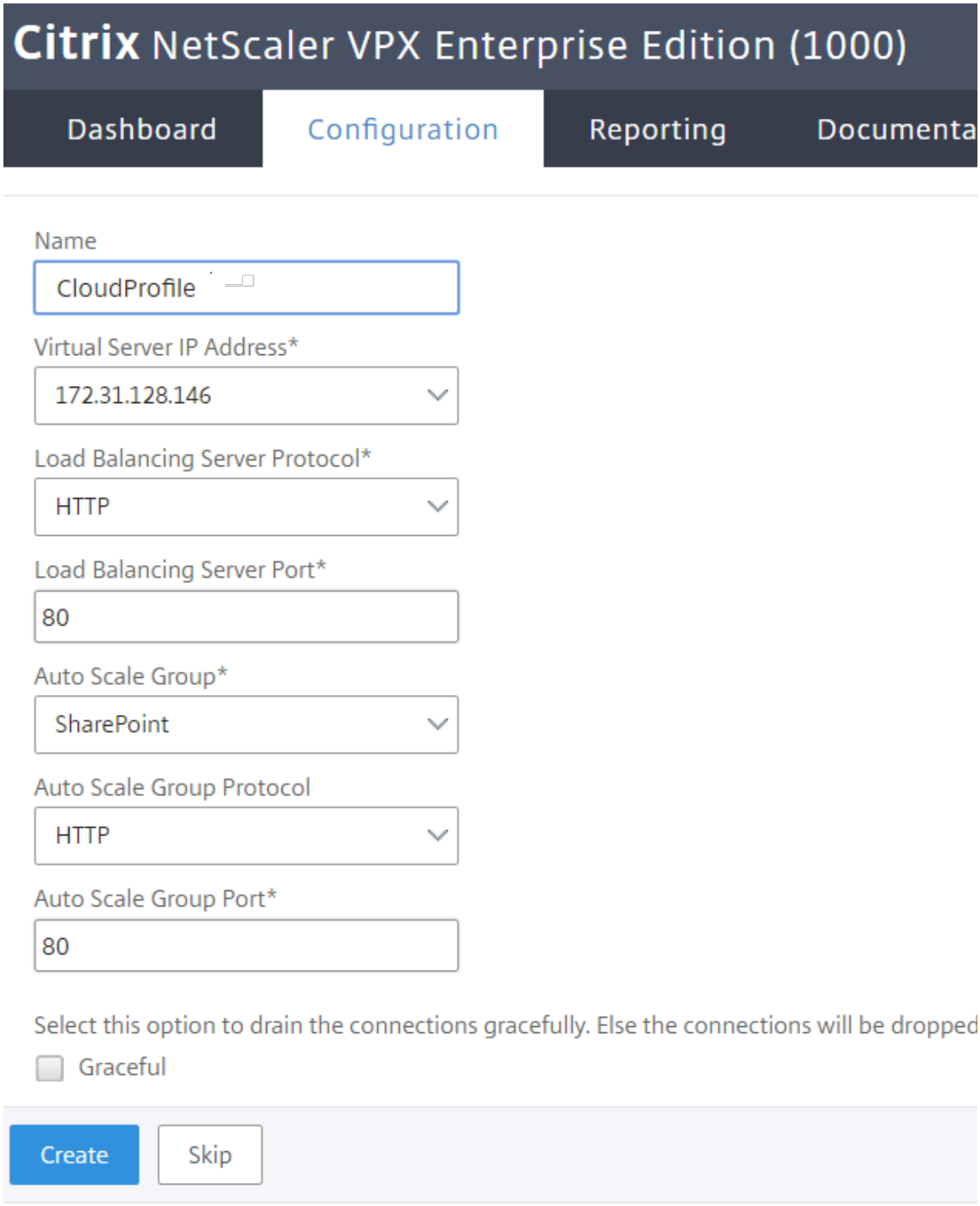
```
16
17                  "ec2:DetachNetworkInterface",
18
19                  "ec2:AttachNetworkInterface",
20
21                  "ec2:StartInstances",
22
23                  "ec2:StopInstances",
24
25                  "ec2:RebootInstances",
26
27                  "autoscaling:\*",
28
29                  "sns:\*",
30
31                  "sqs:\*"
32
33               “iam: SimulatePrincipalPolicy”
34
35               “iam: GetRole”
36
37              \],
38
39              "Resource": "\*",
40
41              "Effect": "Allow"
42
43           }
44
45
46      \]
47
48   }
49
50  <!--NeedCopy-->
```

Ensure the IAM Role of an instance has proper permissions.

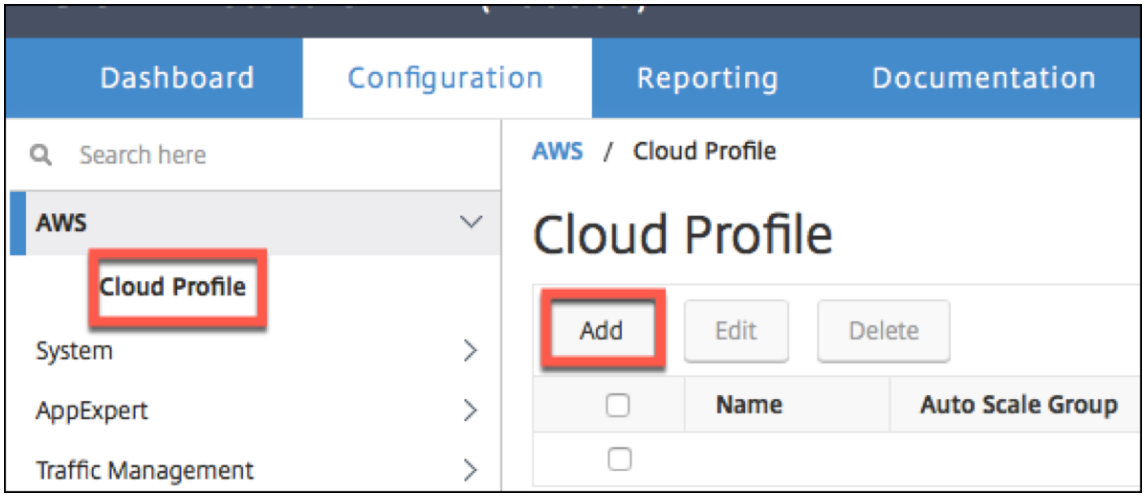
- The virtual server IP address is autopopulated from the free IP address available to the VPX instance. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/MultipleIP.html#ManageMultipleIP
- Autoscale group is prepopulated from the Autoscale group configured on your AWS account. http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html.

- While selecting the Autoscaling Group protocol and port, ensure your servers listen on those protocol and ports, and you bind the correct monitor in the service group. By default, the TCP monitor is used.
- For SSL Protocol type Autoscaling, after you create the Cloud Profile the load balance virtual server or service group is down because of a missing certificate. You can bind the certificate to the virtual server or service group manually.
- Select the Graceful Timeout option to remove Autoscale servers gracefully. If this option is not selected the server is the Autoscale group is removed immediately after the load goes down, which might cause service interruption for the existing connected clients. Selecting Graceful and giving a timeout means in the event of scale down. The VPX instance does not remove the server immediately but marks one of the servers for graceful deletion. During this period, the instance does not allow new connections to this server. Existing connection are served until the timeout occurs, and after a timeout, the VPX instance removes the server.

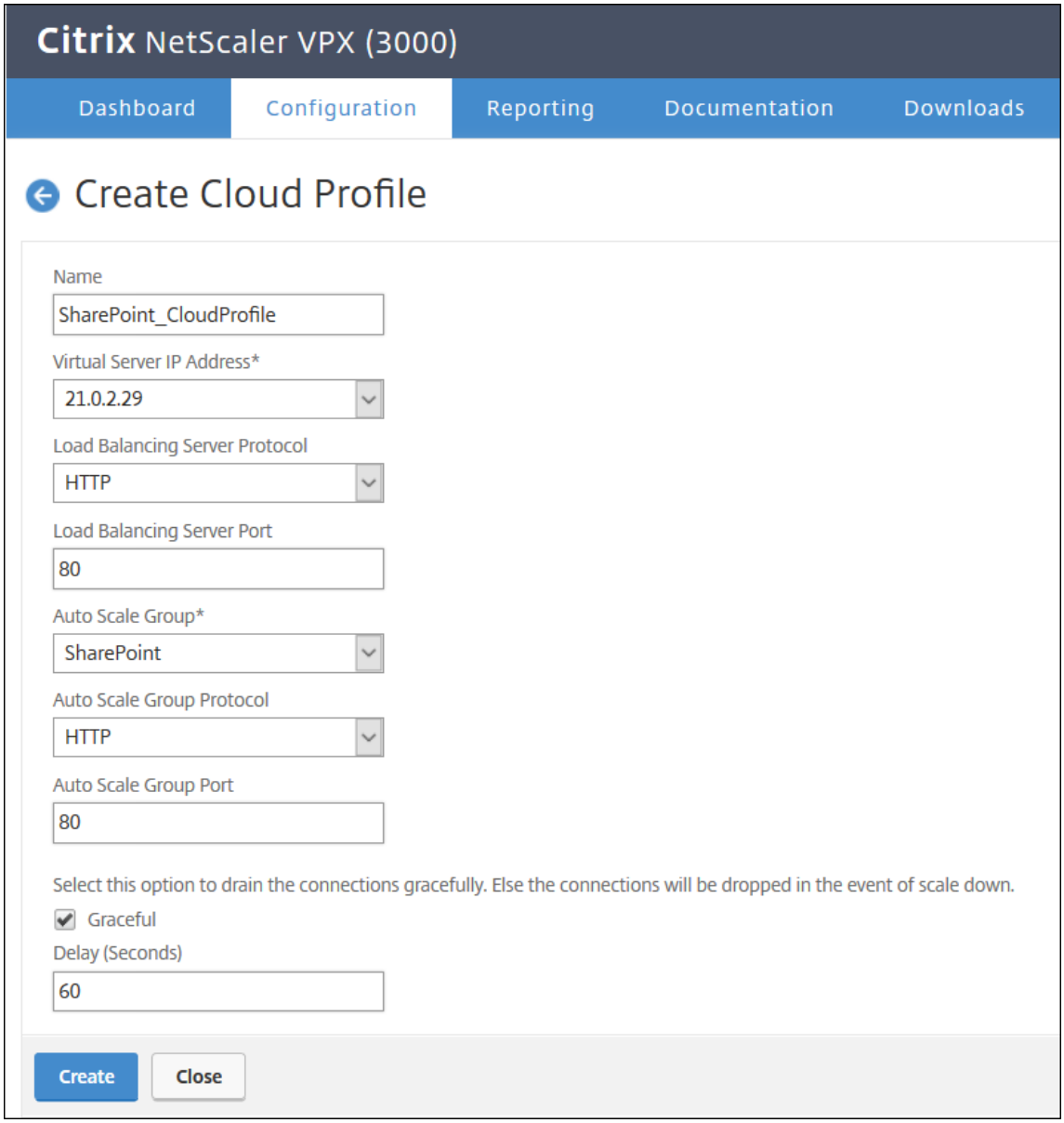
Figure: Default Cloud Profile page



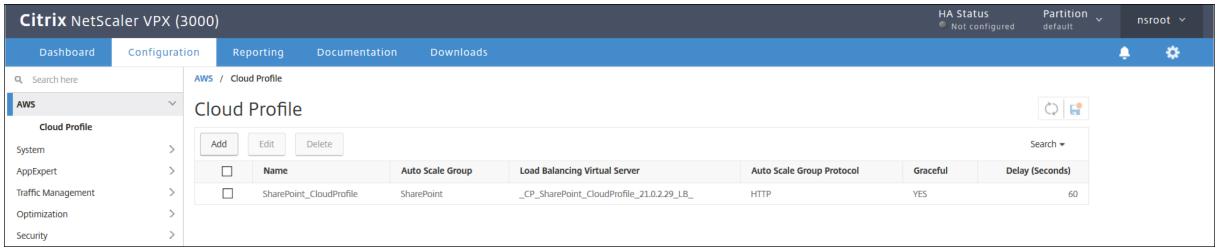
3. After the first time logon if you want to create Cloud Profile, on the GUI go to **System > AWS > Cloud Profile** and click **Add**.



The **Create Cloud Profile** configuration page appears.



Cloud Profile creates a Citrix ADC load-balancing virtual server and a service group with members as the servers of the Autoscaling group. Your back-end servers must be reachable through the SNIP configured on the VPX instance.



> **Note**
>
> To view Autoscale-related information in the AWS console, go to **EC2** > **Dashboard** > **Auto Scaling** > **Auto Scaling Group**.

Configure a Citrix ADC VPX instance to use SR-IOV network interface

September 14, 2021

> **Note**
>
> Support for SR-IOV interfaces in a high availability setup is available from Citrix ADC release 12.0 57.19 onwards.

After you have created a Citrix ADC VPX instance on AWS, you can configure the virtual appliance to use SR-IOV network interfaces, by using the AWS CLI.

In all Citrix ADC VPX models, except Citrix ADC VPX AWS Marketplace Editions of 3G and 5G, SR-IOV is not enabled in the default configuration of a network interface.

Before you start the configuration, read the following topics:

- Prerequisites
- Limitations and Usage Guidelines

This section includes the following topics:

- Change the Interface Type to SR-IOV
- Configure SR-IOV on a High Availability Setup

Change the interface type to SR-IOV

You can run the show interface summary command to check the default configuration of a network interface.

Example 1: The following CLI screen capture shows the configuration of a network interface where SR-IOV is enabled by default on Citrix ADC VPX AWS Marketplace Editions of 3G and 5G.

```
> show interface summary
-------------------------------------------------------------------------
      Interface  MTU       MAC                Suffix
-------------------------------------------------------------------------
1       1/1      1500      0a:1e:2e:17:a2:37  Intel 82599 10G VF Interface
2       LO/1     1500      0a:1e:2e:17:a2:37  Netscaler Loopback interface
 Done
```

Example 2: The following CLI screen capture shows the default configuration of a network interface where SR-IOV is not enabled.


```
 Done
> sh int s
-------------------------------------------------------------------------------
      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------
1     1/1        1500       12:fc:04:c5:d0:12  NetScaler Virtual Interface
2     LO/1       1500       12:fc:04:c5:d0:12  Netscaler Loopback interface
 Done
>
```

For more information about changing the interface type to SR-IOV, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.html

To change the interface type to SR-IOV

1. Shut down the Citrix ADC VPX instance running on AWS.
2. To enable SR-IOV on the network interface, type the following command in the AWS CLI.

   ```
   $ aws ec2 modify-instance-attribute --instance-id \<instance\\_id\> --sriov-net-support simple
   ```

3. To check if SR-IOV has been enabled, type the following command in the AWS CLI.

   ```
   $ aws ec2 describe-instance-attribute --instance-id \<instance\\_id\> --attribute sriovNetSupport
   ```

   Example 3: Network interface type changed to SR-IOV, by using the AWS CLI.

   ```
   aws ec2 modify-instance-attribute --instance-id i-008c1230aaf303bee --sriov-net-support simple
   aws ec2 describe-instance-attribute --instance-id   i-008c1230aaf303bee  --attribute sriovNetSupport
   {
       "InstanceId": "i-008c1230aaf303bee",
       "SriovNetSupport": {
           "Value": "simple"
       }
   }
   ```

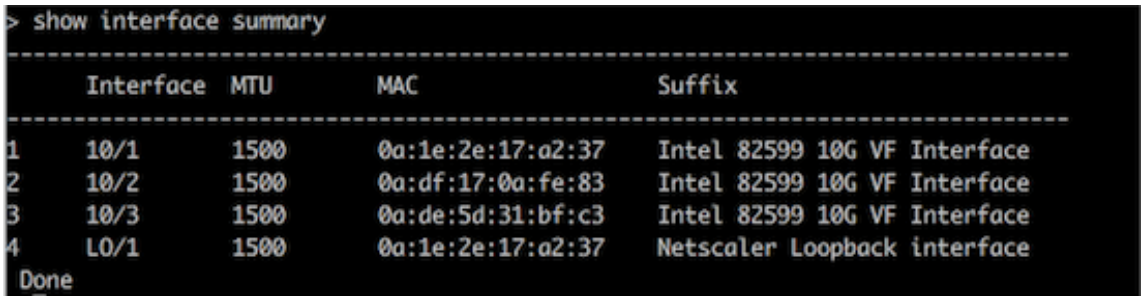
   If SR-IOV is not enabled, value for SriovNetSupport is absent.

   Example 4: In the following example, SR-IOV support is not enabled.

   ```
   {
       "InstanceId": "i-0c3e84cfa65b04cc8",
       "SriovNetSupport": {}
   }
   ```

4. Power on the VPX instance. To see the changed status of the network interface, type “show interface summary” in the CLI.

Example 5: The following screen capture shows the network interfaces with SR-IOV enabled. The interfaces 10/1, 10/2, 10/3 are SR-IOV enabled.



These steps complete the procedure to configure VPX instances to use SR-IOV network interfaces.

Configure SR-IOV on a high availability setup

High availability is supported with SR-IOV interfaces from Citrix ADC release 12.0 build 57.19 onwards.

If the high availability setup was deployed manually or by using the Citrix CloudFormation template for Citrix ADC version 12.0 56.20 and lower, the IAM role attached to the high availability setup must have the following privileges:

- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- autoscaling:*
- sns:*
- sqs:*
- IAM:SimulatePrincipalPolicy
- IAM:GetRole

By default, the Citrix CloudFormation template for Citrix ADC version 12.0 57.19 automatically adds the required privileges to the IAM role.

> **Note**
>
> A high availability setup with SR-IOV Interfaces takes around 100 seconds of downtime.

Related resources:

For more information about IAM roles, see AWS documentation.

Configure a Citrix ADC VPX instance to use Enhanced Networking with AWS ENA

September 14, 2021

After you have created a Citrix ADC VPX instance on AWS, you can configure the virtual appliance to use Enhanced Networking with AWS Elastic Network Adapter (ENA), by using AWS CLI.

Coupled with AWS ENA, enhanced networking provides higher bandwidth, higher packet-per-second (PPS) performance, and consistently lower inter-instance latencies.

Before you start the configuration, read the following topics:

- Prerequisites
- Limitations and Usage Guidelines

The following HA configurations are supported for ENA-enabled instances:

- Private IP addresses can be moved within the same availability zone.
- Elastic IP addresses can be moved across availability zones.

Upgrade a Citrix ADC VPX instance on AWS

September 14, 2021

You can upgrade the EC2 instance type, throughput, software edition, and the system software of a Citrix ADC VPX running on AWS. For certain types of upgrades, Citrix recommends using the High Availability Configuration method to minimize downtime.

> Note:
>
> - Citrix ADC software release 10.1.e-124.1308.e or later for a Citrix ADC VPX AMI (including both utility license and customer license) does not support the M1 and M2 instance families.
> - Because of changes in VPX instance support, downgrading from 10.1.e-124 or a later release to 10.1.123.x or an earlier release is not supported.
> - Most of the upgrades do not require the launch of a new AMI, and the upgrade can be done on the current Citrix ADC AMI instance. If you do want to upgrade to a new Citrix ADC AMI instance, use the high availability configuration method.

Change the EC2 instance type of a Citrix ADC VPX instance on AWS

If your Citrix ADC VPX instances are running release 10.1.e-124.1308.e or later, you can change the EC2 instance type from the AWS console as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

You can also use the above procedure to change the EC2 instance type for a release, earlier than 10.1.e-124.1308.e, unless you want to change the instance type to M3. In that case, you must first follow the standard Citrix ADC upgrade procedure, at, to upgrade the Citrix ADC software to 10.1.e-124 or a later release, and then follow the above steps.

Upgrade the throughput or software edition of a Citrix ADC VPX instance on AWS

To upgrade the software edition (for example, to upgrade from Standard to Premium edition) or throughput (for example, to upgrade from 200 Mbps to 1000mbps), the method depends on the instance's license.

Using a customer license (Bring-Your-Own-License)

If you are using a customer license, you can purchase and download the new license from the Citrix website, and then install the license on the VPX instance. For more information about downloading and installing a license from the Citrix website, see the VPX Licensing Guide.

Using a utility license (Utility license with hourly fee)

AWS does not support direct upgrades for fee-based instances. To upgrade the software edition or throughput of a fee based Citrix ADC VPX instance, launch a new AMI with the desired license and capacity and migrate the older instance configuration to the new instance. This can be achieved by using a Citrix ADC high availability configuration as described in Upgrade to a new Citrix ADC AMI instance by using a Citrix ADC high availability configuration subsection in this page.

Upgrade the system software of a Citrix ADC VPX instance on AWS

If you need to upgrade a VPX instance running 10.1.e-124.1308.e or a later release, follow the standard Citrix ADC upgrade procedure at Upgrade and downgrade a Citrix ADC appliance.

If you need to upgrade a VPX instance running a release older than 10.1.e-124.1308.e to 10.1.e-124.1308.e or a later release, first upgrade the system software, and then change the instance type to M3 as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

Upgrade to a new Citrix ADC AMI instance by using a Citrix ADC high availability configuration

To use the high availability method of upgrading to a new Citrix ADC AMI instance, perform the following tasks:

- Create a new instance with the desired EC2 instance type, software edition, throughput, or software release from the AWS marketplace.
- Configure high availability between the old instance (to be upgraded) and the new instance. After high availability is configured between the old and the new instance, configuration from the old instance is synchronized to the new instance.
- Force an HA failover from the old instance to the new instance. As a result, the new instance becomes primary and starts receiving traffic.
- Stop, and reconfigure or remove the old instance from AWS.

Prerequisites and points to consider

- Ensure you understand how high availability works between two Citrix ADC VPX instances on AWS. For more information about high availability configuration between two Citrix ADC VPX instances on AWS, see Deploy a high availability pair on AWS.
- You must create the new instance in the same availability zone as the old instance, having the exact same security group and subnet.
- High availability setup requires access and secret keys associated with the user's AWS Identity and Access Management (IAM) account for both instances. If the correct key information is not used when creating VPX instances, the HA setup fails. For more information about creating an IAM account for a VPX instance, see Prerequisites.
  - You must use the EC2 console to create the new instance. You cannot use the AWS 1-click launch, because it does not accept the access and secret keys as the input.
  - The new instance must have only one ENI interface.

To upgrade a Citrix ADC VPX Instance by using a high availability configuration, follow these steps:

1. Configure high availability between the old and the new instance. To configure high availability between two Citrix ADC VPX instances, at the command prompt of each instance, type:

   - `add ha node <nodeID> <IPaddress of the node to be added>`
   - `save config`

   Example:

   At the command prompt of the old instance, type:

   ```
   1 add ha node 30 192.0.2.30
   2 Done
   3 <!--NeedCopy-->
   ```


At the command prompt of the new instance, type:

```
add ha node 10 192.0.2.10
Done
<!--NeedCopy-->
```

Note the following:

- In the HA setup, the old instance is the primary node and the new instance is the secondary node.
- The NSIP IP address is not copied from the old instance to the new instance. Therefore, after the upgrade, your new instance has a different management IP address from the previous one.
- The `nsroot` account password of the new instance is set to that of the old instance after HA synchronization.

For more information about high availability configuration between two Citrix ADC VPX instances on AWS, see Deploy a high availability pair on AWS.

2. Force an HA failover. To force a failover in a high availability configuration, at the command prompt of either of the instances, type:

```
force HA failover
<!--NeedCopy-->
```

As the result of forcing a failover, the ENIs of the old instance are migrated to the new instance and traffic flows through the new instance (the new primary node). The old instance (the new secondary node) restarts.

If the following warning message appears, type N to abort the operation:

```
[WARNING]:Force Failover may cause configuration loss, peer health
    not optimum. Reason(s):
HA version mismatch
HA heartbeats not seen on some interfaces
Please confirm whether you want force-failover (Y/N)?
<!--NeedCopy-->
```

The warning message appears because the system software of the two VPX instances is not HA compatible. As a result, the configuration of the old instance cannot be automatically synced to the new instance during a forced failover.

Following is the workaround for this issue:

a) At the Citrix ADC shell prompt of the old instance, type the following command to create a backup of the configuration file (ns.conf):


```
copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp
```

b) Remove the following line from the backup configuration file (ns.conf.bkp):

- `set ns config -IPAddress <IP> -netmask <MASK>`

For example, `set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0`

c) Copy the old instance's backup configuration file (ns.conf.bkp) to the /nsconfig directory of the new instance.

d) At the Citrix ADC shell prompt of the new instance, type the following command to load the old instance's configuration file (ns.conf.bkp) on the new instance:

- `batch -f /nsconfig/ns.conf.bkp`

e) Save the configuration on the new instance.

- `save conifg`

f) At the command prompt of either of the nodes, type the following command to force a failover, and then type Y for the warning message to confirm the force failover operation:

- `force ha failover`

Example:

```
1         > force ha failover
2
3 WARNING]:Force Failover may cause configuration loss, peer health
     not optimum.
4         Reason(s):
5         HA version mismatch
6         HA heartbeats not seen on some interfaces
7         Please confirm whether you want force-failover (Y/N)? Y
8 <!--NeedCopy-->
```

3. Remove the HA configuration, so that the two instances are no longer in an HA configuration. First remove the HA configuration from the secondary node and then remove the HA configuration from the primary node.

   To remove an HA configuration between two Citrix ADC VPX instances, at the command prompt of each instance, type:

```
1     > remove ha node \<nodeID\>
2     >  save config
3 <!--NeedCopy-->
```


For more information about high availability configuration between two VPX instances on AWS, see Deploy a high availability pair on AWS.

Example:

At the command prompt of the old instance (new secondary node), type:

```
1          > remove ha node 30
2            Done
3          > save config
4            Done
5  <!--NeedCopy-->
```

At the command prompt of the new instance (new primary node), type:

```
1          > remove ha node 10
2            Done
3          > save config
4            Done
5  <!--NeedCopy-->
```

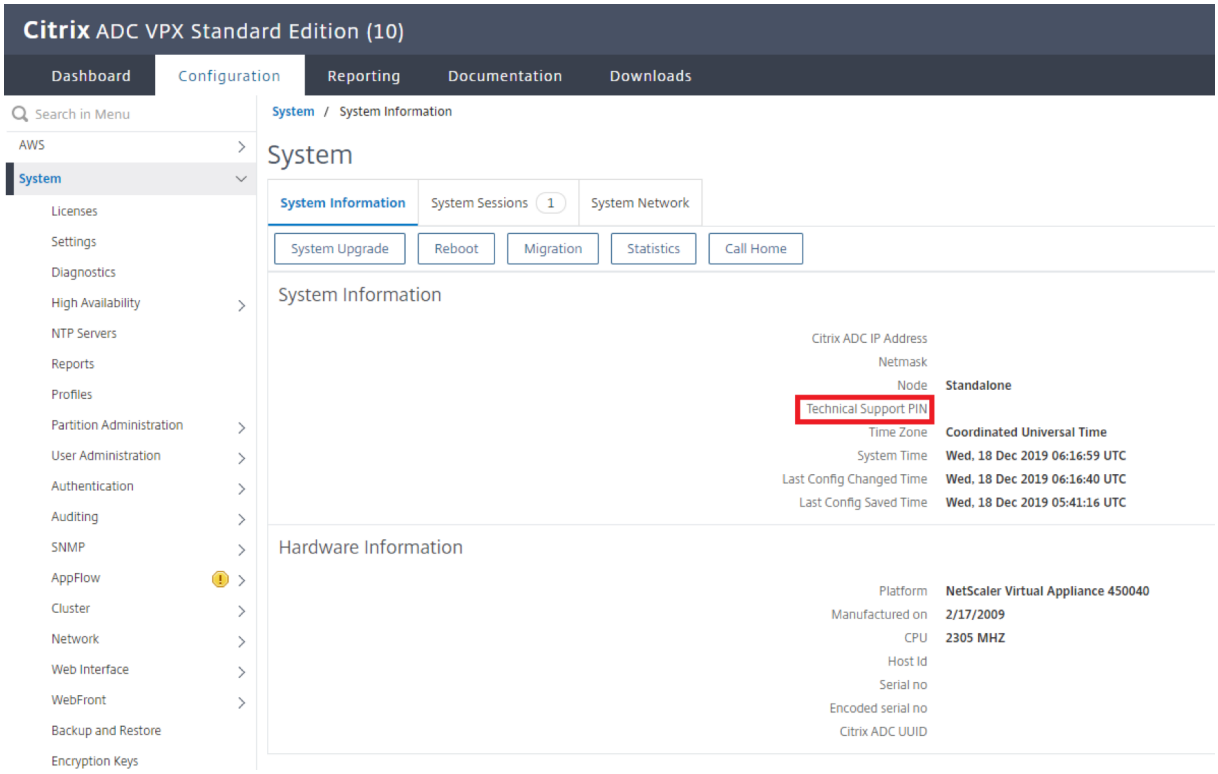
Troubleshoot a VPX instance on AWS

September 14, 2021

Amazon does not provide console access to a Citrix ADC VPX instance. To troubleshoot, you have to use the AWS GUI to view the activity log. You can debug only if the network is connected. To view an instance's System Log, right-click the instance and select System Log.

Citrix provides support for AWS Marketplace-licensed Citrix ADC VPX instances (utility license with hourly fee) on AWS. To file a support case, find your AWS account number and support PIN code, and call Citrix support. You will also be asked for your name and email address. To find the support PIN, log on to the VPX GUI and navigate to the System page.

Here is an example of a system page showing the support PIN.



AWS FAQs

September 14, 2021

- **Does a Citrix ADC VPX instance support the encrypted volumes in AWS?**

  Encryption and decryption happen at the hypervisor level, and hence it works seamlessly with any instance. For more information about the encrypted volumes see the following AWS document:

  https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html

- **What is the best way to provision Citrix ADC VPX instance on AWS?**

  You can provision a Citrix ADC VPX instance on AWS by any of the following ways:

  - AWS CloudFormation Template (CFT) in AWS marketplace
  - Citrix ADM
  - AWS Quick Starts
  - Citrix AWS CFTs in GitHub
  - Citrix Terraform Scripts in GitHub
  - Citrix Ansible Playbooks in GitHub
  - AWS EC2 launch workflow

You can choose any of the listed options based on the automation tool that you use.

For more details about the options, see Citrix ADC VPX on AWS.

- **How to upgrade Citrix ADC VPX instance in AWS?**

  To upgrade the Citrix ADC VPX instance in AWS, you can upgrade the system software or upgrade to a new Citrix ADC VPX Amazon Machine Image (AMI) by following the procedure at Upgrade a Citrix ADC VPX instance on AWS.

  The recommended way to upgrade a Citrix ADC VPX instance is using the ADM service by following the procedure at Use jobs to upgrade Citrix ADC instances.

- **What is the HA failover time for Citrix ADC VPX in AWS?**

  - HA failover of Citrix ADC VPX within the AWS availability zone takes around 3 seconds.
  - HA failover of Citrix ADC VPX across AWS availability zones takes around 5 seconds.

- **What level of support is provided for Citrix ADC VPX marketplace subscription customers who provide the technical support PIN?**

  By default, the "Select for Software" service is provided to customers who provide the technical support PIN.

- **In High availability across different zones using Elastic IP deployment, do we need to create Multiple IPSets for each application?**

  Yes. If there are multiple applications with multiple VIPs mapped to multiple EIPs then multiple IPSets are required. Therefore during HA failover, all the primary VIP mappings of EIPs are changed to secondary (new primary) VIPs.

- **Why is INC mode enabled in high availability across different zone deployments?**

  HA pairs across availability zones are in different networks. For HA synchronization, network configuration must not be synchronized. This is achieved by enabling INC mode on HA pair.

- **Can HA node in one availability zone communicate with back-end servers in another availability zone, provided those availabilty zones are in same VPC?**

  Yes, subnets in different availability zones of the same VPC are reachable by adding an extra route pointing to the backend-server subnet via SNIP. For example, if the SNIP subnet of ADC in AZ1 is 192.168.3.0/24 and the backend-server subnet in AZ2 is 192.168.6.0/24, then a route must be added in the Citrix ADC appliance present in AZ1 as 192.168.6.0 255.255.255.0 192.168.3.1.

- **Can High availability across different zones using Elastic IP and High availability across different zones using Private IP deployments work together?**

  Yes, both the configurations can be applied on the same HA Pair.

- **In High availability across different zones using Private IP deployment, if there are multiple subnets with multiple route tables in a VPC, how does a secondary node in HA pair know about the route table to be checked during HA failover?**

  Secondary node is aware of the primary NICs and searches across all the route tables in a VPC.

- **What is the size of the /var partition when using the default image for VPX on AWS? How to increase the disk space?**

  The size of the root disk is limited to 20 GB to keep the disk image small.

  If you want to increase the /var/core/ or the /var/crash/ directory space, attach an extra disk. To increase the /var size, currently, you must attach an extra disk and create a symbolic link to /var, after copying the critical contents to the new disk.

- **How many packet engines are activated and allocated to vCPUs?**

  The packet engines (PEs) are limited by the number of licensed vCPUs. The Citrix ADC daemons are not pinned to any particular vCPU and might run on any of the non-PE vCPUs. According to AWS, the C5.9xlarge is a 36vCPU instance with 72 GB memory. With pooled licensing, the Citrix ADC VPX instance deploys with the maximum number of PEs. In this case, 19 PEs run on cores 1–19. However, ADC management processes run from CPUs 20–31.

- **How to decide the right AWS instance for ADC?**

  1. Understand your use case and requirements like throughput, PPS, SSL requirement, and average packet size.
  2. Choose the right ADC offering and licensing that meets your requirements, such as VPX bandwidth offerings or vCPU based licensing.
  3. Based on the chosen offering, decide on the AWS instance.

  Example:

  A 5 Gbps license enables 5 data packet engines. Hence, the vCPU requirement is 6 (5+1 for management). But 6 vCPU instance is not available. So an 8 vCPU is good enough to reach that throughput provided you choose a network that supports 5 Gbps bandwidth. For example, you must choose m5.2xlarge for a 5 Gbps bandwidth license to enable max PE allocation for 5 Gbps license. But if you use vCPU license that is not limited by throughput, you might get 5 Gbps throughput using the m5.xlarge instance itself.

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 4,750
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 4,750
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 4,750
m5.4xlarge	16	64	EBS-Only	Up to 10	4,750

- **Is three NICs-three subnets deployment mandatory for ADC in AWS**?

  `Three NICs-three subnets` is the recommended deployment, where each one for management, client and server network. This deployment gives better traffic isolation and VPX performance. Two NICs-two subnets, and one NIC-one subnet are the other available options. Citrix don't recommend multiple NICs sharing a subnet in AWS, such as two NICs–one subnet deployment. Because it might lead to networking issues such as asymmetric routing. For more information, see Best practices for configuring network interfaces in AWS.

Deploy a Citrix ADC VPX instance on Microsoft Azure

September 14, 2021

When you deploy a Citrix ADC VPX instance on Microsoft Azure Resource Manager (ARM), you can use both of the following feature sets to achieve your business needs:

- Azure cloud computing capabilities
- Citrix ADC load balancing and traffic management features

You can deploy Citrix ADC VPX instances on ARM either as standalone instances or as high availability pairs in active-standby modes.

You can deploy a Citrix ADC VPX instance on the Microsoft Azure in two ways:

- Through Azure Marketplace. The Citrix ADC VPX virtual appliance is available as an image in the Microsoft Azure Marketplace.
- Using the Citrix ADC Azure Resource Manager (ARM) json template available on GitHub.For more information, see the GitHub repository for Citrix NetScaler solution templates.

The Microsoft Azure stack is an integrated platform of hardware and software that delivers the Microsoft Azure public cloud services in a local data center to let organizations construct hybrid clouds. You can now deploy the Citrix ADC VPX instances on the Microsoft Azure stack.

Prerequisite

You need some prerequisite knowledge before deploying a Citrix VPX instance on Azure.

- Familiarity with Azure terminology and network details. For information, see Azure terminology.
- Knowledge of a Citrix ADC appliance. For detailed information the Citrix ADC appliance, see Citrix ADC
- Knowledge of Citrix ADC networking. See the Networking topic.

How a Citrix ADC VPX instance works on Azure

In an on-premises deployment, a Citrix ADC VPX instance requires at least three IP addresses:

- Management IP address, called NSIP address
- Subnet IP (SNIP) address for communicating with the server farm
- Virtual server IP (VIP) address for accepting client requests

For more information, see Network architecture for Citrix ADC VPX instances on Microsoft Azure.

> **Note**
>
> VPX virtual appliances can be deployed on any instance type that has two or more Intel VT-X cores and more than 2 GB memory. For more information on system requirements, see Citrix ADC VPX data sheet. Currently, Citrix ADC VPX instance supports only the Intel processors.

In an Azure deployment, you can provision a Citrix ADC VPX instance on Azure in three ways:

- Multi-NIC multi-IP architecture
- Single NIC multi IP architecture
- Single NIC single IP

Depending on your need, you can use any of these supported architecture types.

Multi-NIC multi-IP architecture

In this deployment type, you can have more than one network interfaces (NICs) attached to a VPX instance. Any NIC can have one or more IP configurations - static or dynamic public and private IP addresses assigned to it.

For more information, see the following use cases:

- Configure a high-availability setup with multiple IP addresses and NICs
- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands

> Note
>
> To avoid MAC moves and interface mutes on Azure environments, Citrix recommends you to create a VLAN per data interface (without tag) of ADC VPX instance and bind the primary IP of NIC in Azure. For more information, see CTX224626 article.

Single NIC multi IP architecture

In this deployment type, one network interfaces (NIC) associated with multiple IP configurations - static or dynamic public and private IP addresses assigned to it.
For more information, see the following use cases:

- Configure multiple IP addresses for a Citrix ADC VPX standalone instance
- Configure multiple IP addresses for a Citrix ADC VPX standalone instance by using PowerShell commands

Single NIC single IP

In this deployment type, one network interfaces (NIC) associated with a single IP address, which is used to perform the functions of NSIP, SNIP, and VIP.

For more information, see the following use case:

- Configure a Citrix ADC VPX standalone instance

> **Note**
>
> The single IP mode is available only in Azure deployments. This mode is not available for a Citrix ADC VPX instance on your premises, on AWS, or in other type of deployment.

Citrix ADC VPX licensing

A Citrix ADC VPX instance on Azure requires a license. The following licensing options are available for Citrix ADC VPX instances running on Azure.

- **Subscription-based licensing**: Citrix ADC VPX appliances are available as paid instances on Azure Marketplace. Subscription based licensing is a pay-as-you-go option. Users are charged hourly. The following VPX models and license types are available on Azure Marketplace.

VPX model	License type	Recommended instance
VPX10	Standard, Advanced, Premium	Standard_D2s_v4
VPX200	Standard, Advanced, Premium	Standard_D2s_v4
VPX1000*	Standard, Advanced, Premium	Standard_D4s_v4
VPX3000*	Standard, Advanced, Premium	Standard_D8s_v4

> *: For VPX 1000 and VPX 3000 models, you must enable Accelerated networking on Citrix ADC VPX instances to get the desired performance.

Citrix provides technical support for subscription-based license instances. To file a support case, see Support for Citrix ADC on Azure – Subscription license with hourly price.

- **Bring your own license (BYOL)**: If you bring your own license (BYOL), see the VPX Licensing Guide at http://support.citrix.com/article/CTX122426. You have to:
  - Use the licensing portal within Citrix website to generate a valid license.
  - Upload the license to the instance.

VPX model	License type	Recommended instance
VPX10	Standard, Advanced, Premium	Standard_D2s_v4
VPX200	Standard, Advanced, Premium	Standard_D2s_v4
VPX1000*	Standard, Advanced, Premium	Standard_D4s_v4
VPX3000*	Standard, Advanced, Premium	Standard_D8s_v4
VPX5000*	Standard, Advanced, Premium	Standard_D8s_v4
VPX8000*	Standard, Advanced, Premium	Standard_D8s_v4
VPX10000*	Standard, Advanced, Premium	Standard_D16s_v4

> *: From VPX 1000 to VPX 10000 models, you must enable Accelerated networking on Citrix ADC VPX instances to get the desired performance.

- **Citrix ADC VPX Check-In/Check-Out licensing**: For more information, see Citrix ADC VPX Check-In/Check-Out Licensing.

> **Note**
>
> In an Azure stack environment, **BYOL** is the only available licensing option.

Starting with NetScaler release 12.0 56.20, VPX Express for on-premises and cloud deployments does not require a license file. For more information on Citrix ADC VPX Express, see the "Citrix ADC VPX Express license" section in Citrix ADC licensing overview.

> **Note**
>
> Regardless of the subscription-based hourly license bought from Azure Marketplace, in rare cases, the Citrix ADC VPX instance deployed on Azure might come up with a default Citrix ADC license. This happens due to issues with the Azure Instance Metadata Service (IMDS).

> Do a warm restart before making any configuration change on the Citrix ADC VPX instance, to enable the correct Citrix ADC VPX license.

Limitations

Running the Citrix ADC VPX load balancing solution on ARM imposes the following limitations:

- The Azure architecture does not accommodate support for the following NetScaler features:
  - IPv6
  - Gratuitous ARP (GARP)
  - L2 Mode
  - Tagged VLAN
  - Dynamic Routing
  - virtual MAC
  - USIP
  - Jumbo Frames
  - Clustering

  > Note
  >
  > With the Citrix Application Delivery Management (ADM) Autoscale feature (cloud deployment), the ADC instances support clustering on all licenses. For information, see Autoscaling of Citrix ADC VPX in Microsoft Azure using Citrix ADM.

- If you expect that you might have to shut down and temporarily deallocate the Citrix ADC VPX virtual machine at any time, assign a static Internal IP address while creating the virtual machine. If you do not assign a static internal IP address, Azure might assign the virtual machine a different IP address each time it restarts, and the virtual machine might become inaccessible.
- In an Azure deployment, only the following Citrix ADC VPX models are supported: VPX 10, VPX 200, VPX 1000, and VPX 3000. For for information, see the Citrix ADC VPX Data Sheet.

  If you use a Citrix ADC VPX instance with a model number higher than VPX 3000, the network throughput might not be the same as specified by the instance's license. However, other features such as SSL throughput and SSL transactions per second might improve.
- The "deployment ID" that is generated by Azure during virtual machine provisioning is not visible to the user in ARM. You cannot use the deployment ID to deploy Citrix ADC VPX appliance on ARM.
- The Citrix ADC VPX instance supports 20 Mb/s throughput and standard edition features when it's initialized.
- The Citrix ADC VPX instances on Azure with accelerated networking enabled, provides better performance. Azure accelerated networking is supported on Citrix ADC VPX instances from release

13.0 build 76.x onwards. To enable accelerated networking on ADC VPX, Citrix recommends you to use an Azure instance type which supports accelerated networking.

- For a XenApp and XenDesktop deployment, a VPN virtual server on a VPX instance can be configured in the following modes:
  - Basic mode, where the `ICAOnly` VPN virtual server parameter is set to ON. The Basic mode works fully on an unlicensed Citrix ADC VPX instance.
  - SmartAccess mode, where the `ICAOnly` VPN virtual server parameter is set to OFF. The SmartAccess mode works for only five Citrix ADC AAA session users on an unlicensed Citrix ADC VPX instance.

> **Note**
>
> To configure the SmartControl feature, you must apply a Premium license to the Citrix ADC VPX instance.

Azure terminology

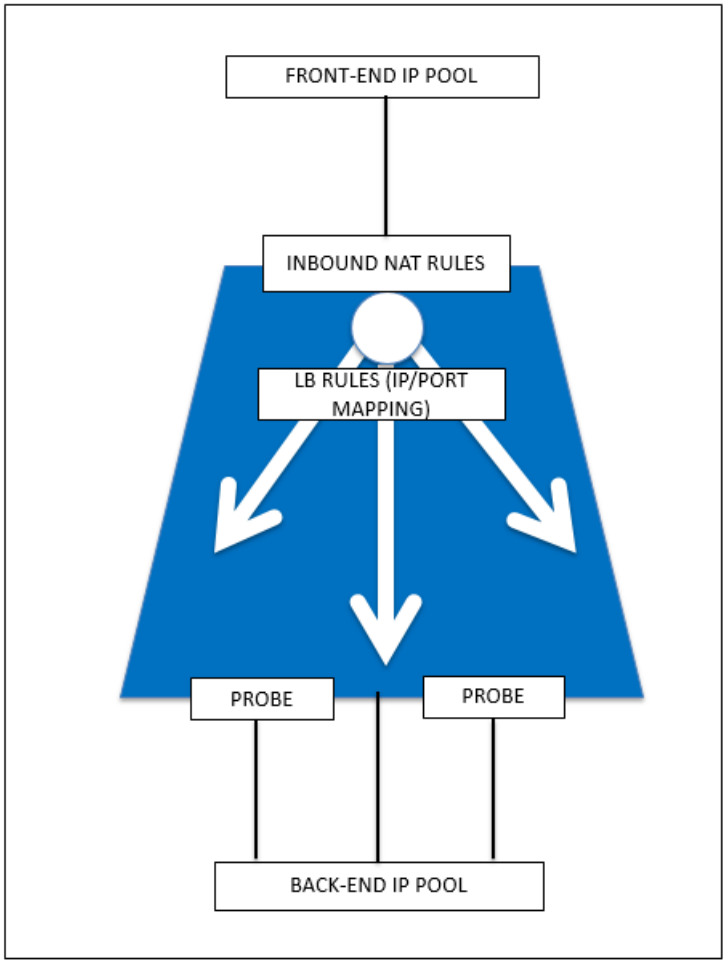
September 14, 2021

Some of the Azure terms that are used in the Citrix ADC VPX Azure documentation are listed below.

1. Azure Load Balancer – Azure load balancer is a resource that distributes incoming traffic among computers in a network. Traffic is distributed among virtual machines defined in a load-balancer set. A load balancer can be external or internet-facing, or it can be internal.
2. Azure Resource Manager (ARM) – ARM is the new management framework for services in Azure. Azure Load Balancer is managed using ARM-based APIs and tools.
3. Back-End Address Pool – These are IP addresses associated with the virtual machine NIC (NIC) to which load will be distributed.
4. BLOB - Binary Large Object – Any binary object like a file or an image that can be stored in Azure storage.
5. Front-End IP Configuration – An Azure Load balancer can include one or more front-end IP addresses, also known as a virtual IPs (VIPs). These IP addresses serve as ingress for the traffic.
6. Instance Level Public IP (ILPIP) – An ILPIP is a public IP address that you can assign directly to your virtual machine or role instance, rather than to the cloud service that your virtual machine or role instance resides in. This does not take the place of the VIP (virtual IP) that is assigned to your cloud service. Rather, it's an extra IP address that you can use to connect directly to your virtual machine or role instance.

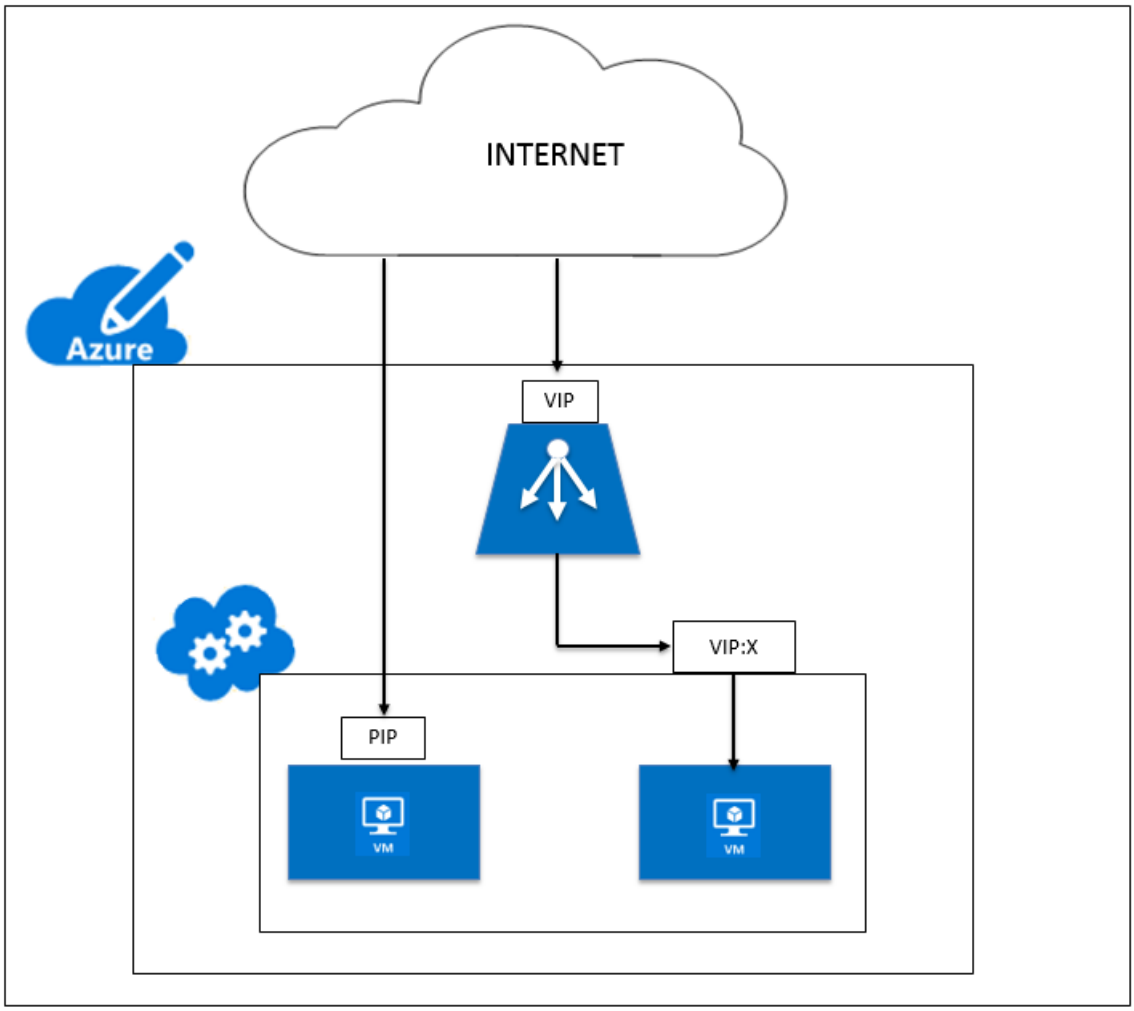
   Note: In the past, an ILPIP was referred to as a PIP, which stands for public IP.

7. Inbound NAT Rules – This contains rules mapping a public port on the load balancer to a port for a specific virtual machine in the back end address pool.
8. IP-Config - It can be defined as an IP address pair (public IP and private IP) associated with an individual NIC. In an IP-Config, the public IP address can be NULL. Each NIC can have multiple IP-Config associated with it, which can be up to 255.
9. Load Balancing Rules – A rule property that maps a given front-end IP and port combination to a set of back-end IP addresses and port combination. With a single definition of a load balancer resource, you can define multiple load balancing rules, each rule reflecting a combination of a front end IP and port and back end IP and port associated with virtual machines.



10. Network security group – Contains a list of Access Control List (ACL) rules that allow or deny network traffic to your virtual machine instances in a virtual network. NSGs can be associated with either subnets or individual virtual machine instances within that subnet. When a network security group is associated with a subnet, the ACL rules apply to all the virtual machine instances in that subnet. In addition, traffic to an individual virtual machine can be restricted further by associating a network security group directly to that virtual machine.

11. Private IP addresses – Used for communication within an Azure virtual network, and your on-premises network when you use a VPN gateway to extend your network to Azure. Private IP addresses allow Azure resources to communicate with other resources in a virtual network or an on-premises network through a VPN gateway or ExpressRoute circuit, without using an Internet-reachable IP address. In the Azure Resource Manager deployment model, a private IP address is associated with the following types of Azure resources – virtual machines, internal load balancers (ILBs), and application gateways.
12. Probes – This contains health probes used to check availability of virtual machines instances in the back end address pool. If a particular virtual machine does not respond to health probes for some time, then it is taken out of traffic serving. Probes enable you to keep track of the health of virtual instances. If a health probe fails, the virtual instance will be taken out of rotation automatically.
13. Public IP Addresses (PIP) – PIP is used for communication with the Internet, including Azure public-facing services and is associated with virtual machines, Internet-facing load balancers, VPN gateways, and application gateways.
14. Region - An area within a geography that does not cross national borders and that contains one or more data centers. Pricing, regional services, and offer types are exposed at the region level. A region is typically paired with another region, which can be up to several hundred miles away, to form a regional pair. Regional pairs can be used as a mechanism for disaster recovery and high availability scenarios. Also referred to generally as location.
15. Resource Group - A container in Resource Manager holds related resources for an application. The resource group can include all of the resources for an application, or only those resources that are logically grouped together
16. Storage Account – An Azure storage account gives you access to the Azure blob, queue, table, and file services in Azure Storage. Your storage account provides the unique namespace for your Azure storage data objects.
17. Virtual Machine – The software implementation of a physical computer that runs an operating system. Multiple virtual machines can run simultaneously on the same hardware. In Azure, virtual machines are available in a variety of sizes.
18. Virtual Network - An Azure virtual network is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network. You can also further segment your VNet into subnets and launch Azure IaaS virtual machines and cloud services (PaaS role instances). Additionally, you can connect the virtual network to your on-premises network using one of the connectivity options available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.



Network architecture for Citrix ADC VPX instances on Microsoft Azure

September 14, 2021

In Azure Resource Manager (ARM), a Citrix ADC VPX virtual machine (VM) resides in a virtual network. A single network interface can be created in a given subnet of the Virtual Network and can be attached to the VPX instance. You can filter network traffic to and from a VPX instance in an Azure virtual network with a network security group. A network security group contains security rules that allow or deny inbound network traffic to or outbound network traffic from a VPX instance. For more information, see Security groups.

Network security group filters the requests to the Citrix ADC VPX instance, and the VPX instance sends them to the servers. The response from a server follows the same path in reverse. The Network security group can be configured to filter a single VPX VM, or, with subnets and virtual networks, can filter

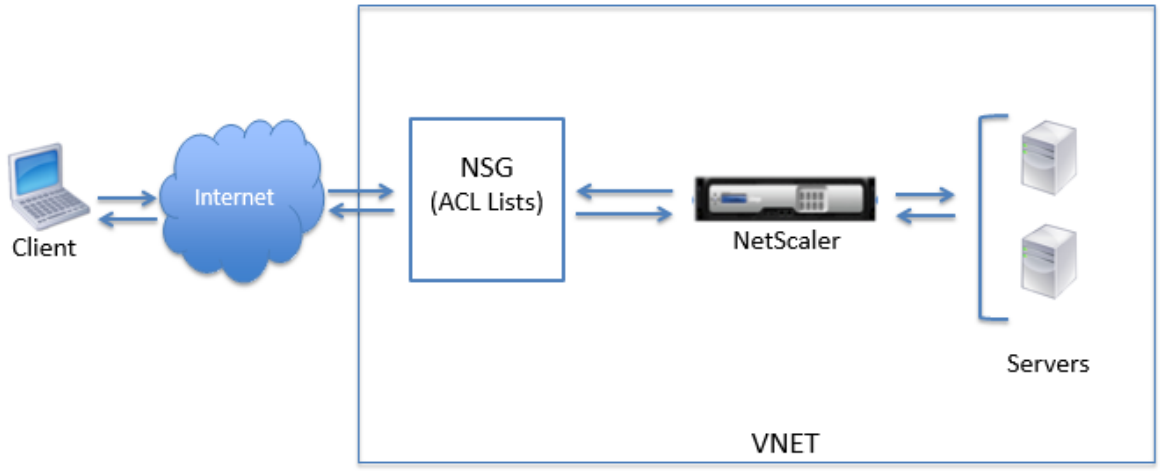
traffic in deployment of multiple VPX instances.

The NIC contains network configuration details such as the virtual network, subnets, internal IP address, and Public IP address.

While on ARM, it is good to know the following IP addresses that are used to access the VMs deployed with a single NIC and a single IP address:

- Public IP (PIP) address is the internet-facing IP address configured directly on the virtual NIC of the NetScaler VM. This allows you to directly access a VM from the external network.
- Citrix ADC IP (also known as NSIP) address is the internal IP address configured on the VM. It is non-routable.
- Virtual IP address (VIP) is configured by using the NSIP and a port number. Clients access NetScaler services through the PIP address, and when the request reaches the NIC of the NetScaler VPX VM or the Azure load balancer, the VIP gets translated to internal IP (NSIP) and internal port number.
- Internal IP address is the private internal IP address of the VM from the virtual network's address space pool. This IP address cannot be reached from the external network. This IP address is by default dynamic unless you set it to static. Traffic from the internet is routed to this address according to the rules created on the network security group. The network security group integrates with the NIC to selectively send the right type of traffic to the right port on the NIC, which depends on the services configured on the VM.

The following figure shows how traffic flows from a client to a server through a NetScaler VPX instance provisioned in ARM.

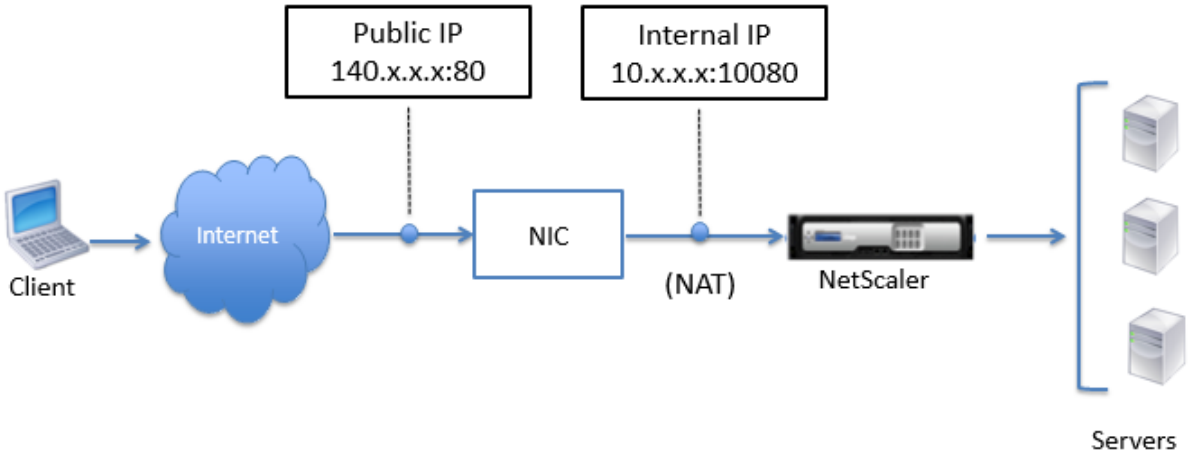


Traffic flow through network address translation

You can also request a public IP (PIP) address for your Citrix ADC VPX instance (instance level). If you use this direct PIP at the VM level, you need not define inbound and outbound rules to intercept the

network traffic. The incoming request from the Internet is received on the VM directly. Azure performs network address translation (NAT) and forwards the traffic to the internal IP address of the VPX instance.

The following figure shows how Azure performs network address translation to map the NetScaler internal IP address.



In this example, the Public IP assigned to the network security group is 140.x.x.x and the internal IP address is 10.x.x.x. When the inbound and outbound rules are defined, public HTTP port 80 is defined as the port on which the client requests are received, and a corresponding private port, 10080, is defined as the port on which the Citrix ADC VPX instance listens. The client request is received on the Public IP address (140.x.x.x). Azure performs network address translation to map the PIP to the internal IP address 10.x.x.x on port 10080, and forwards the client request.

> **Note**
>
> Citrix ADC VPX VMs in high availability are controlled by external or internal load balancers that have inbound rules defined on them to control the load balancing traffic. The external traffic is first intercepted by these load balancers and the traffic is diverted according to the load balancing rules configured, which has back-end pools, NAT rules, and health probes defined on the load balancers.

Port usage guidelines

You can configure more inbound and outbound rules n network security group while creating the Citrix ADC VPX instance or after the virtual machine is provisioned. Each inbound and outbound rule is associated with a public port and a private port.

Before configuring network security group rules, note the following guidelines regarding the port numbers you can use:

1. The Citrix ADC VPX instance reserves the following ports. You cannot define these as private ports when using the Public IP address for requests from the internet.

   Ports 21, 22, 80, 443, 8080, 67, 161, 179, 500, 520, 3003, 3008, 3009, 3010, 3011, 4001, 5061, 9000, 7000.

   However, if you want internet-facing services such as the VIP to use a standard port (for example, port 443) you have to create port mapping by using the network security group. The standard port is then mapped to a different port that is configured on the NetScaler for this VIP service.

   For example, a VIP service might be running on port 8443 on the VPX instance but be mapped to public port 443. So, when the user accesses port 443 through the Public IP, the request is directed to private port 8443.

2. Public IP address does not support protocols in which port mapping is opened dynamically, such as passive FTP or ALG.
3. High availability does not work for traffic that uses a public IP address (PIP) associated with a VPX instance, instead of a PIP configured on the Azure load balancer.

> **Note**
>
> In Azure Resource Manager, a Citrix ADC VPX instance is associated with two IP addresses - a public IP address (PIP) and an internal IP address. While the external traffic connects to the PIP, the internal IP address or the NSIP is non-routable. To configure VIP in VPX, use the internal IP address and any of the free ports available. Do not use the PIP to configure VIP.

Configure a Citrix ADC VPX standalone instance

September 14, 2021

You can provision a single Citrix ADC VPX instance in Azure Resource Manager (ARM) portal in a standalone mode by creating the virtual machine and configuring other resources.

Before you begin

Ensure that you have the following:

- A Microsoft Azure user account
- Access to Microsoft Azure Resource Manager
- Microsoft Azure SDK
- Microsoft Azure PowerShell

On the Microsoft Azure Portal page, log on to the Azure Resource Manager portal by providing your user name and password.

> **Note**
>
> In ARM portal, clicking an option in one pane opens a new pane to the right. Navigate from one pane to another to configure your device.

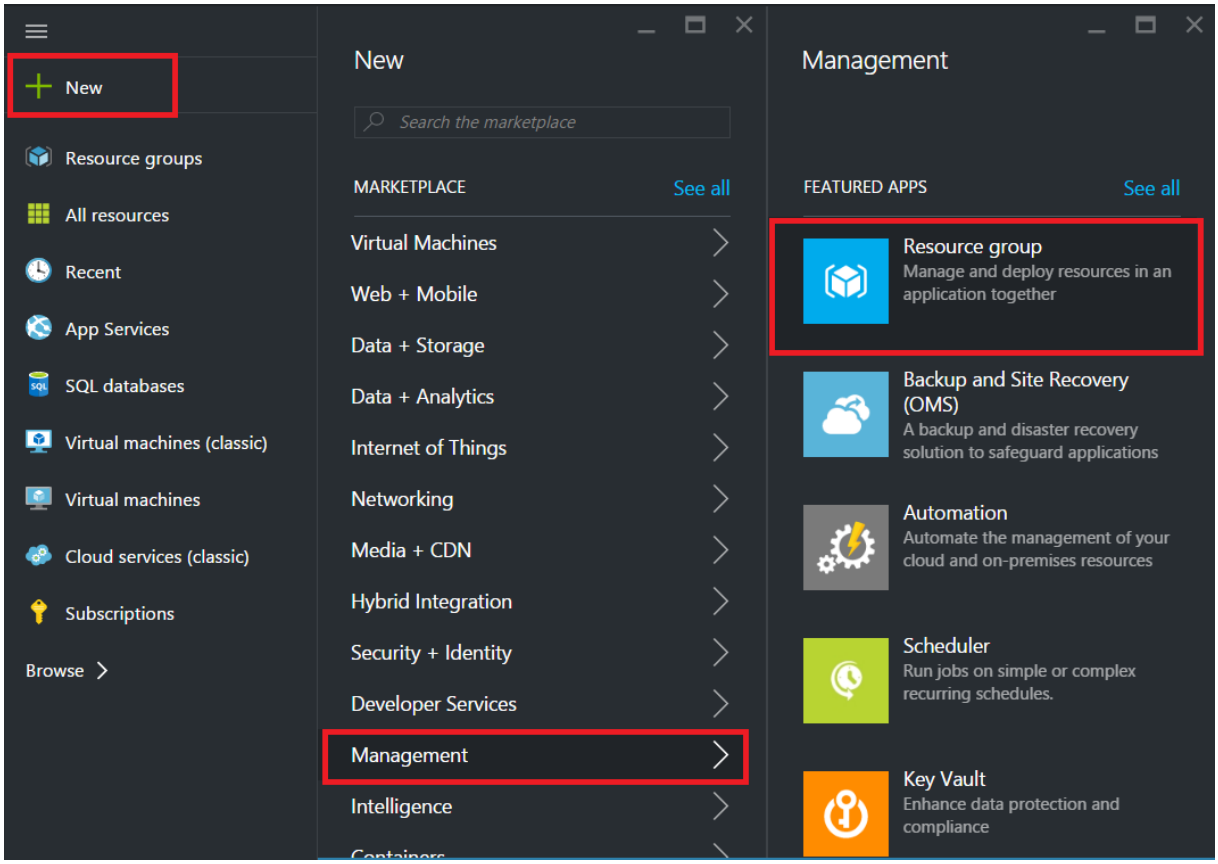
Summary of configuration steps

1. Configure a resource group
2. Configure a network security group
3. Configure virtual network and its subnets
4. Configure a storage account
5. Configure an availability set
6. Configure a Citrix ADC VPX instance.

Configure a resource group

Create a new resource group that is a container for all your resources. Use the resource group to deploy, manage, and monitor your resources as a group.

1. Click **New** > **Management** > **Resource group**.
2. In the **Resource group** pane, enter the following details:
   - Resource group name
   - Resource group location
3. Click **Create**.

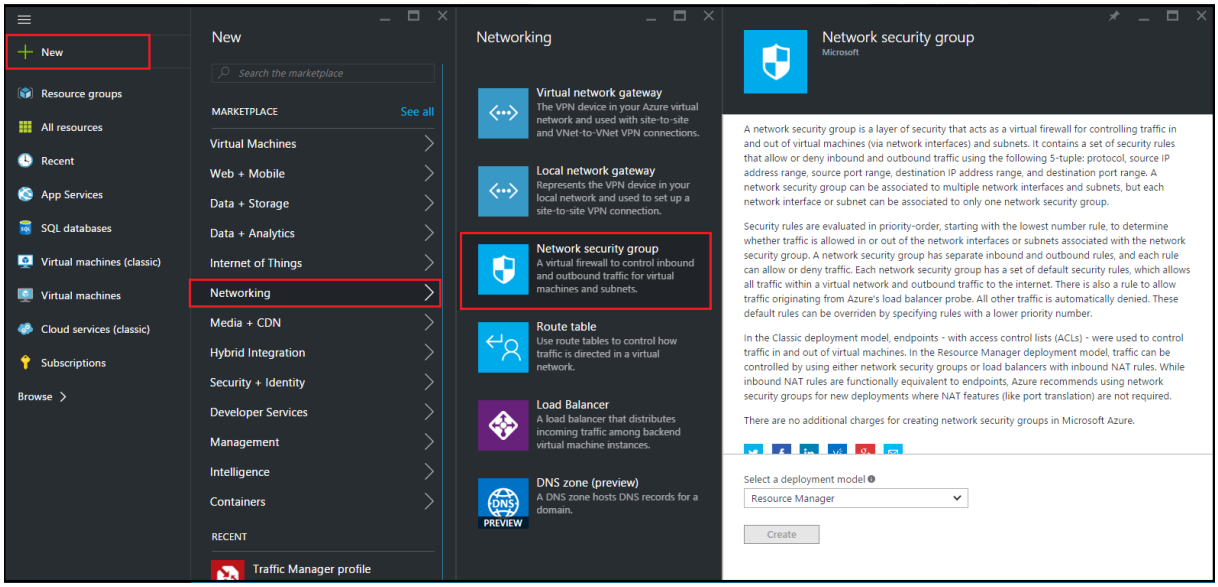


Configure a network security group

Create a network security group to assign inbound and outbound rules to control the incoming and outgoing traffic within the virtual network. Network security group allows you to define security rules for a single virtual machine and also to define security rules for a virtual network subnet.

1. Click **New** > **Networking** > **Network security group**.
2. In the **Create network security group** pane, enter the following details, and then click **Create**.
    - Name - type a name for the security group
    - Resource group - select the resource group from the drop-down list

> **Note**
>
> Ensure that you have selected the correct location. The list of resources that appear in the drop-down list is different for different locations.

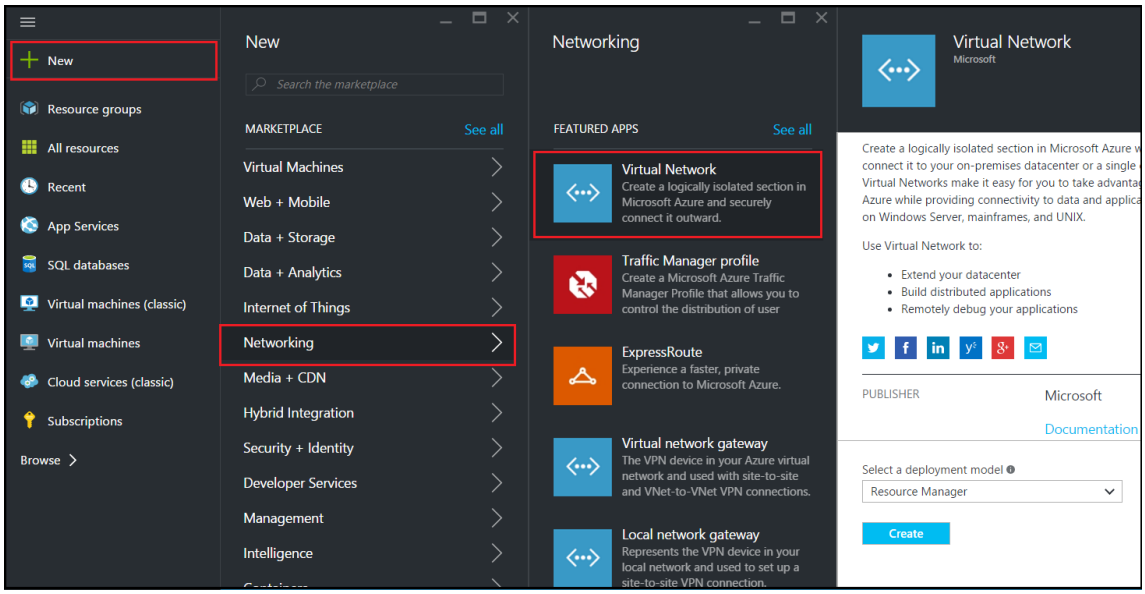


Configure a virtual network and subnets

Virtual networks in ARM provide a layer of security and isolation to your services. VMs and services that are part of the same virtual network can access each other.

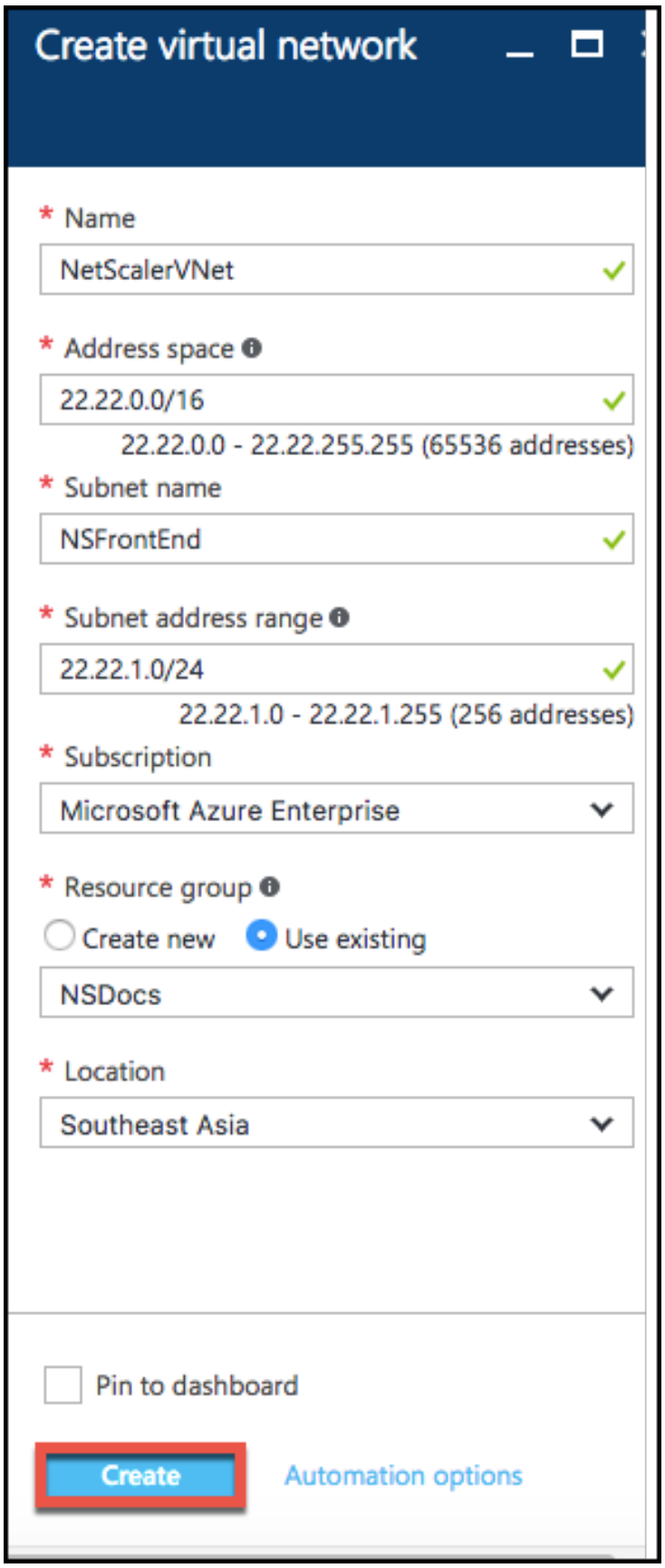
For these steps to create a virtual network and subnets.

1. Click **New** > **Networking** > **Virtual Network**.
2. In the **Virtual Network** pane, ensure the deployment mode is **Resource Manager** and click **Create**.



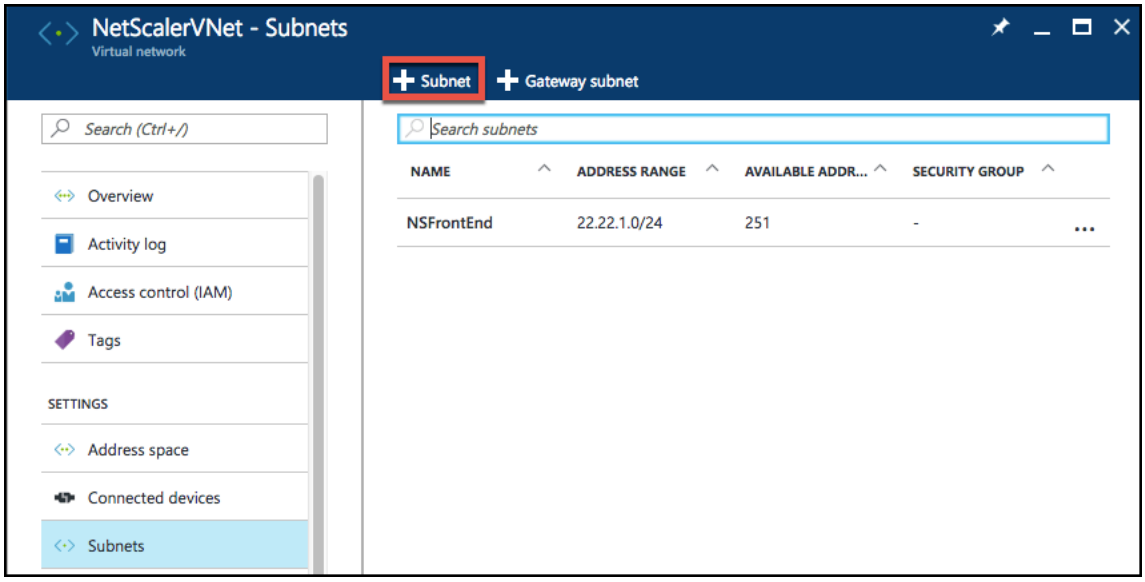
3. In the **Create virtual network** pane, enter the following values, and then click **Create**.
    - Name of the virtual network

- Address space - type the reserved IP address block for the virtual network
- Subnet - type the name of the first subnet (you create the second subnet later in this step)
- Subnet address range - type the reserved IP address block of the subnet
- Resource group - select the resource group created earlier from the drop-down list

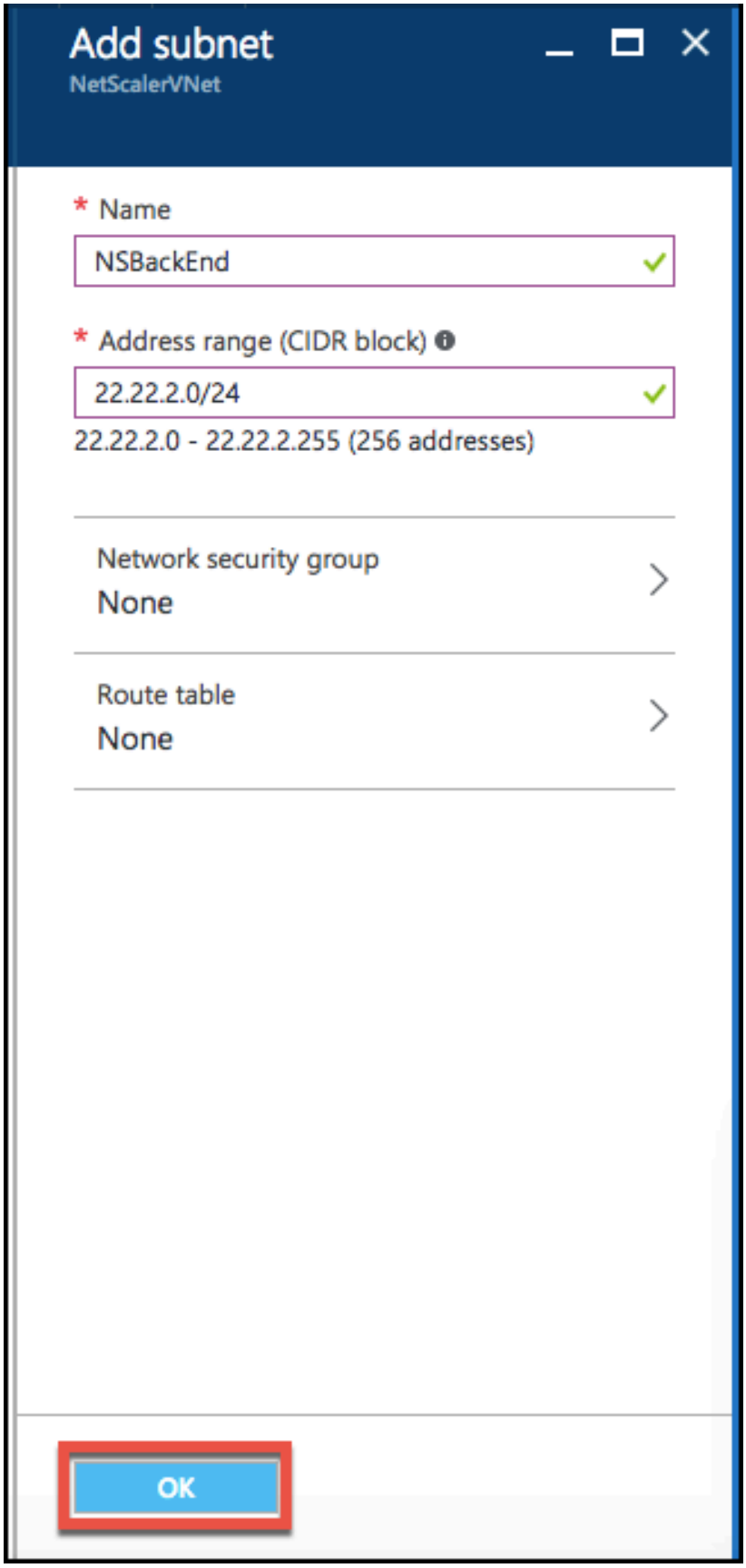


Configure the second subnet

1. Select the newly created virtual network from **All resources** pane and in the **Settings** pane, click **Subnets**.



2. Click **+Subnet** and create the second subnet by entering the following details.
   - Name of the second subnet
   - Address range - type the reserved IP address block of the second subnet
   - Network security group - select the network security group from the drop-down list
3. Click **Create**.

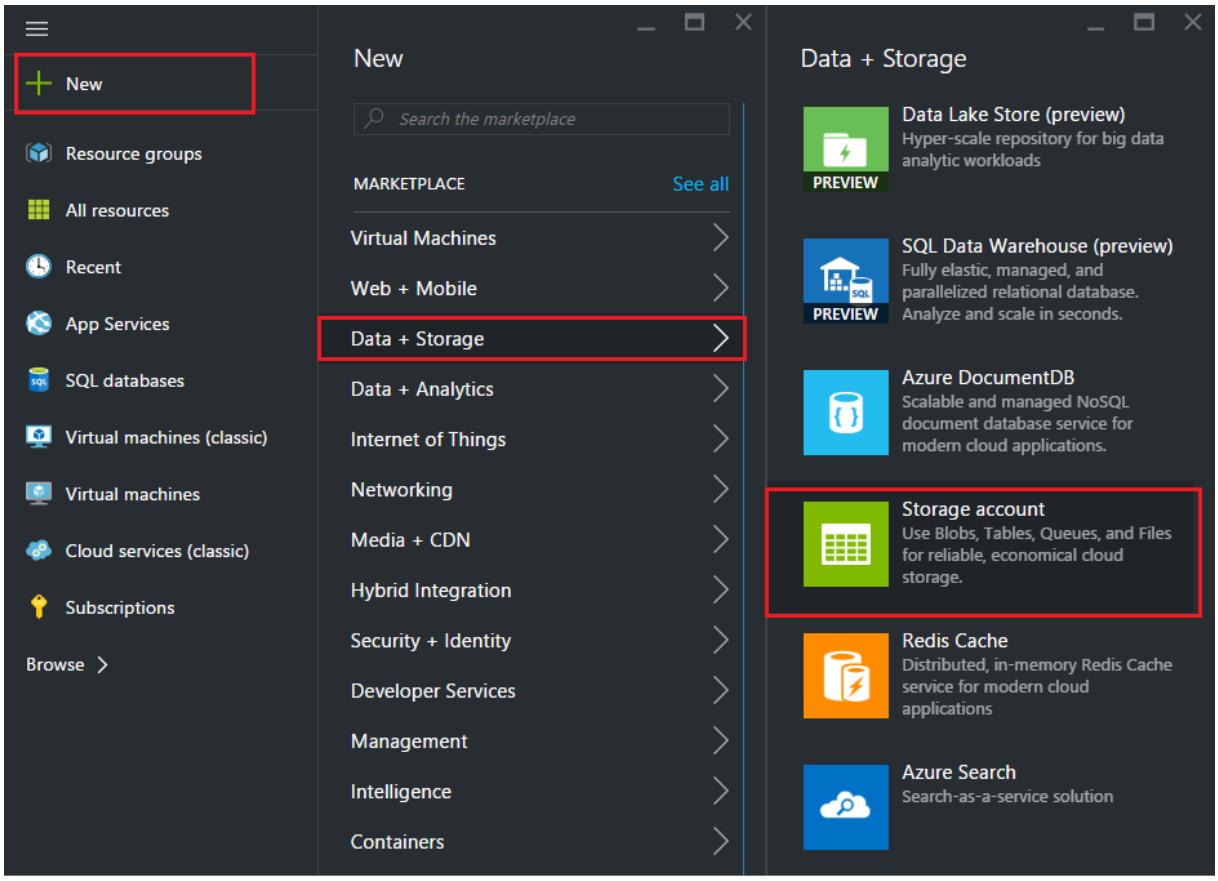


Configure a storage account

The ARM IaaS infrastructure storage includes all services where we can store data in the form of blobs, tables, queues, and files. You can also create applications using these forms of storage data in ARM.

Create a storage account to store all your data.

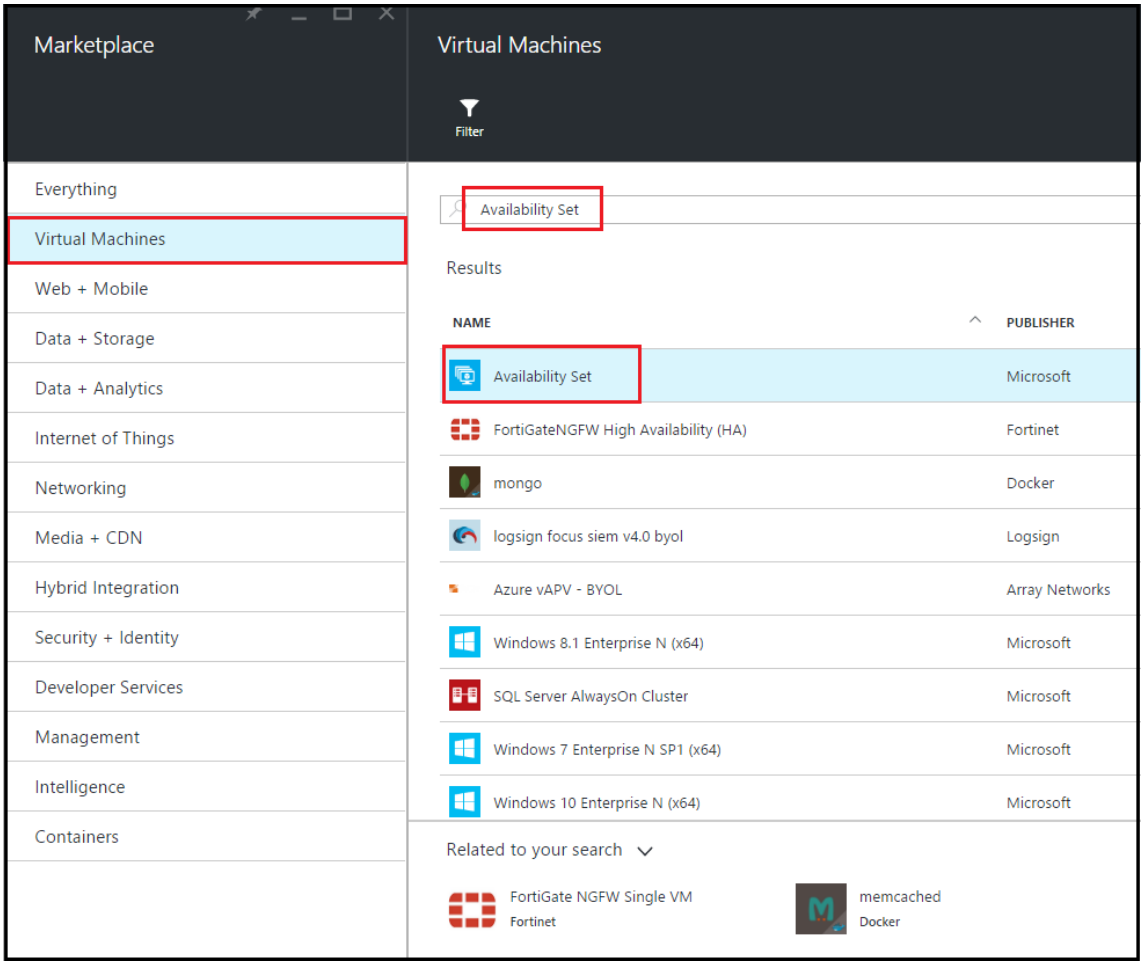
1. Click **+New** > **Data + Storage** > **Storage account**.
2. In the **Create storage account** pane, enter the following details:
    - Name of the account
    - Deployment mode - make sure to select **Resource Manager**
    - Account kind - select **General purpose** from the drop-down list
    - Replication - select **Locally redundant storage** from the drop-down list
    - Resource group - select the newly created resource group from the drop-down list
3. Click **Create**.



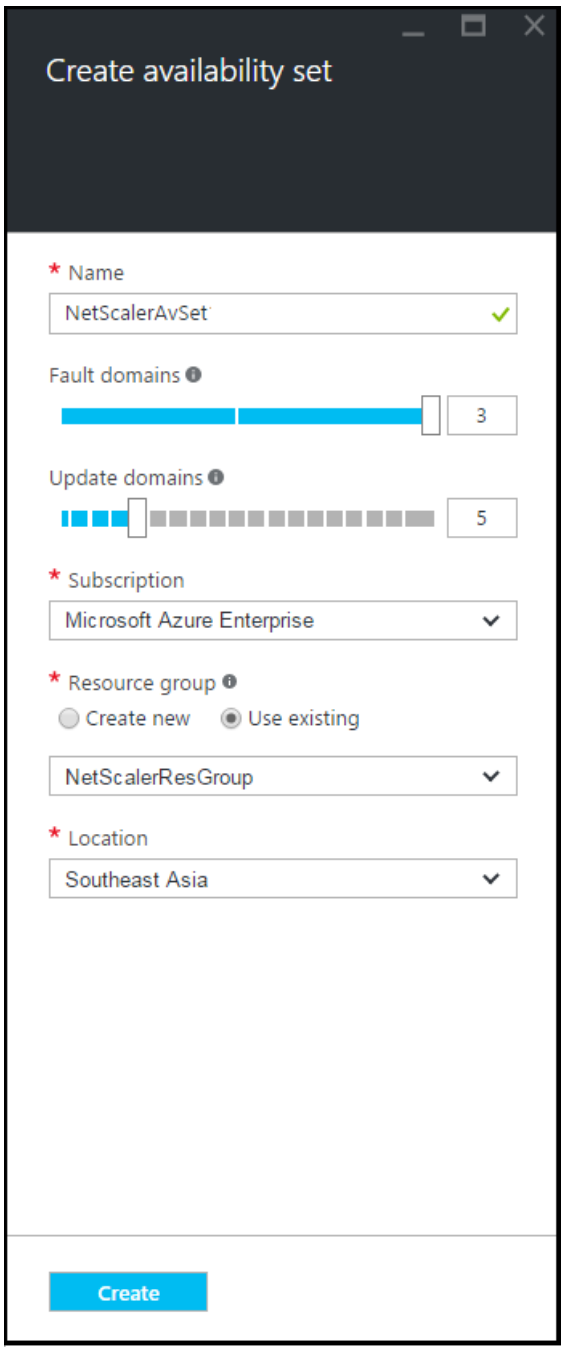
Configure an availability set

An availability set guarantee that at least one VM is kept up and running in case of planned or un-planned maintenance. Two or more VMs under the same 'availability set' are placed on different fault domains to achieve redundant services.

1. Click **+New**.
2. Click **See all** in the MARKETPLACE pane and click **Virtual Machines**.
3. Search for availability set, and then select **Availability set** entity from the list displayed.



4. Click **Create,** and in the **Create availability set** pane, enter the following details:
   - Name of the set
   - Resource group - select the newly created resource group from the drop-down list
5. Click **Create**.



Configure a Citrix ADC VPX instance

Create an instance of Citrix ADC VPX in the virtual network. Obtain the Citrix ADC VPX image from the Azure Marketplace, and then use the Azure Resource Manager portal to create a Citrix ADC VPX instance.

Before you begin creating the Citrix ADC VPX instance, make sure that you have created a virtual network with required subnets in which the instance resides. You can create virtual networks during

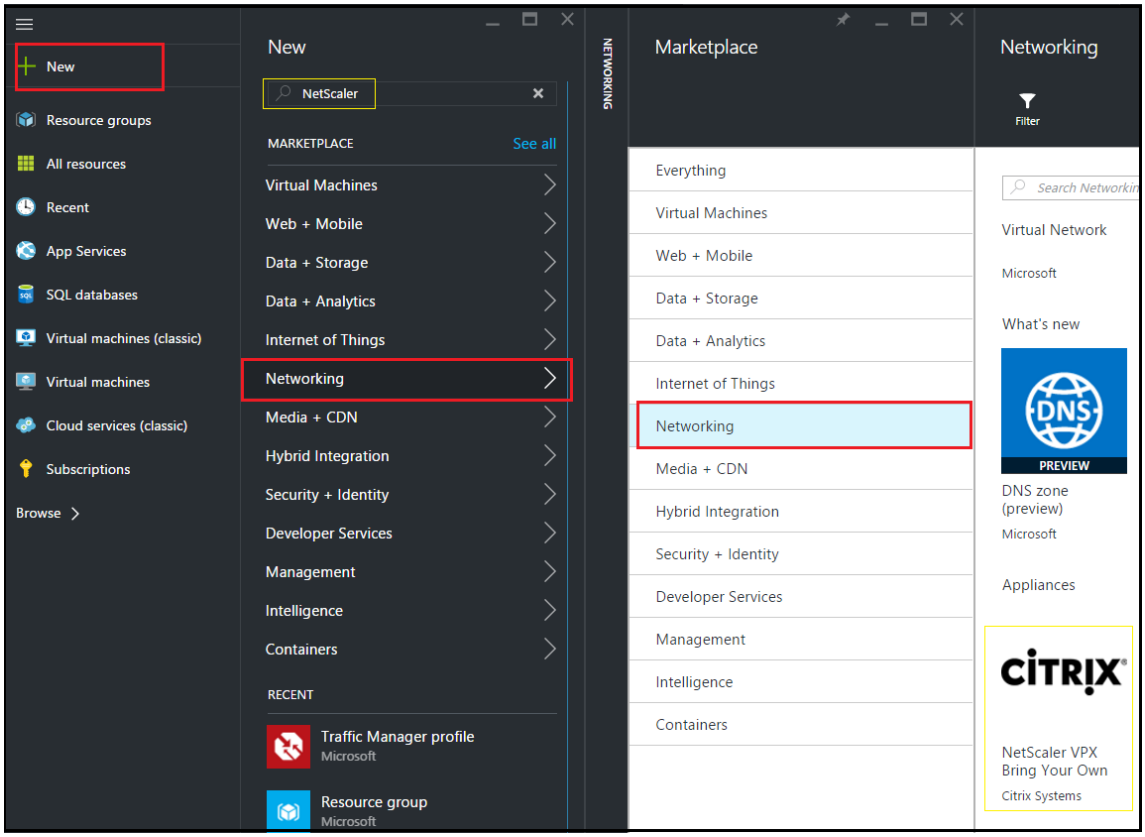
VM provisioning, but without the flexibility to create different subnets. For information about creating virtual networks, see http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/.

Optionally, configure DNS server and VPN connectivity that allows a virtual machine to access internet resources.

> **Note**
>
> Citrix recommends that you create resource group, network security group, virtual network, and other entities before you provision the Citrix ADC VPX VM, so that the network information is available during provisioning.

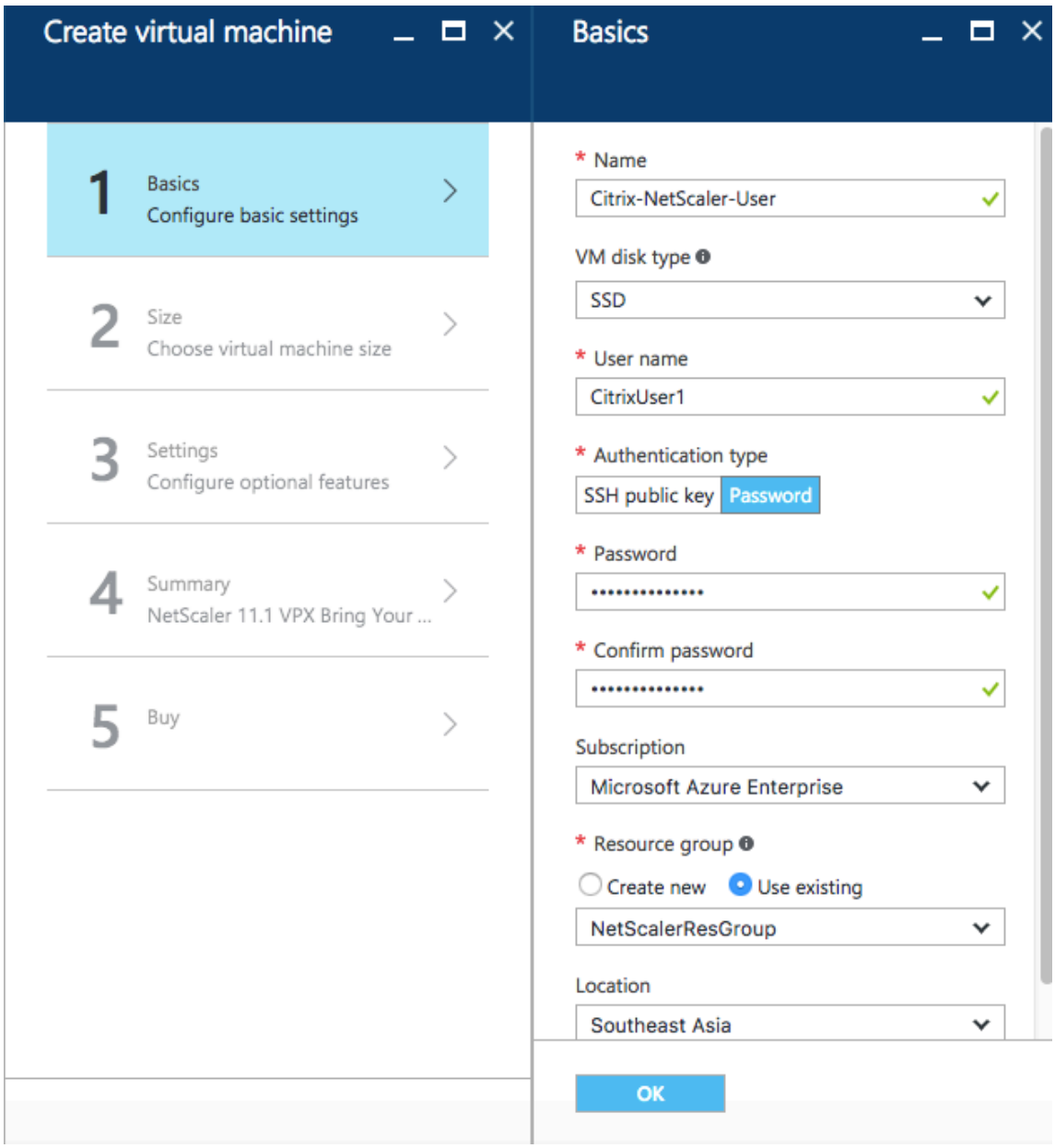
1. Click **+New** > **Networking**.
2. Click **See All** and in the Networking pane, click **Citrix ADC 13.0**.
3. Select **Citrix ADC 13.0 VPX Bring Your Own License** from the list of software plans.

   As a quick way to find any entity on ARM portal, you can also type the name of the entity in the Azure Marketplace search box and press <Enter>. Type NetScaler in the search box to find the Citrix NetScaler images.



> **Note**
>
> Ensure to select the latest image. Your Citrix NetScaler image might have the release number in the name.

4. On the **Citrix ADC VPX Bring Your Own License** page, from the drop-down list, select **Resource Manager** and click **Create**.



5. In the **Create virtual machine** pane, specify the required values in each section to create a virtual machine. Click **OK** in each section to save your configuration.

Basic:

- Name - specify a name for the Citrix ADC VPX instance

- VM disk type - select SSD (default value) or HDD from the drop-down menu
- User name and Password - specify a user name and password to access the resources in the resource group that you have created
- Authentication Type - select SSH Public Key or Password
- Resource group - select the resource group you have created from the drop-down list

You can create a resource group here, but Citrix recommends that you create a resource group from Resource groups in Azure Resource Manager and then select the group from the drop-down list.

> **Note**
>
> In an Azure stack environment, in addition to the basic parameters, specify the following parameters:
>
> - Azure stack domain
> - Azure stack tenant (Optional)
> - Azure client (Optional)
> - Azure client secret (Optional)

Size:

Depending on the VM disk type, SDD, or HDD, you selected in Basic settings, the disk sizes are displayed.

- Select a disk size according to your requirement and click **Select**.

Settings:

- Select the default (Standard) disk type
- Storage account - select the storage account
- Virtual network - select the virtual network
- Subnet - set the subnet address
- Public IP address - select the type of IP address assignment
- Network security group - select the security group that you have created. Ensure that inbound and outbound rules are configured in the security group.
- Availability Set - select the availability set from the drop-down menu box

Summary:

The configuration settings are validated and the Summary page displays the result of the validation. If the validation fails, the Summary page displays the reason of the failure. Go back to the particular section and make changes as required. If the validation passes, click **OK**.

Buy:

Review the offer details and legal terms on the Purchase page and click **Purchase**.

For high availability deployment, create two independent instances of Citrix ADC VPX in the same availability set and in the same resource group to deploy them in active-standby configuration.

Configure multiple IP addresses for a Citrix ADC VPX standalone instance

September 14, 2021

This section explains how to configure a standalone Citrix ADC VPX instance with multiple IP addresses, in Azure Resource Manager (ARM). The VPX instance can have one or more NIC attached to it, and each NIC can have one or more static or dynamic public and private IP addresses assigned to it. You can assign multiple IP addresses as NSIP, VIP, SNIP, and so on.

For more information, see the Azure documentation Assign multiple IP addresses to virtual machines using the Azure portal.

If you want to use PowerShell commands, see Configuring multiple IP addresses for a Citrix ADC VPX instance in standalone mode by using PowerShell commands.

Use case

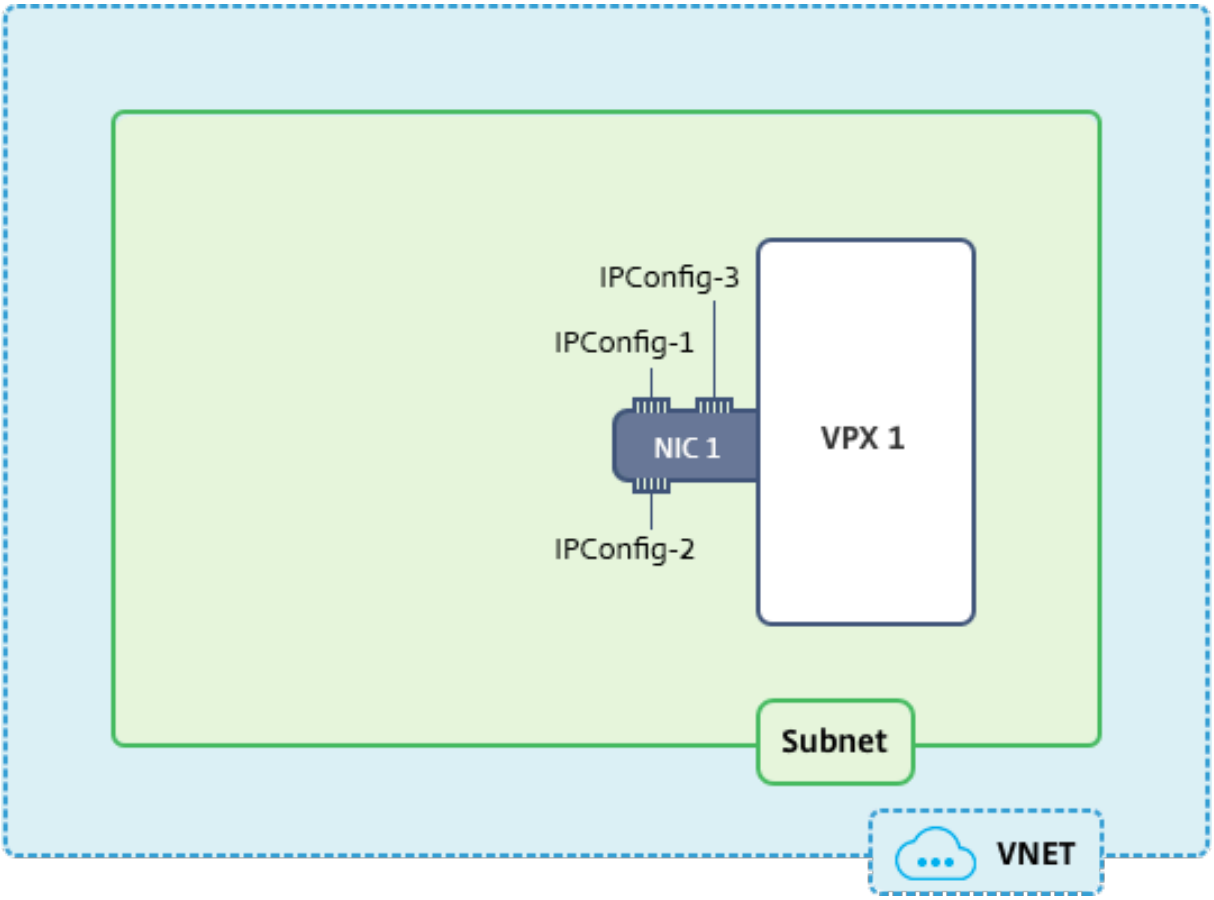
In this use case, a standalone Citrix ADC VPX appliance is configured with a single NIC that is connected to a virtual network (VNET). The NIC is associated with three IP configurations (ipconfig), each server a different purpose - as shown in the table.

IP config	Associated with	Purpose
ipconfig1	Static public IP address; static private IP address	Serves management traffic
ipconfig2	Static public IP address; static private address	Serves client-side traffic
ipconfig3	Static private IP address	Communicates with back-end servers

> **Note**
>
> `IPConfig-3` is not associated with any public IP address.

Diagram: Topology

Here is the visual representation of the use case.



> **Note**
>
> In a multi-NIC, multi-IP Azure Citrix ADC VPX deployment, the private IP associated with the primary (first) `IPConfig` of the primary (first) NIC is automatically added as the management NSIP of the appliance. The remaining private IP addresses associated with `IPConfigs` need to be added in the VPX instance as a VIP or SNIP by using the `add ns ip` command, according to your requirement.

Before you begin

Before you begin, create a VPX instance by following the steps given at this link:

Configure a Citrix ADC VPX standalone instance

For this use case, the NSDoc0330VM VPX instance is created.

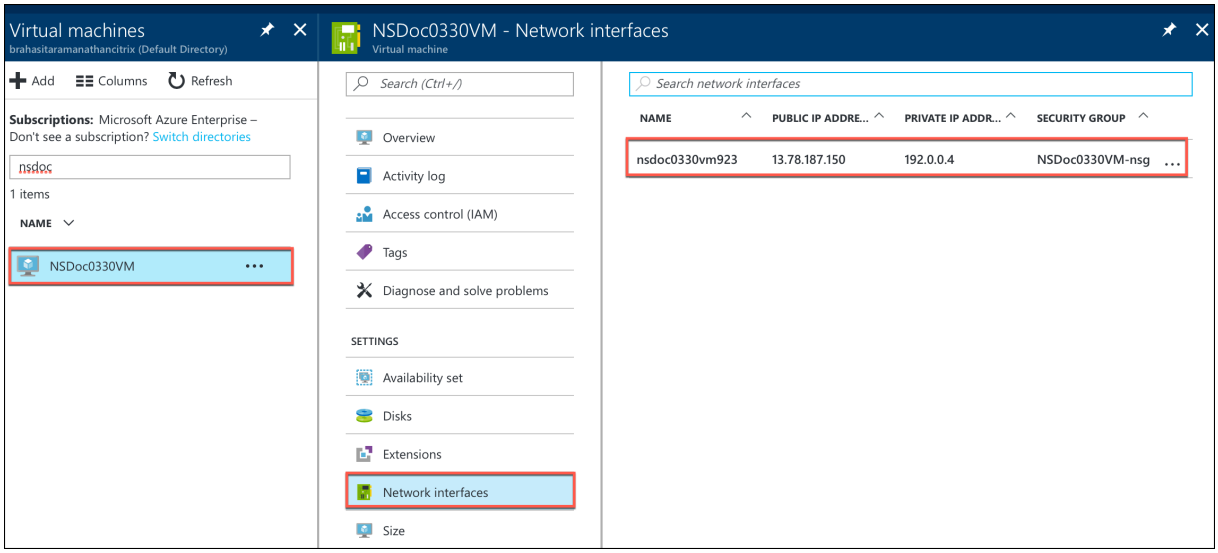
Procedure to configure multiple IP addresses for a Citrix ADC VPX instance in standalone mode.

For configuring multiple IP addresses for a Citrix ADC VPX appliance in standalone mode:

1. Add IP addresses to the VM
2. Configure Citrix ADC -owned IP addresses

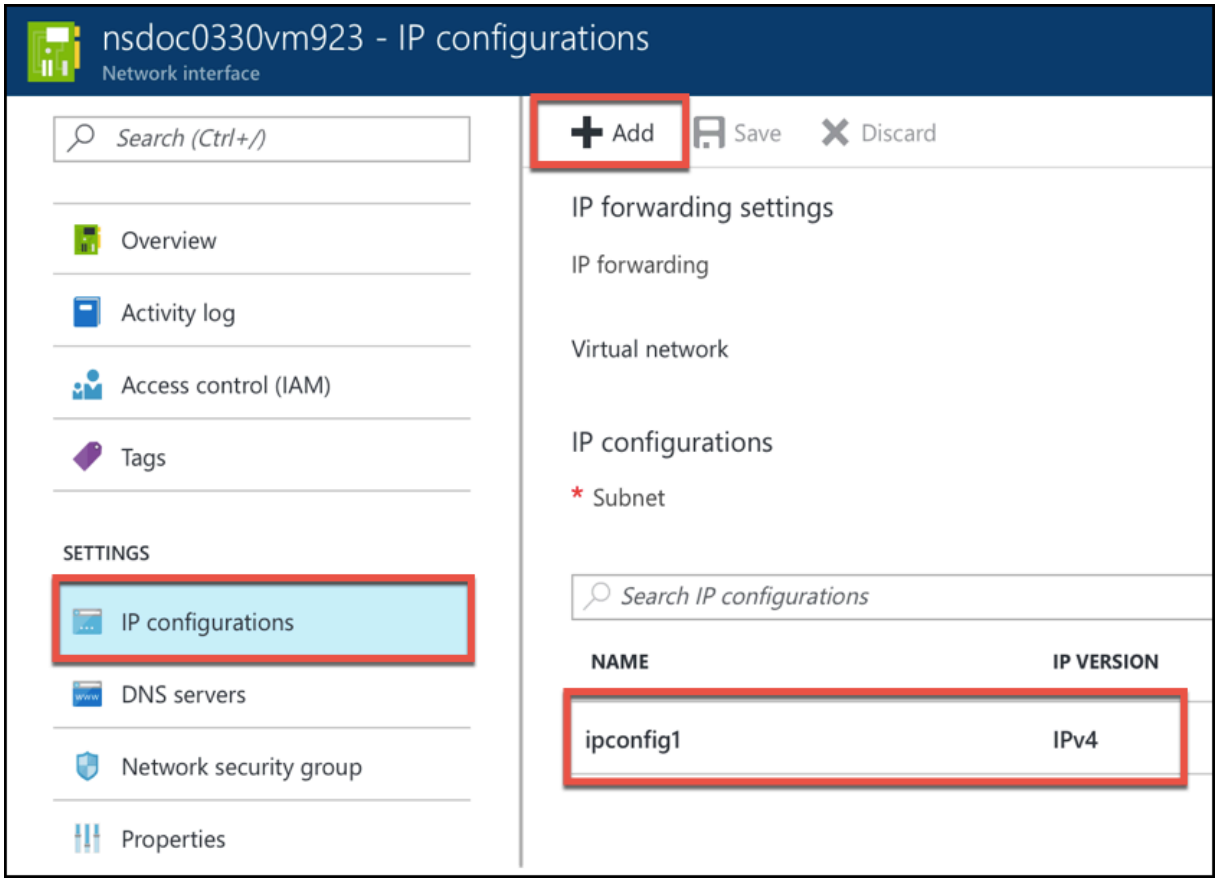
Step 1: Add IP addresses to the VM

1. In the portal, click **More services > type virtual machines** in the filter box, and then click **Virtual machines**.
2. In the **Virtual machines** blade, click the VM you want to add IP addresses to. Click **Network interfaces** in the virtual machine blade that appears, and then select the network interface.



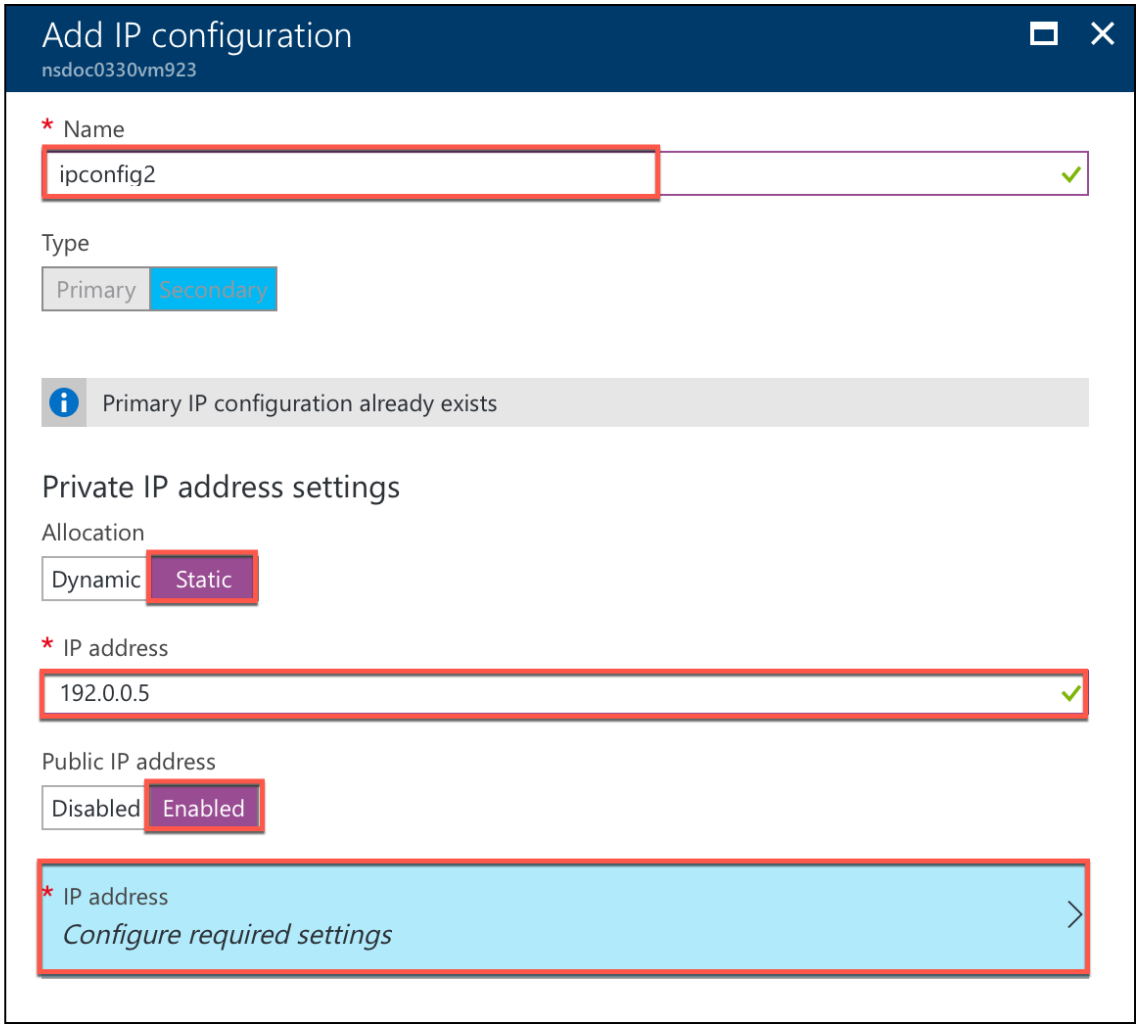
In the blade that appears for the NIC you selected, click **IP configurations**. The existing IP configuration that was assigned when you created the VM, **ipconfig1**, is displayed. For this use case, make sure the IP addresses associated with ipconfig1 are static. Next, create two more IP configurations: ipconfig2 (VIP) and ipconfig3 (SNIP).

To create more `ipconfigs`, create **Add**.



In the **Add IP configuration** window, enter a **Name**, specify allocation method as **Static**, enter an IP address (192.0.0.5 for this use case), and enable **Public IP address**.

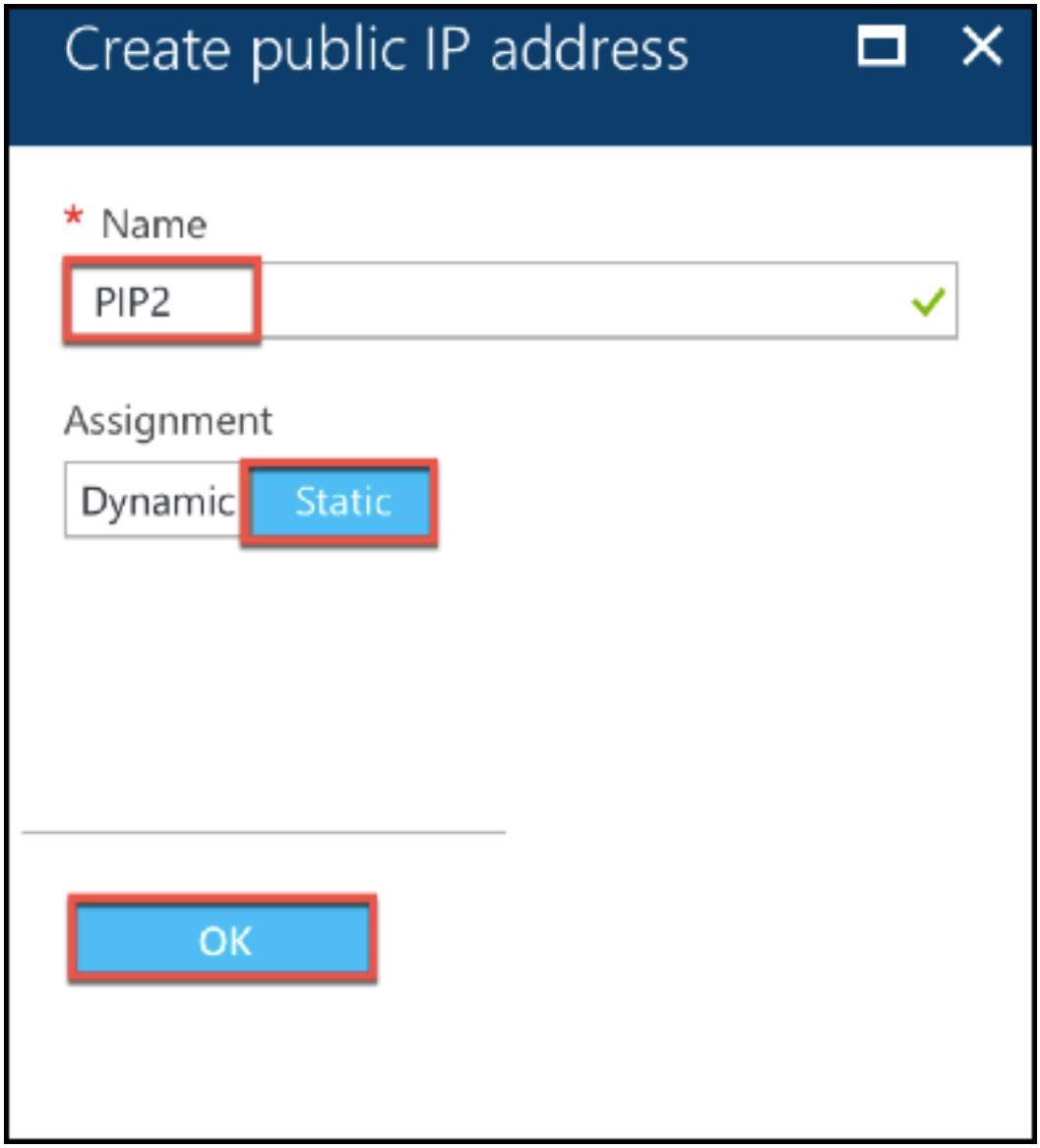
> **Note**
>
> Before adding a static private IP address, check for IP address availability and make sure the IP address belongs to the same subnet to which the NIC is attached.



Next, click **Configure required settings** to create a static public IP address for ipconfig2.

By default, public IPs are dynamic. To make sure that the VM always uses the same public IP address, create a static Public IP.

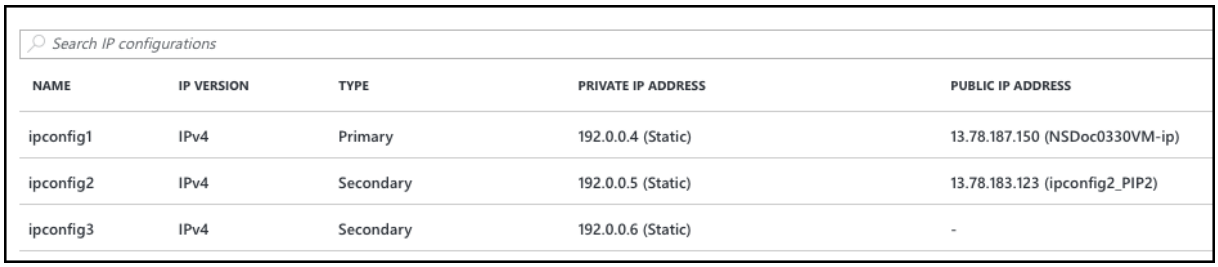
In the Create public IP address blade, add a Name, under Assignment click **Static**. And then click **OK**.



Note

Even when you set the allocation method to static, you cannot specify the actual IP address assigned to the public IP resource. Instead, it gets allocated from a pool of available IP addresses in the Azure location the resource is created in.

Follow the steps to add one more IP configuration for ipconfig3. Public IP is not mandatory.



Step 2: Configure Citrix ADC-owned IP addresses

Configure the Citrix ADC-owned IP addresses by using the GUI or the command `add ns ip`. For more information, see Configuring Citrix ADC-Owned IP Addresses.

Configure a high-availability setup with multiple IP addresses and NICs

September 14, 2021

In a Microsoft Azure deployment, a high-availability configuration of two Citrix ADC VPX instances is achieved by using the Azure Load Balancer (ALB). This is achieved by configuring a health probe on ALB, which monitors each VPX instance by sending a health probe at every 5 seconds to both primary and secondary instances.

In this setup, only the primary node responds to health probes and the secondary does not. Once the primary sends the response to the health probe, the ALB starts sending the data traffic to the instance. If the primary instance misses two consecutive health probes, ALB does not redirect traffic to that instance. On failover, the new primary starts responding to health probes and the ALB redirects traffic to it. The standard VPX high availability failover time is three seconds. The total failover time that might take for traffic switching can be a maximum of 13 seconds.

You can deploy a pair of Citrix ADC VPX instances with multiple NICs in an active-passive high availability (HA) setup on Azure. Each NIC can contain multiple IP addresses.

The following options are available for a multi-NIC high availability deployment:

- High availability using Azure availability set
- High availability using Azure availability zones

For more information about Azure Availability Set and Availability Zones, see the Azure documentation Manage the availability of Linux virtual machines.

High availability using availability set

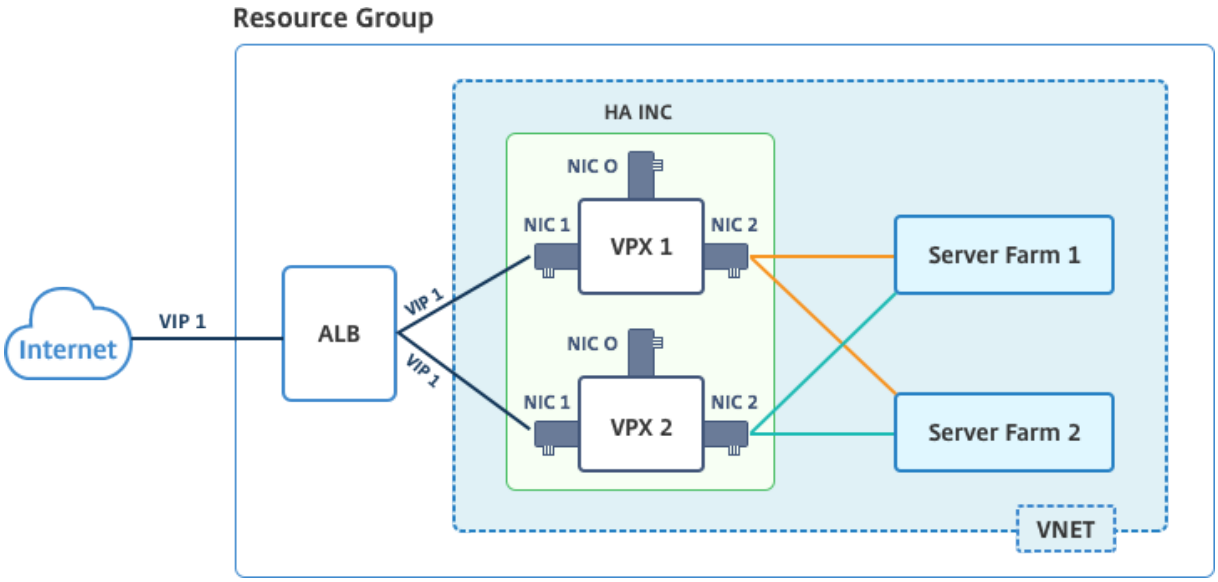
A high availability setup using a availability set must meet the following requirements:

- An HA Independent Network Configuration (INC) configuration
- The Azure Load Balancer (ALB) in Direct Server Return (DSR) mode

All traffic goes through the primary node. The secondary node remains in standby mode until the primary node fails.

> **Note**
>
> For a Citrix VPX high availability deployment on the Azure cloud to work, you need a floating public IP (PIP) that can be moved between the two VPX nodes. The Azure Load Balancer (ALB) provides that floating PIP, which is moved to the second node automatically in the event of a failover.

Diagram: Example of a high availability deployment architecture, using Azure Availability Set



In an active-passive deployment, the ALB front end public IP (PIP) addresses are added as the VIP addresses in each VPX node. In HA-INC configuration, the VIP addresses are floating and SNIP addresses are instance specific.

You can deploy a VPX pair in active-passive high availability mode in two ways by using:

- **Citrix ADC VPX standard high availability template**: use this option to configure an HA pair with the default option of three subnets and six NICs.
- **Windows PowerShell commands**: use this option to configure an HA pair according to your subnet and NIC requirements.

This topic describes how to deploy a VPX pair in active-passive HA setup by using the Citrix template. If you want to use PowerShell commands, see Configuring an HA Setup with Multiple IP Addresses and NICs by Using PowerShell Commands.

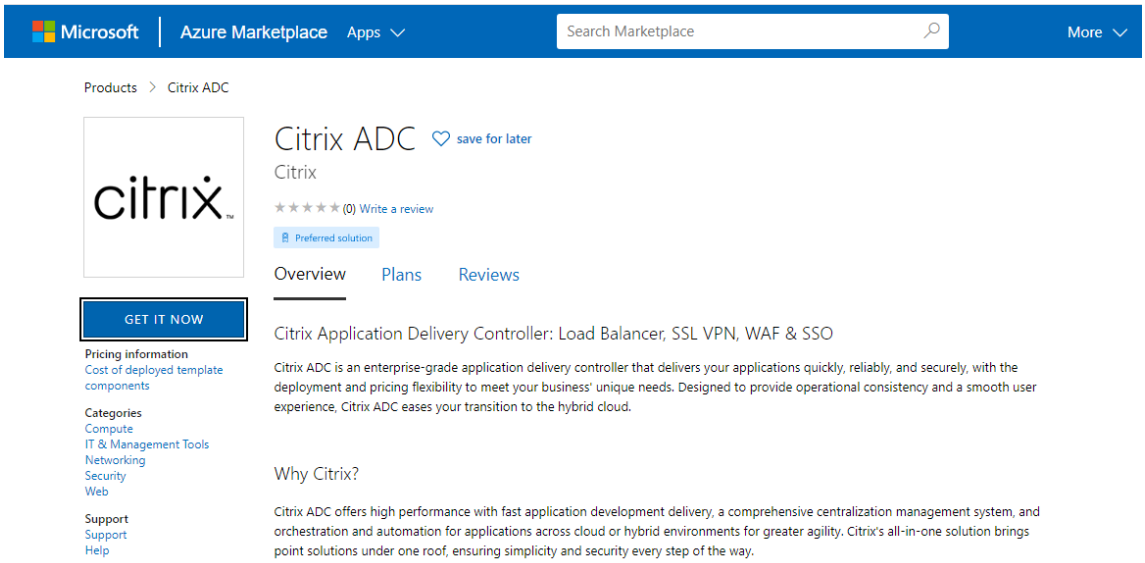
Configure HA-INC nodes by using the Citrix high availability template

You can quickly and efficiently deploy a pair of VPX instances in HA-INC mode by using the standard template. The template creates two nodes, with three subnets and six NICs. The subnets are for management, client, and server-side traffic, and each subnet has two NICs for both the VPX instances.

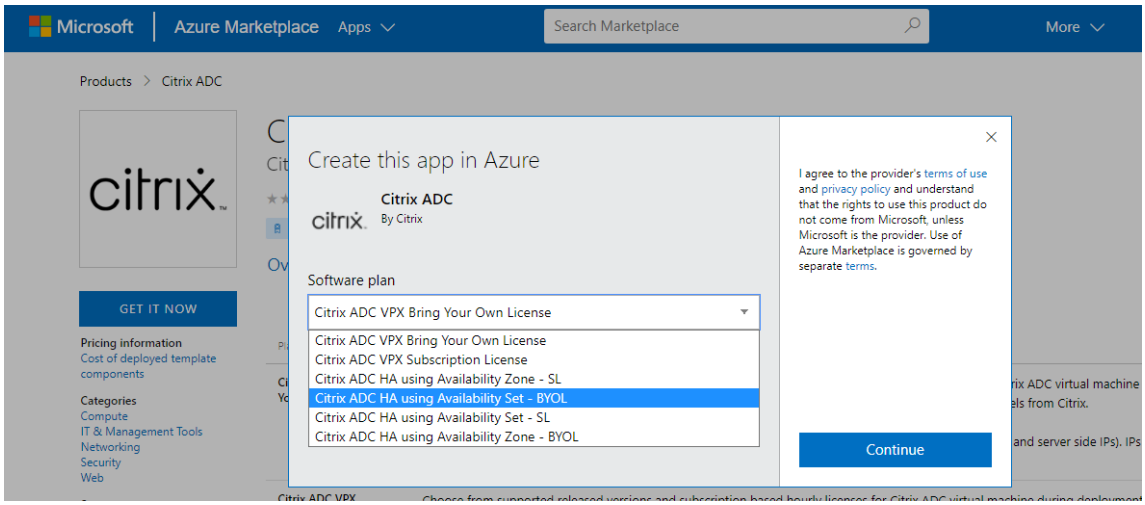
You can get the Citrix ADC HA Pair template at the Azure Marketplace.

Complete the following steps to launch the template and deploy a high availability VPX pair, by using Azure availability sets.

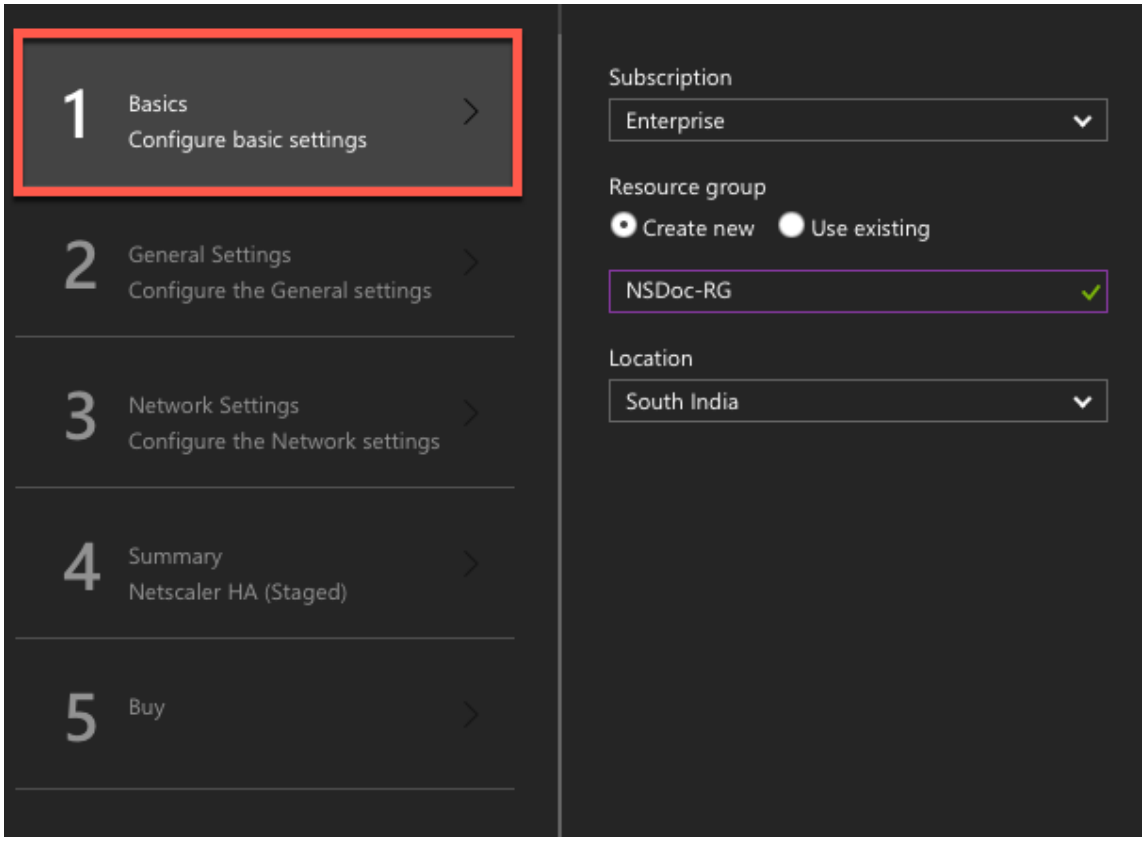
1. From Azure Marketplace, search **Citrix ADC**.



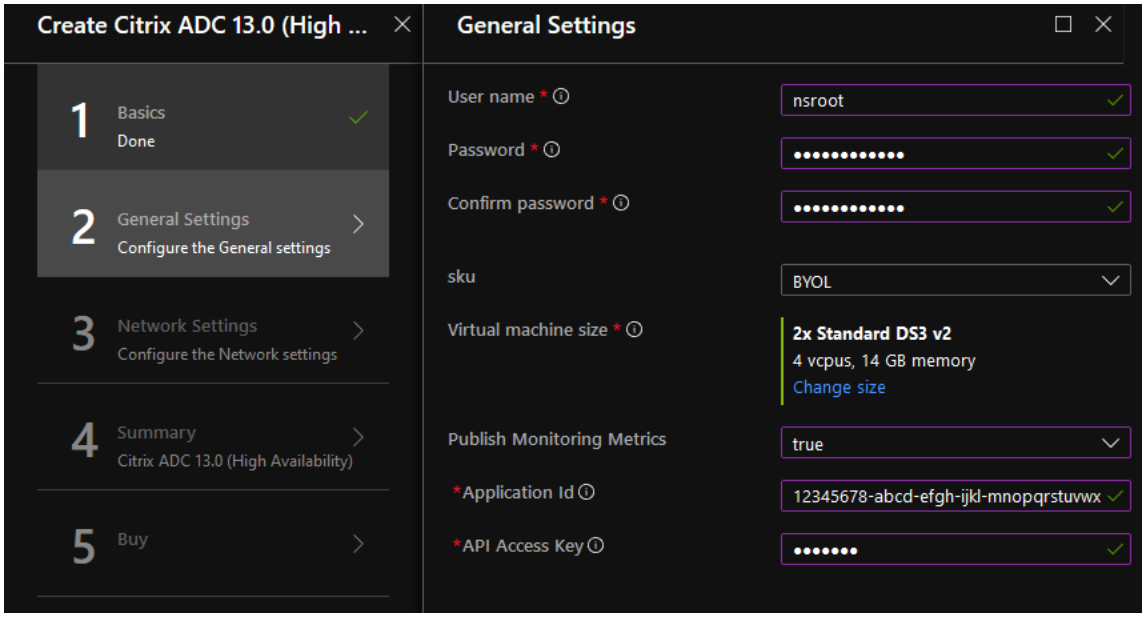
2. Click **GET IT NOW**.
3. Select the required HA deployment along with license, and click **Continue**.



4. The **Basics** page appears. Create a Resource Group and select **OK**.



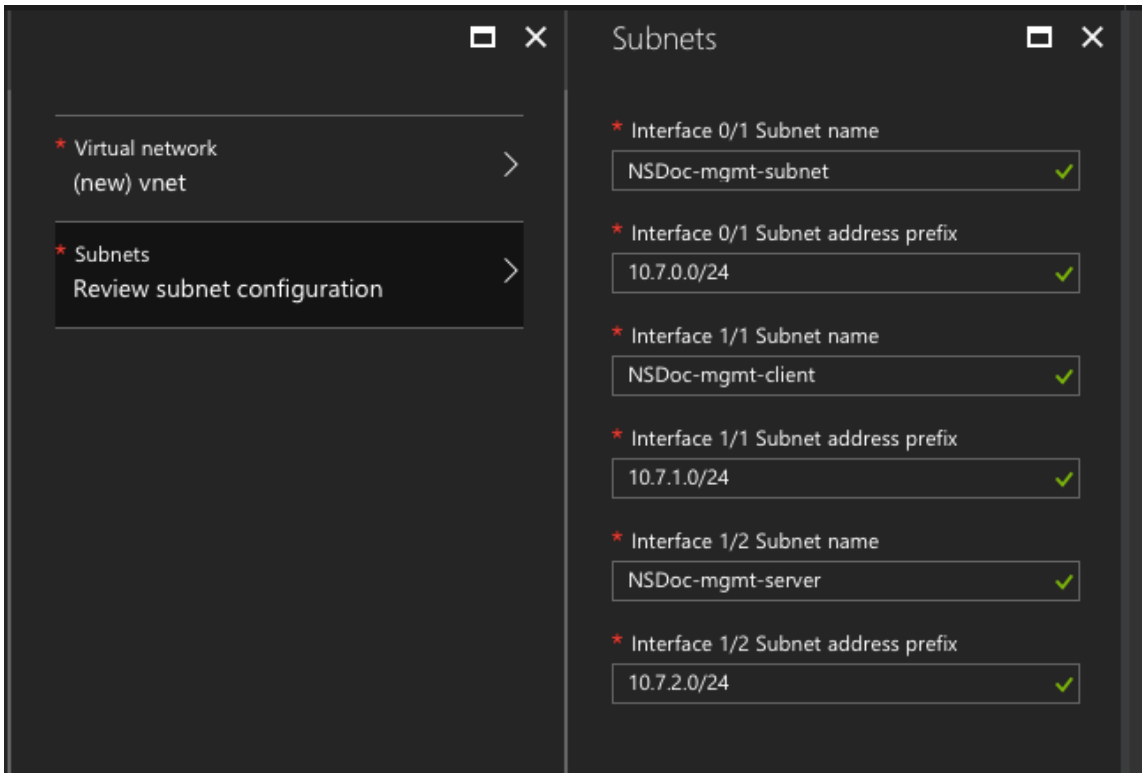
5. The **General Settings** page appears. Type the details and select **OK**.



> Note:
>
> By default, the **Publishing Monitoring Metrics** option is set to **false**. If you want to enable this option, select **true**.

> Create an Azure Active Directory (ADD) application and service principal that can access resources. Assign contributor role to the newly created AAD application. For more information, see Use portal to create an Azure Active Directory application and service principal that can access resources.

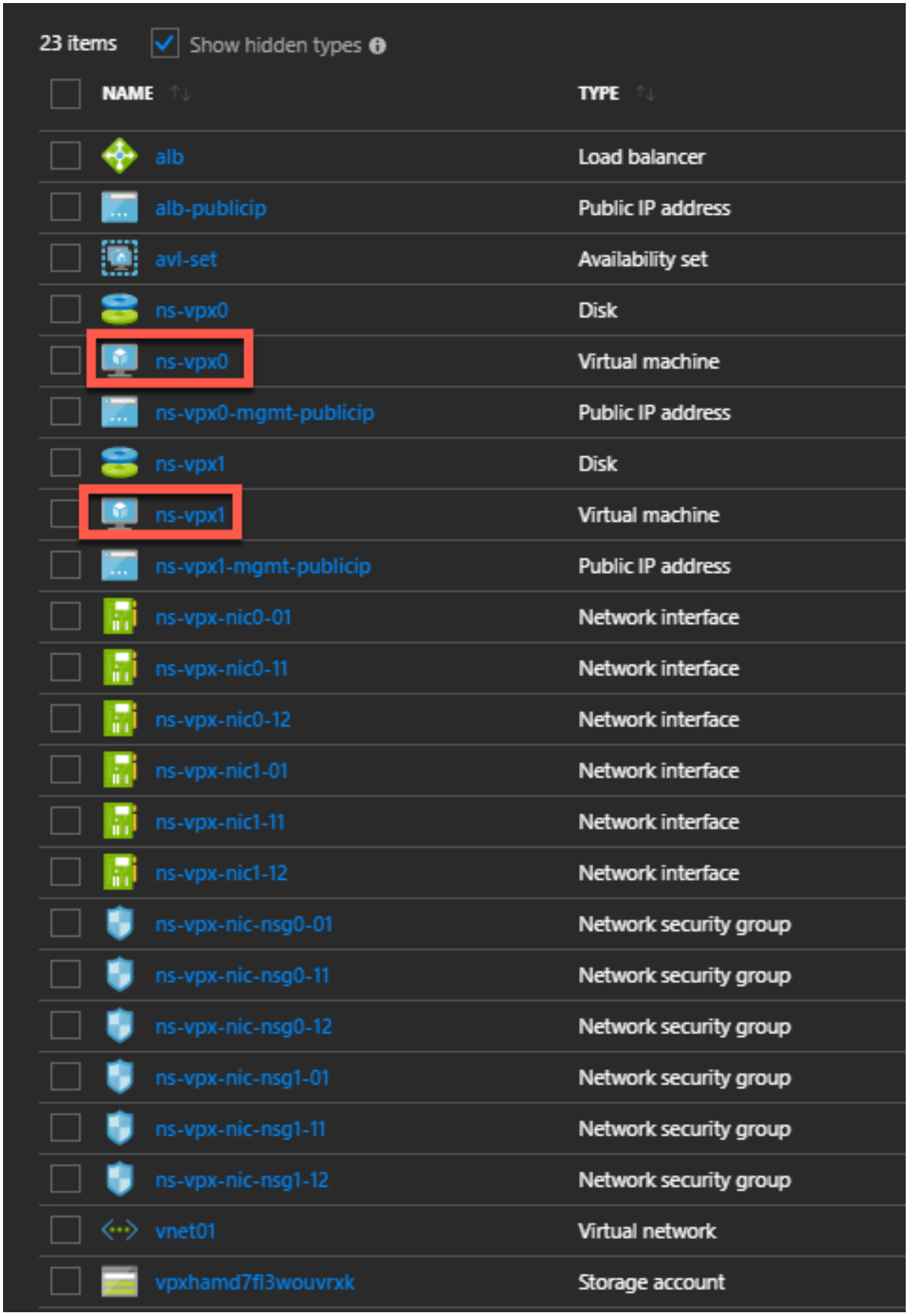
6. The **Network Settings** page appears. Check the VNet and subnet configurations, edit the required settings, and select **OK**.



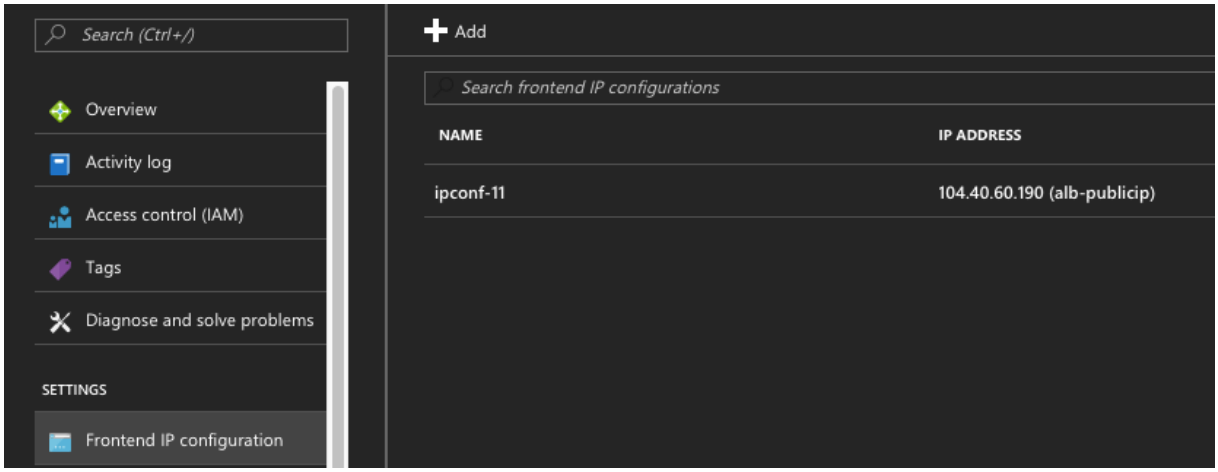
7. The **Summary** page appears. Review the configuration and edit accordingly. Select **OK** to confirm.
8. The **Buy** page appears. Select **Purchase** to complete the deployment.

It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the **Resource Group** in the Azure portal to see the configuration details, such as LB rules, back-end pools, health probes. The high availability pair appears as ns-vpx0 and ns-vpx1.

If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.



Next, you need to configure the load-balancing virtual server with the **ALB's Frontend public IP (PIP) address**, on primary node. To find the ALB PIP, select ALB > **Frontend IP configuration**.



See the **Resources** section for more information about how to configure the load-balancing virtual server.

Resources:

The following links provide additional information related to HA deployment and virtual server configuration:

- Configuring high availability nodes in different subnets
- Set up basic load balancing

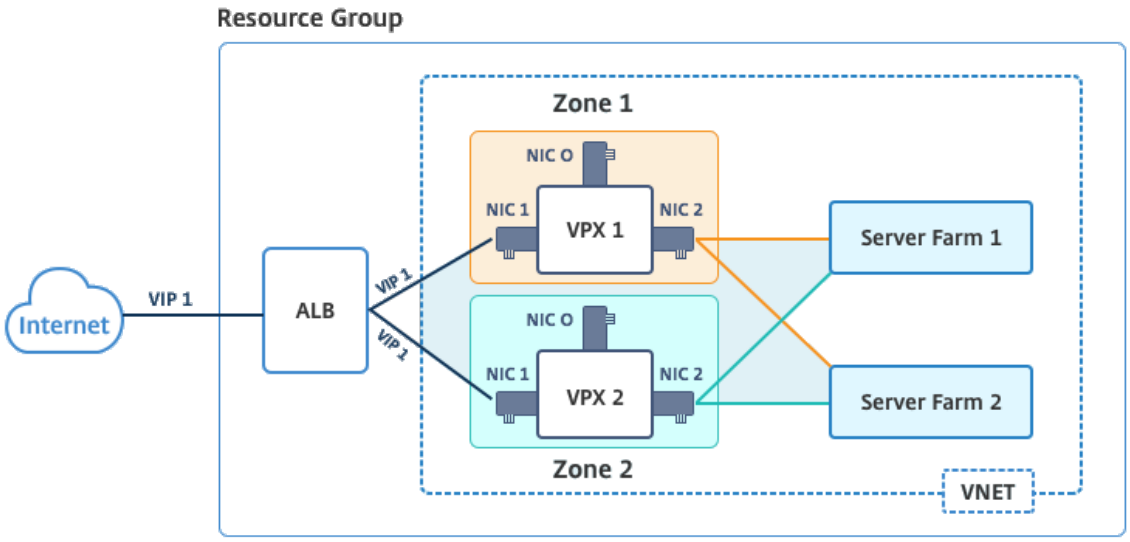
Related resources:

- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands
- Configuring GSLB on Active-Standby HA Deployment on Azure

High availability using availability zones

Azure Availability Zones are fault-isolated locations within an Azure region, providing redundant power, cooling, and networking and increasing resiliency. Only specific Azure regions support Availability Zones. For more information, see the Azure documentation [What are Availability Zones in Azure.

Diagram: Example of a high availability deployment architecture, using Azure Availability Zones



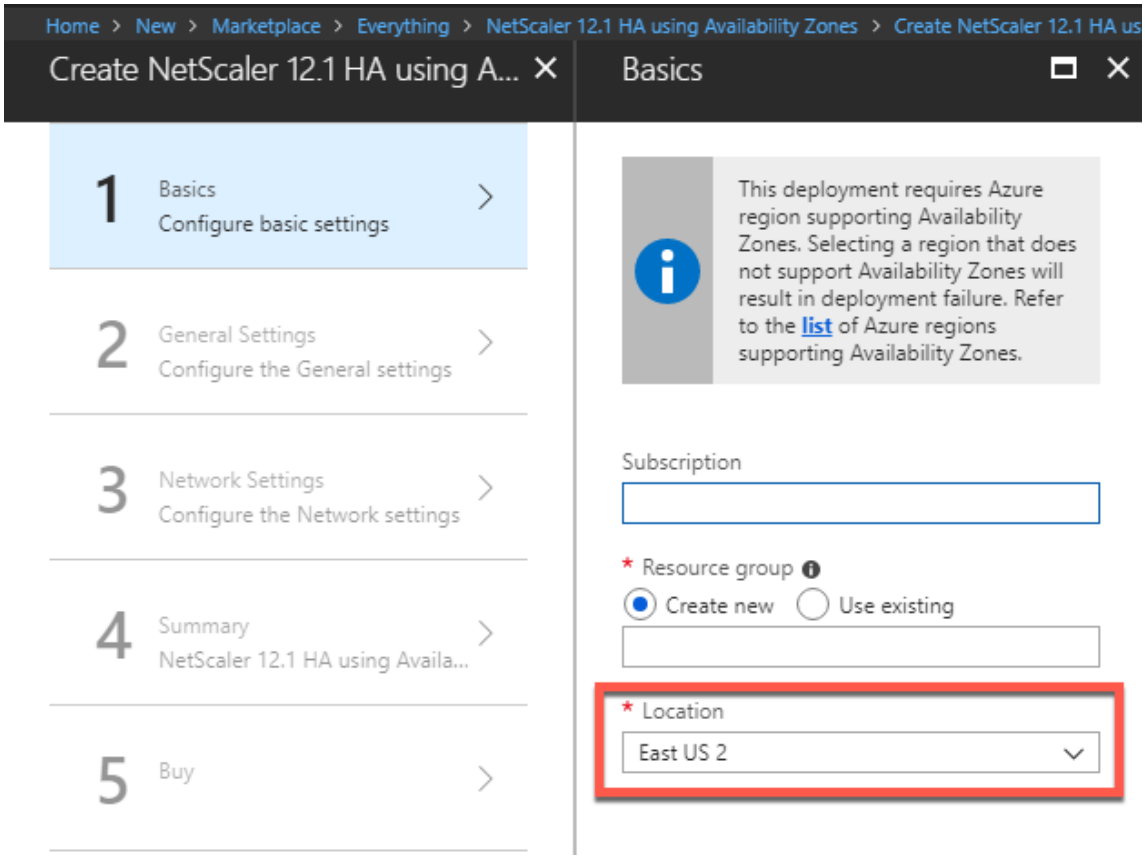
You can deploy a VPX pair in high availability mode by using the template called “NetScaler 13.0 HA using Availability Zones,” available in Azure Marketplace.

Complete the following steps to launch the template and deploy a high availability VPX pair, by using Azure Availability Zones.

1. From Azure Marketplace, select and initiate the Citrix solution template.

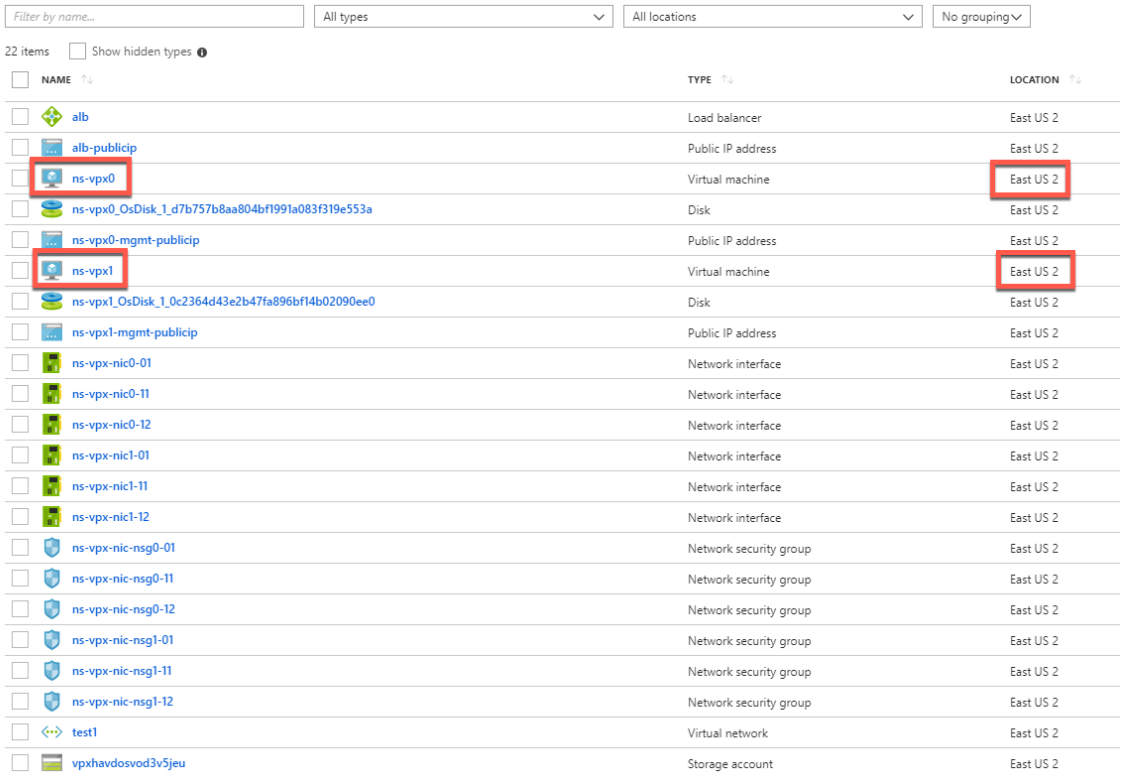
2. Ensure deployment type is Resource Manager and select **Create**.
3. The **Basics** page appears. Enter the details and click **OK**.

> Note: Ensure that you select an Azure region that supports Availability Zones. For more information about regions that support Availability Zones, see Azure documentation What are Availability Zones in Azure?



4. The **General Settings** page appears. Type the details and select **OK**.
5. The **Network Setting** page appears. Check the VNet and subnet configurations, edit the required settings, and select **OK**.
6. The **Summary** page appears. Review the configuration and edit accordingly. Select **OK** to confirm.
7. The **Buy** page appears. Select **Purchase** to complete the deployment.

   It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the **Resource Group** to see the configuration details, such as LB rules, back-end pools, health probes, and so on, in the Azure portal. The high availability pair appears as ns-vpx0 and ns-vpx1. Also, you can see the location under the **Location** column.



If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.

Monitor your instances using metrics in Azure monitor

You can use metrics in the Azure monitor data platform to monitor a set of Citrix ADC VPX resources such as CPU, memory utilization, and throughput. Metrics service monitors Citrix ADC VPX resources that run on Azure, in real time. You can use **Metrics Explorer** to access the collected data. For more information, see Azure Monitor Metrics overview.

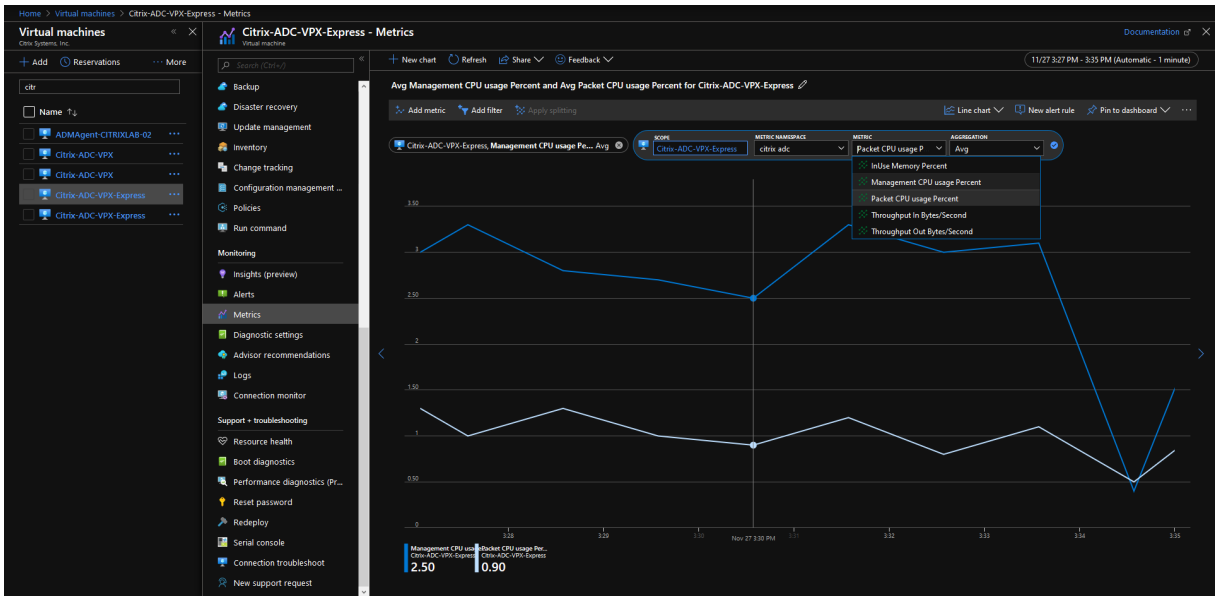
Points to note

- If you deploy a Citrix ADC VPX instance on Azure by using the Azure Marketplace offer, Metrics service is disabled by default.
- The Metrics service is not supported in Azure CLI.
- Metrics are available for CPU (management and packet CPU usage), memory, and throughput (inbound and outbound).

How to view metrics in Azure monitor

To view metrics in the Azure monitor for your instance, perform these steps:

1. Log on to **Azure Portal > Virtual Machines**.
2. Select the virtual machine that is the Primary Node.
3. In the **Monitoring** section, click **Metrics**.
4. From the **Metric Namespace** drop-down menu, click **Citrix ADC**.
5. Under **All metrics** in **Metrics** drop-down menu, click the metrics you want to view.
6. Click **Add metric** to view another metric on the same chart. Use the Chart options to customize your chart.



Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands

September 14, 2021

You can deploy a pair of Citrix ADC VPX instances with multiple NICs in an active-passive high availability (HA) setup on Azure. Each NIC can contain multiple IP addresses.

An active-passive deployment requires:

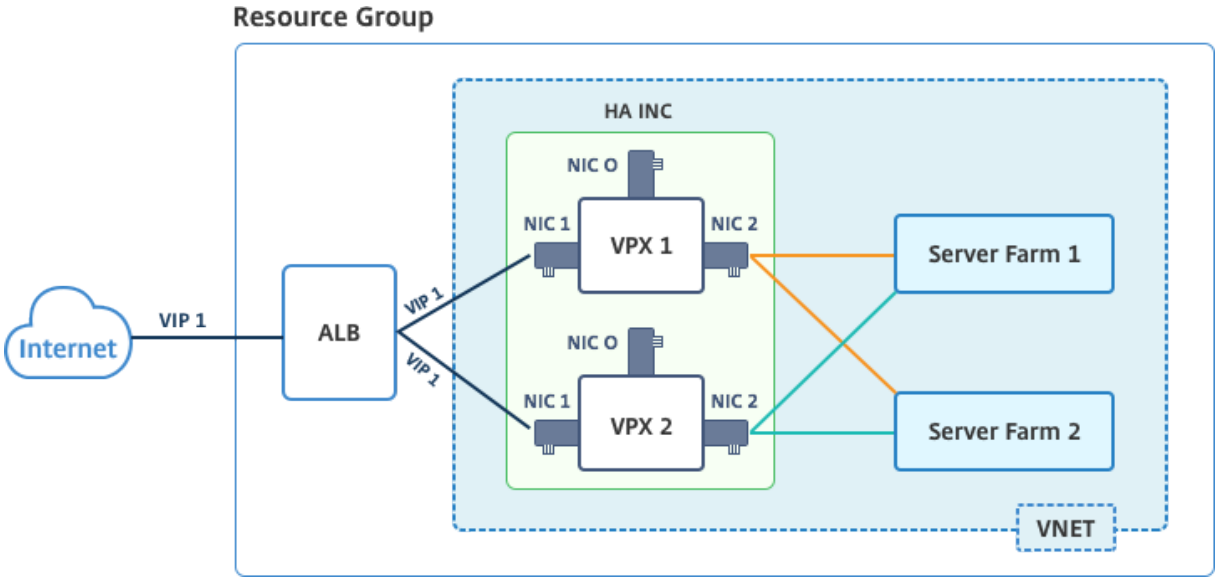
- An HA Independent Network Configuration (INC) configuration
- The Azure Load Balancer (ALB) in Direct Server Return (DSR) mode

All traffic goes through the primary node. The secondary node remains in standby mode until the primary node fails.

> **Note**
>
> For a Citrix ADC VPX high availability deployment on an Azure cloud to work, you need a floating

> public IP (PIP) that can be moved between the two high-availability nodes. The Azure Load Balancer (ALB) provides that floating PIP, which is moved to the second node automatically in the event of a failover.

Diagram: Example of an active-passive deployment architecture



In an active-passive deployment, the ALB floating public IP (PIP) addresses are added as the VIP addresses in each VPX node. In HA-INC configuration, the VIP addresses are floating and SNIP addresses are instance specific.

ALB monitors each VPX instance by sending health probe at every 5 seconds and redirects traffic to that instance only that sends health probes response on regular interval. So in an HA setup, the primary node responds to health probes and secondary does not. If the primary instances miss two consecutive health probes, ALB does not redirect traffic to that instance. On failover, the new primary starts responding to health probes and the ALB redirects traffic to it. The standard VPX high availability failover time is three seconds. The total failover time that might take for traffic switching can be maximum of 13 seconds.

You can deploy a VPX pair in active-passive HA setup in two ways by using:

- **Citrix ADC VPX Standard high availability template**: use this option to configure an HA pair with the default option of three subnets and six NICs.
- **Windows PowerShell commands**: use this option to configure an HA pair according to your subnet and NIC requirements.

This topic describes how to deploy a VPX pair in active-passive HA setup by using PowerShell commands. If you want to use the Citrix ADC VPX Standard HA template, see Configuring an HA Setup with Multiple IP Addresses and NICs.

Configure HA-INC nodes by using PowerShell Commands

Scenario: HA-INC PowerShell deployment

In this scenario, you deploy a Citrix ADC VPX pair by using the topology given in the table. Each VPX instance contains three NICs, with each NIC is deployed in a different subnet. Each NIC is assigned an IP configuration.

ALB	VPX1	VPX2
ALB is associated with public IP 3 (pip3)	Management IP is configured with IPConfig1, which includes one public IP (pip1) and one private IP (12.5.2.24); nic1; Mgmtsubnet=12.5.2.0/24	Management IP is configured with IPConfig5, which includes one public IP (pip3) and one private IP (12.5.2.26);nic4;Mgmtsubnet=12.5.2.0/24
LB rules and port configured are HTTP (80),SSL (443), health probe (9000)	Client-side IP is configured with IPConfig3, which includes one private IP(12.5.1.27);nic2; FrontEndsubet=12.5.1.0/24	Client-side IP is configured with IPConfig7, which includes one private IP (12.5.1.28);nic5;FrontEndsubet=12.5.1.0/24
-	Server-side IP is configured with IPConfig4, which includes one private IP(12.5.3.24); nic3;BackendSubnet=12.5.3.0/2	Server-side IP is configured with IPConfig8, which includes one private IP(12.5.3.28);nic6;BackendSubnet=12.5.3.0/2
-	Rules and ports for NSG are SSH (22),HTTP (80),HTTPS (443)	-

Parameter settings

The following parameter settings are used in this scenario.

$locName= “South east Asia”

$rgName = “MulitIP-MultiNIC-RG”

$nicName1= “VM1-NIC1”

$nicName2 = “VM1-NIC2”

$nicName3= “VM1-NIC3”

$nicName4 = “VM2-NIC1”

$nicName5= “VM2-NIC2”

$nicName6 = “VM2-NIC3”

$vNetName = “Azure-MultiIP-ALB-vnet”

$vNetAddressRange= “12.5.0.0/16”

$frontEndSubnetName= “frontEndSubnet”

$frontEndSubnetRange= “12.5.1.0/24”

$mgmtSubnetName= “mgmtSubnet”

$mgmtSubnetRange= “12.5.2.0/24”

$backEndSubnetName = “backEndSubnet”

$backEndSubnetRange = “12.5.3.0/24”

$prmStorageAccountName = “multiipmultinicbstorage”

$avSetName = “multiple-avSet”

$vmSize= “Standard_DS4_V2”

$publisher = “Citrix”

$offer = “netscalervpx-120”

$sku = “netscalerbyol”

$version=”latest”

$pubIPName1=”VPX1MGMT”

$pubIPName2=”VPX2MGMT”

$pubIPName3=”ALBPIP”

$domName1=”vpx1dns”

$domName2=”vpx2dns”

$domName3=”vpxalbdns”

$vmNamePrefix=”VPXMultiIPALB”

$osDiskSuffix1=”osmultiipalbdiskdb1”

$osDiskSuffix2=”osmultiipalbdiskdb2”

$lbName= “MultiIPALB”

$frontEndConfigName1= “FrontEndIP”

$backendPoolName1= “BackendPoolHttp”

$lbRuleName1= “LBRuleHttp”

$healthProbeName= “HealthProbe”

$nsgName=”NSG-MultiIP-ALB”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

To complete the deployment, complete the following steps by using PowerShell commands:

1. Create a resource group, storage account, and availability set
2. Create a network security group and add rules
3. Create a virtual network and three subnets
4. Create public IP addresses
5. Create IP configurations for VPX1
6. Create IP configurations for VPX2
7. Create NICs for VPX1
8. Create NICs for VPX2
9. Create VPX1
10. Create VPX2
11. Create ALB

Create a resource group, storage account, and availability set.

```
New-AzureRmResourceGroup -Name $rgName -Location $locName


$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $rgName -Type Standard_LRS
     -Location $locName


$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $rgName -Location $locName
```

Create a network security group and add rules.

```
$rule1 = New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
    Description "Allow HTTP" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 101


```



```
-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 80


$rule2 = New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
    Description "Allow HTTPS" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 110


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 443


$rule3 = New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -
    Description "Allow SSH" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 120


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 22


$nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
    Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

Create a virtual network and three subnets.

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontEndSubnetName -AddressPrefix $frontEndSubnetRange (this
    parameter value should be as per your requirement)


$mgmtSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $mgmtSubnetName
     -AddressPrefix $mgmtSubnetRange


$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backEndSubnetName -AddressPrefix $backEndSubnetRange


$vnet =New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
    $rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
    $frontendSubnet,$backendSubnet, $mgmtSubnet


```



```
$subnetName ="frontEndSubnet"


$subnet1=$vnet.Subnets|?{
 $_.Name -eq $subnetName }


$subnetName="backEndSubnet"


$subnet2=$vnet.Subnets|?{
 $_.Name -eq $subnetName }


$subnetName="mgmtSubnet"


$subnet3=$vnet.Subnets|?{
 $_.Name -eq $subnetName }
```

Create public IP addresses.

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $rgName -DomainNameLabel $domName1 -Location $locName -
    AllocationMethod Dynamic

$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $rgName -DomainNameLabel $domName2 -Location $locName -
    AllocationMethod Dynamic

$pip3=New-AzureRmPublicIpAddress -Name $pubIPName3 -ResourceGroupName
    $rgName -DomainNameLabel $domName3 -Location $locName -
    AllocationMethod Dynamic
```

Create IP configurations for VPX1.

```
$IpConfigName1 = "IPConfig1"


$IPAddress = "12.5.2.24"


$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
```



```
    Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip1
    -Primary
8
9
10 $IPConfigName3="IPConfig-3"
11
12
13 $IPAddress="12.5.1.27"
14
15
16 $IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
18
19 $IPConfigName4 = "IPConfig-4"
20
21
22 $IPAddress = "12.5.3.24"
23
24
25 $IPConfig4 = New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

Create IP configurations for VPX2.

```
1  $IpConfigName5 = "IPConfig5"
2
3
4  $IPAddress="12.5.2.26"
5
6
7  $IPConfig5=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName5 -
    Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip2
    -Primary
8
9
10 $IPConfigName7="IPConfig-7"
11
12
13 $IPAddress="12.5.1.28"
14
15
16 $IPConfig7=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName7 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
```



```
18
19 $IPConfigName8="IPConfig-8"
20
21
22 $IPAddress="12.5.3.28"
23
24
25 $IPConfig8=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName8 -
       Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

Create NICs for VPX1.

```
1 $nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig1 -
      NetworkSecurityGroupId $nsg.Id
2
3
4 $nic2=New-AzureRmNetworkInterface -Name $nicName2 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig3 -
      NetworkSecurityGroupId $nsg.Id
5
6
7 $nic3=New-AzureRmNetworkInterface -Name $nicName3 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig4 -
      NetworkSecurityGroupId $nsg.Id
```

Create NICs for VPX2.

```
1 $nic4=New-AzureRmNetworkInterface -Name $nicName4 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig5 -
      NetworkSecurityGroupId $nsg.Id
2
3
4 $nic5=New-AzureRmNetworkInterface -Name $nicName5 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig7 -
      NetworkSecurityGroupId $nsg.Id
5
6
7 $nic6=New-AzureRmNetworkInterface -Name $nicName6 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig8 -
      NetworkSecurityGroupId $nsg.Id
```

Create VPX1.

This step includes the following substeps:

- Create VM config object
- Set credentials, OS, and image
- Add NICs
- Specify OS disk and create VM

```
$suffixNumber = 1

$vmName=$vmNamePrefix + $suffixNumber

$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id

$cred=Get-Credential -Message "Type the name and password for VPX
     login."

$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred

$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.
    Id -Primary

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.
    Id

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.
    Id

$osDiskName=$vmName + "-" + $osDiskSuffix1

$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "
    vhds/" + $osDiskName + ".vhd"

$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -
    VhdUri $osVhdUri -CreateOption fromImage

Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product
    $offer -Name $sku

New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
    $locName
```


Create VPX2.

```
```
$suffixNumber=2

$vmName=$vmNamePrefix + $suffixNumber

$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id

$cred=Get-Credential -Message "Type the name and password for VPX login
    ."

$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred

$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic4.Id -
    Primary

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic5.Id

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic6.Id

$osDiskName=$vmName + "-" + $osDiskSuffix2

$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     + $osDiskName + ".vhd"

$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage

```

 
```

```
37
38 Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
       Name $sku
39
40
41 New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
       $locName
42 <!--NeedCopy--> ```
```

To view private and public IP addresses assigned to the NICs, type the following commands:

```
1 ```
2 $nic1.IPConfig
3
4
5 $nic2.IPConfig
6
7
8 $nic3.IPConfig
9
10
11 $nic4.IPConfig
12
13
14 $nic5.IPConfig
15
16
17 $nic6.IPConfig
18 <!--NeedCopy--> ```
```

Create Azure load balance (ALB).

This step includes the following substeps:

- Create front end IP config
- Create health probe
- Create back end address pool
- Create load-balancing rules (HTTP and SSL)
- Create ALB with front end IP config, back end address pool, and LB rule
- Associate IP config with back end pools

```
$frontEndIP1=New-AzureRmLoadBalancerFrontendIpConfig -Name $frontEndConfigName1
 -PublicIpAddress $pip3
```



```
$healthProbe=New-AzureRmLoadBalancerProbeConfig -Name $healthProbeName
-Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

$beAddressPool1=New-AzureRmLoadBalancerBackendAddressPoolConfig -Name
$backendPoolName1

$lbRule1=New-AzureRmLoadBalancerRuleConfig -Name $lbRuleName1 -FrontendIpConfigur
 $frontEndIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -
Protocol Tcp -FrontendPort 80 -BackendPort 80 -EnableFloatingIP

$lb=New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name $lbName -
Location $locName -FrontendIpConfiguration $frontEndIP1 -LoadBalancingRule
 $lbRule1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb.
BackendAddressPools[0])

$nic5.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb.
BackendAddressPools[0])

$lb=$lb |Set-AzureRmLoadBalancer

$nic2=$nic2 | Set-AzureRmNetworkInterface

$nic5=$nic5 | Set-AzureRmNetworkInterface
```

After you've successfully deployed the Citrix ADC VPX pair, log on to each VPX instance to configure HA-INC, and SNIP and VIP addresses.

1. Type the following command to add HA nodes.

   ```
   add ha node 1 PeerNodeNSIP -inc Enabled
   ```

2. Add private IP addresses of client-side NICs as SNIPs for VPX1 (NIC2) and VPX2 (NIC5)

   ```
   add nsip privateIPofNIC2 255.255.255.0 -type SNIP
   add nsip privateIPofNIC5 255.255.255.0 -type SNIP
   ```

3. Add load-balancing virtual server on the primary node with front-end IP address (public IP) of ALB.

   ```
   add lb virtual server v1 HTTP FrontEndIPofALB 80
   ```

Related resources:

Configuring GSLB on Active-Standby HA Deployment on Azure

Configure a Citrix ADC VPX instance to use Azure accelerated networking

September 14, 2021

Accelerated networking enables the single root I/O virtualization (SR-IOV) virtual function (VF) NIC to a virtual machine, which improves the networking performance. You can use this feature with heavy workloads that need to send or receive data at higher throughput with reliable streaming and lower CPU utilization.
When a NIC is enabled with accelerated networking, Azure bundles the NIC's existing para virtualized (PV) interface with an SR-IOV VF interface. The support of SR-IOV VF interface enables and enhances the throughput of the Citrix ADC VPX instance.

Accelerated networking provides the following benefits:

- Lower latency
- Higher packets per second (pps) performance
- Enhanced throughput
- Reduced jitter
- Decreased CPU utilization

> Note
>
> Azure accelerated networking is supported on Citrix ADC VPX instances from release 13.0 build 76.29 onwards.

Prerequisites

- Ensure that your VM size matches the requirements for Azure accelerated networking.
- Stop VMs (individual or in an availability set) before enabling accelerated networking on any NIC.

Limitations

Accelerated networking can be enabled only on some instance types. For more information, see Supported instance types.

NICs supported for accelerated networking

Azure provides Mellanox ConnectX3 and ConnectX4 NICs in the SR-IOV mode for accelerated networking.

When accelerated networking is enabled on a Citrix ADC VPX interface, Azure bundles either ConnectX3 or ConnectX4 interface with the existing PV interface of a Citrix ADC VPX appliance.

For more information about enabling accelerated networking before attaching an interface to a VM, see Create a network interface with accelerated networking.

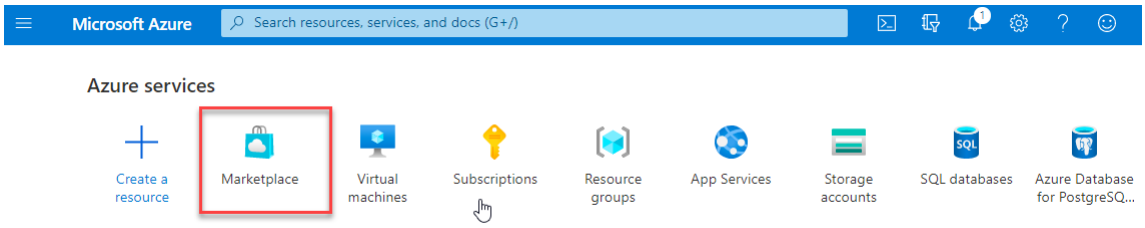
For more information about enabling accelerated networking on an existing interface on a VM, see Enable existing interfaces on a VM.

How to enable accelerated networking on Citrix ADC VPX instance using the Azure console

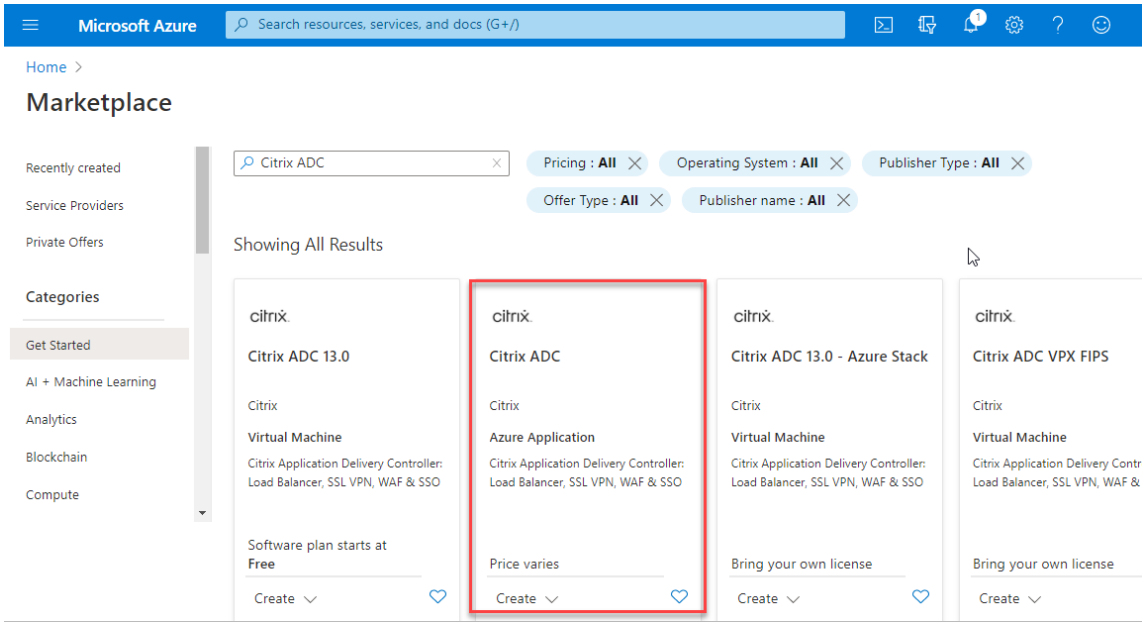
You can enable accelerated networking on a specific interface using the Azure console or the Azure PowerShell.

Do the following steps to enable accelerated networking by using Azure availability sets or availability zones.

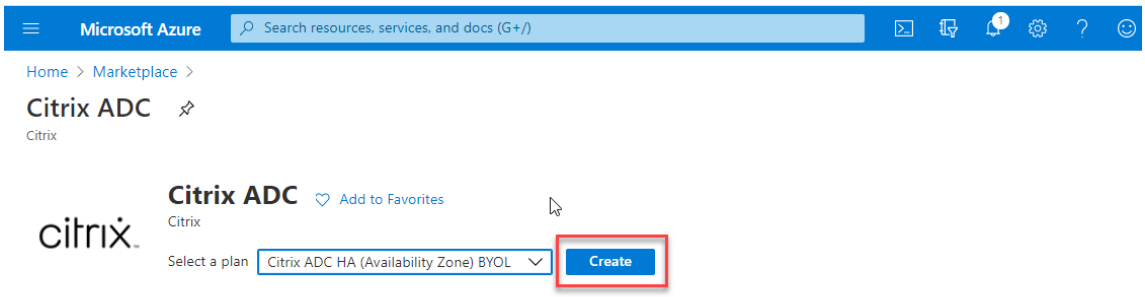
1. Log in to Azure portal, and navigate to **Azure Marketplace**.



2. From the **Azure Marketplace**, search **Citrix ADC**.

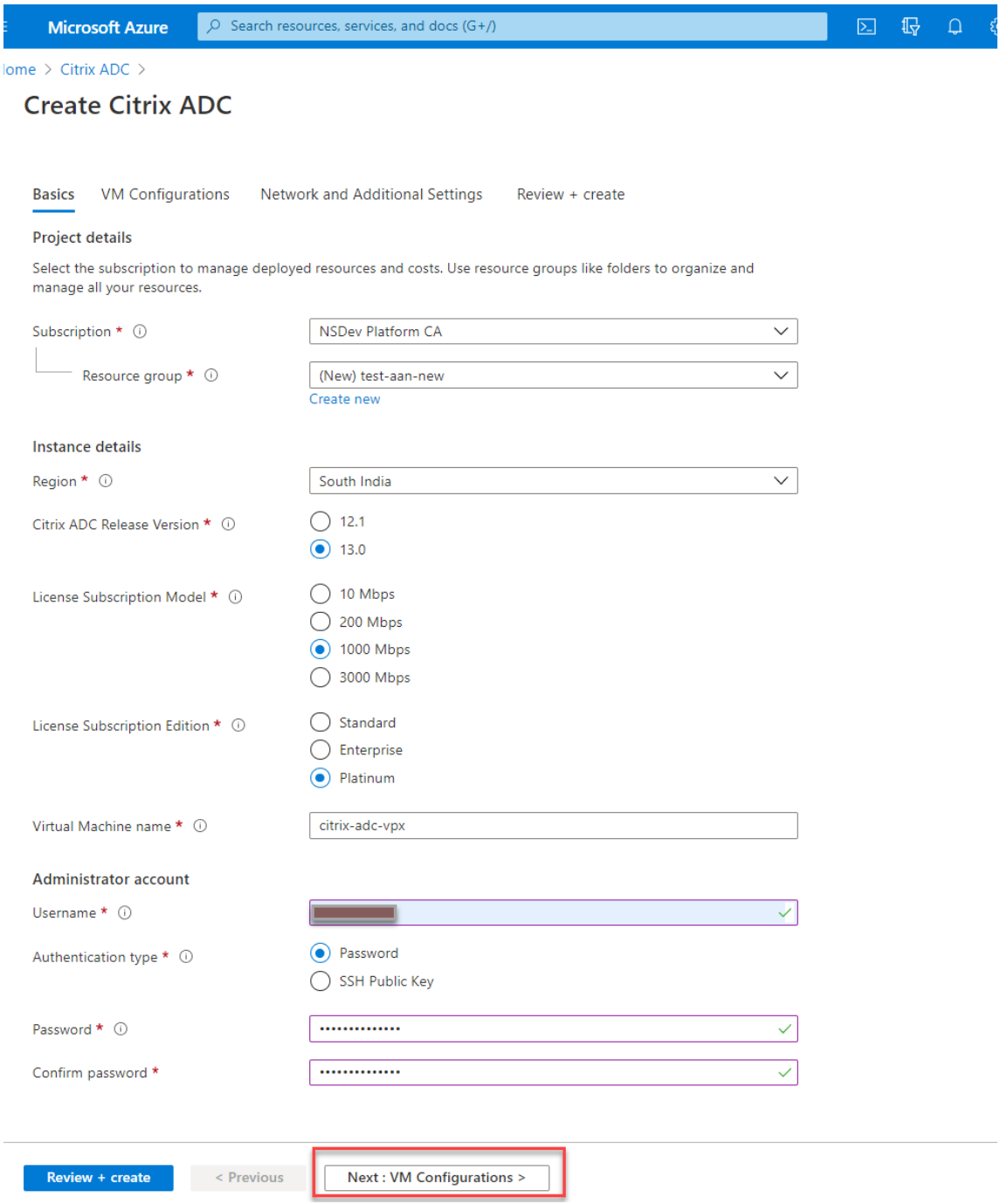


3. Select a non-FIPS Citrix ADC plan along with license, and click **Create**.



The **Create Citrix ADC** page appears.

4. In the **Basics** tab, create a Resource Group. Under the **Parameters** tab, enter details for the Region, Admin user name, Admin Password, license type (VM SKU), and other fields.



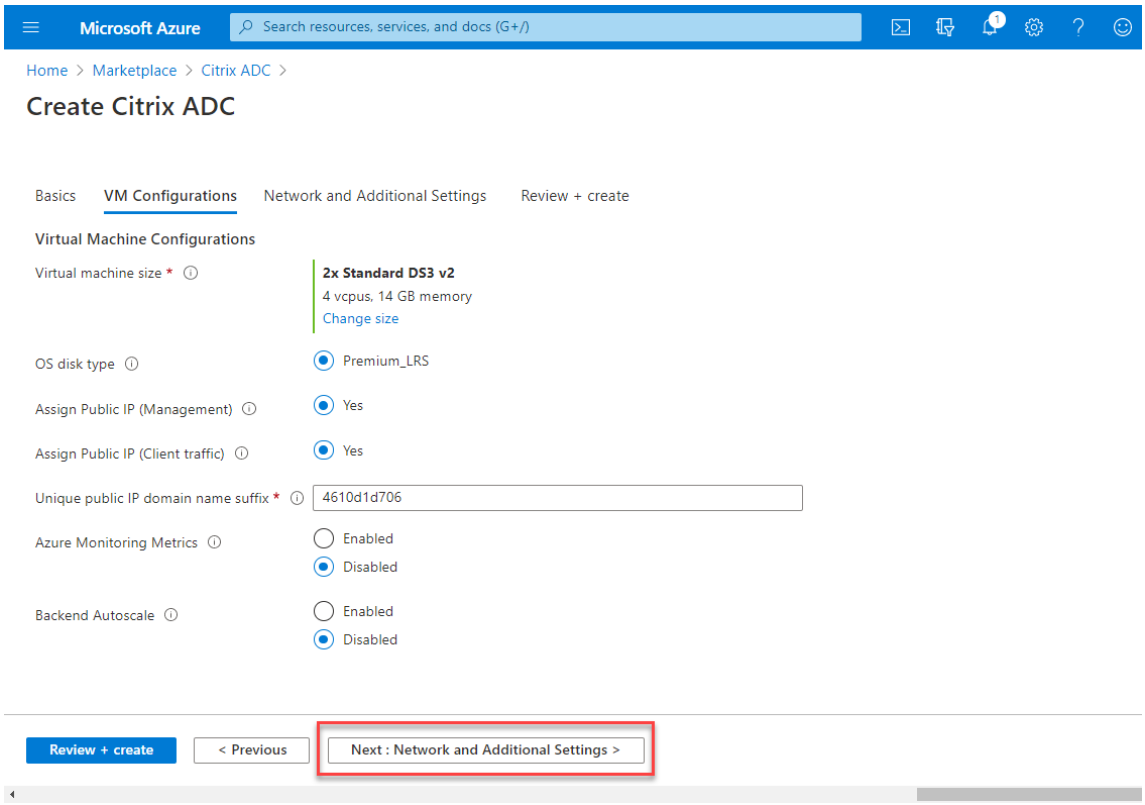
5. Click **Next : VM Configurations >**.

   On the **VM Configurations** page, perform the following:

   a) Configure public IP domain name suffix.

b) Enable or disable **Azure Monitoring Metrics**.

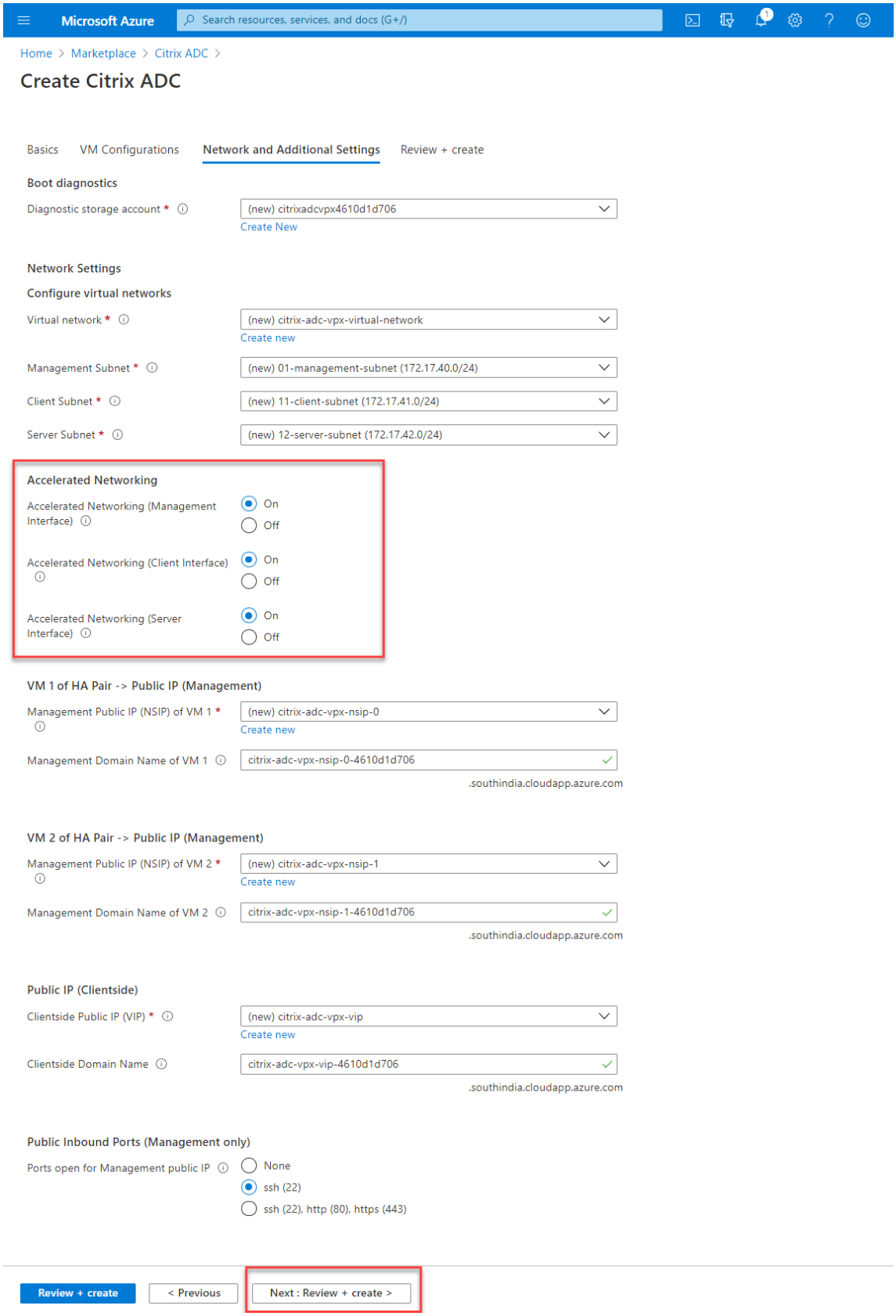
c) Enable or disable **Backend Autoscale**.



6. Click **Next: Network and Additional settings >**.

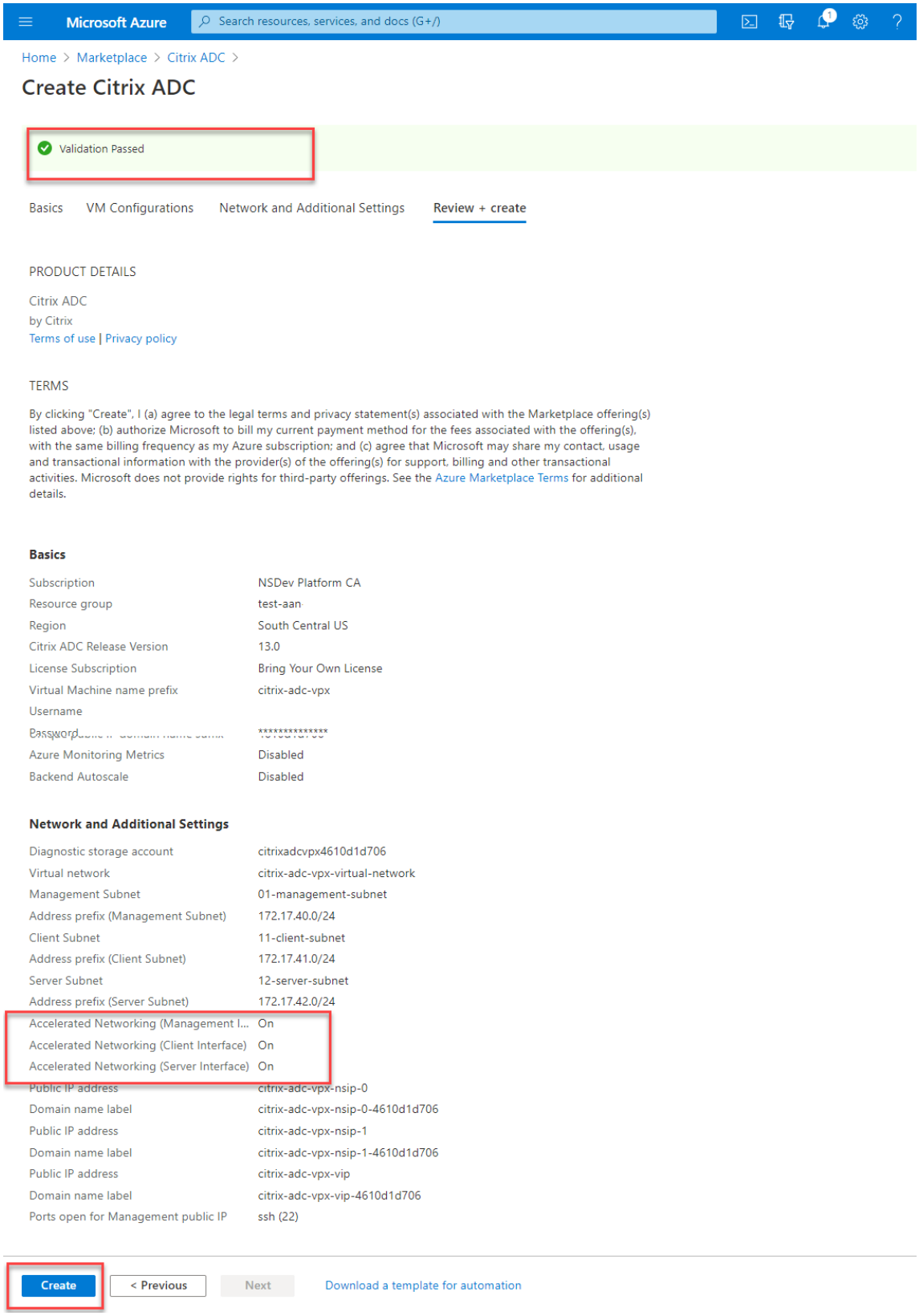
   On the **Network and Additional Settings** page, create a Boot diagnostics account and configure the network settings.

   Under the **Accelerated Networking** section, you have the option to enable or disable the accelerated networking separately for the Management interface, Client interface, and Server interface.

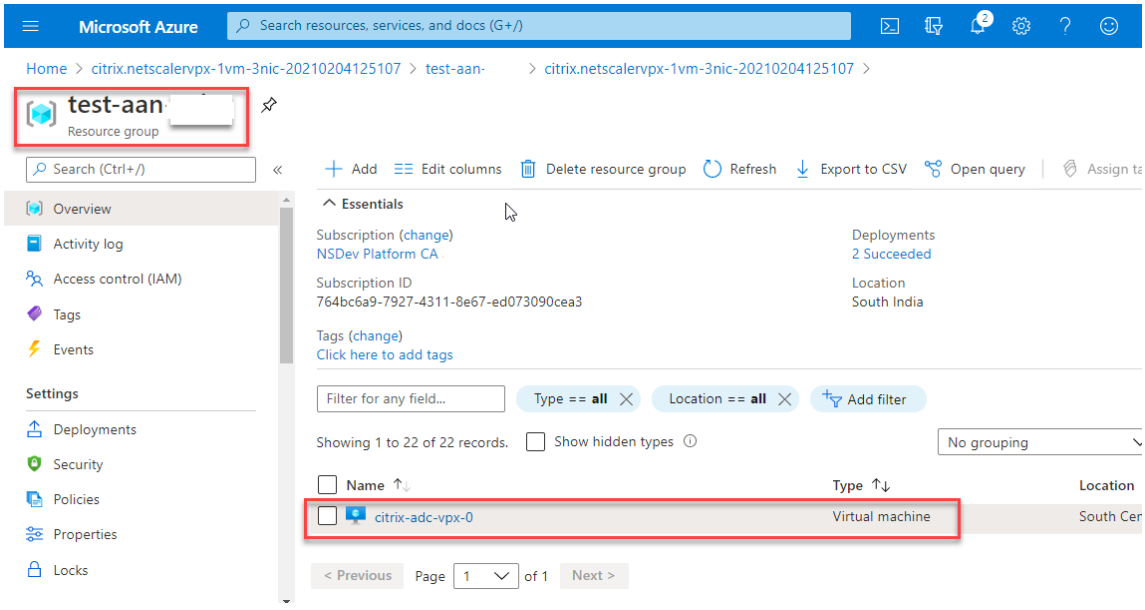


7. Click **Next: Review + create >**.

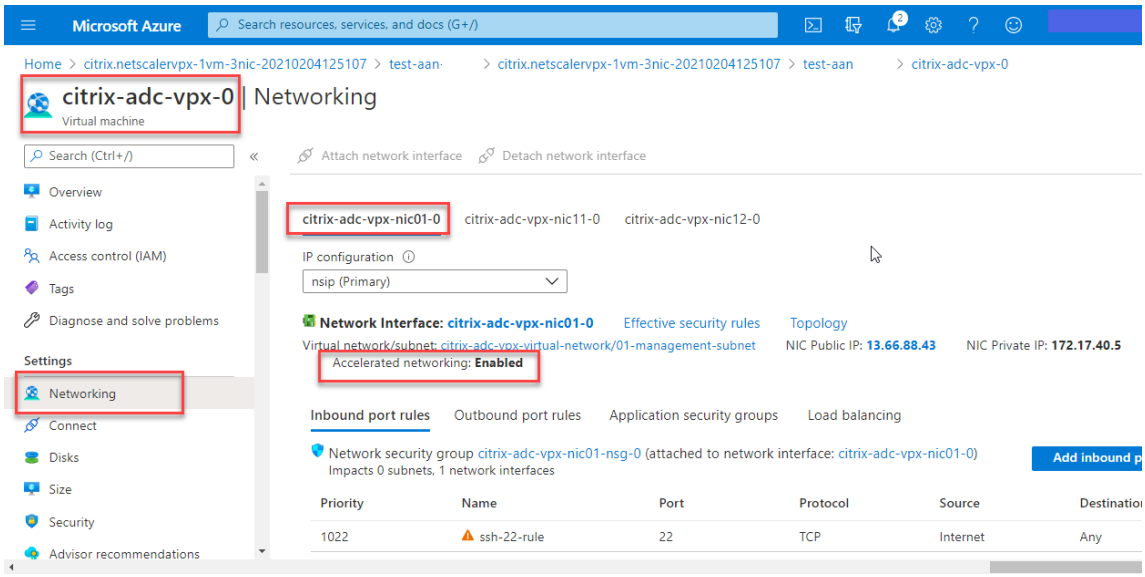
   After the validation is successful, review the basic settings, VM configurations, network and additional settings, and click **Create**. It might take some time for the Azure Resource Group to be created with the required configurations.



8. After the deployment is complete, select the **Resource Group** to see the configuration details.



9. To verify the Accelerated Networking configurations, select **Virtual machine > Networking**. The Accelerated Networking status is displayed as **Enabled** or **Disabled** for each NIC.



Enable accelerated networking using Azure PowerShell

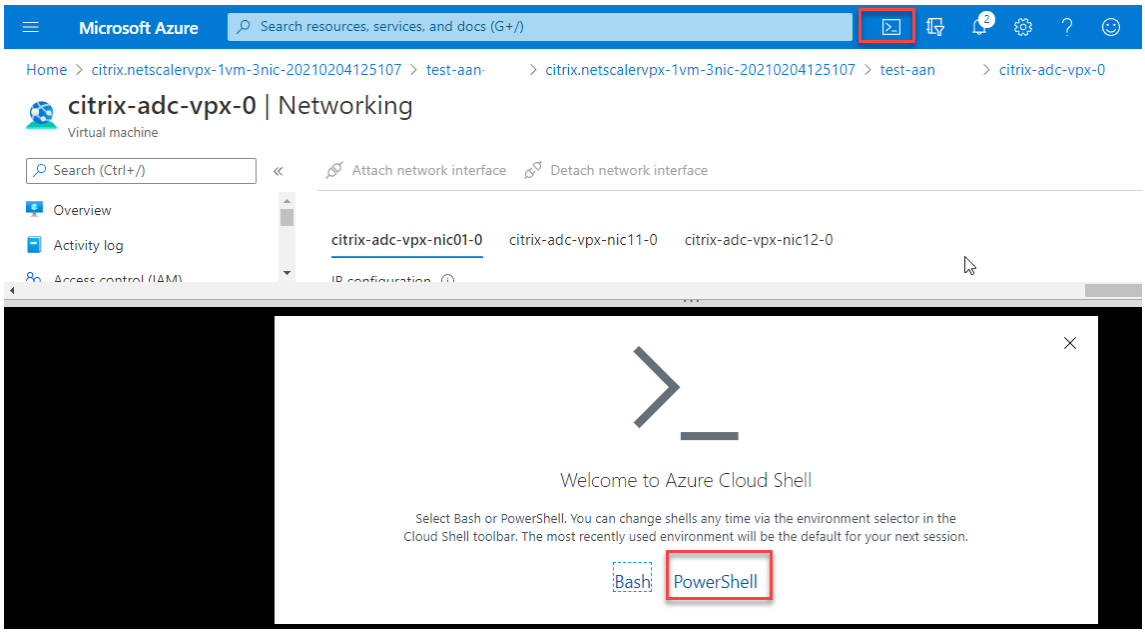
If you need to enable accelerated networking after the VM creation, you can do so using Azure PowerShell.

> Note:
>
> Ensure to stop the VM before you enable Accelerated Networking using Azure PowerShell.

Perform the following steps to enable accelerated networking by using Azure PowerShell.

1. Navigate to **Azure portal**, click the **PowerShell** icon on the right-hand top corner.

   > Note:
   >
   > If you are in the Bash mode, change to the PowerShell mode.



2. At the command prompt, run the following command:

```
az network nic update --name <nic-name> --accelerated-networking [true | false] --resource-group <resourcegroup-name>
<!--NeedCopy-->
```

The accelerated networking parameter accepts either of the following values:

- **True:** Enables accelerated networking on the specified NIC.
- **False:** Disables accelerated networking on the specified NIC.

To enable accelerated networking on a specific NIC:

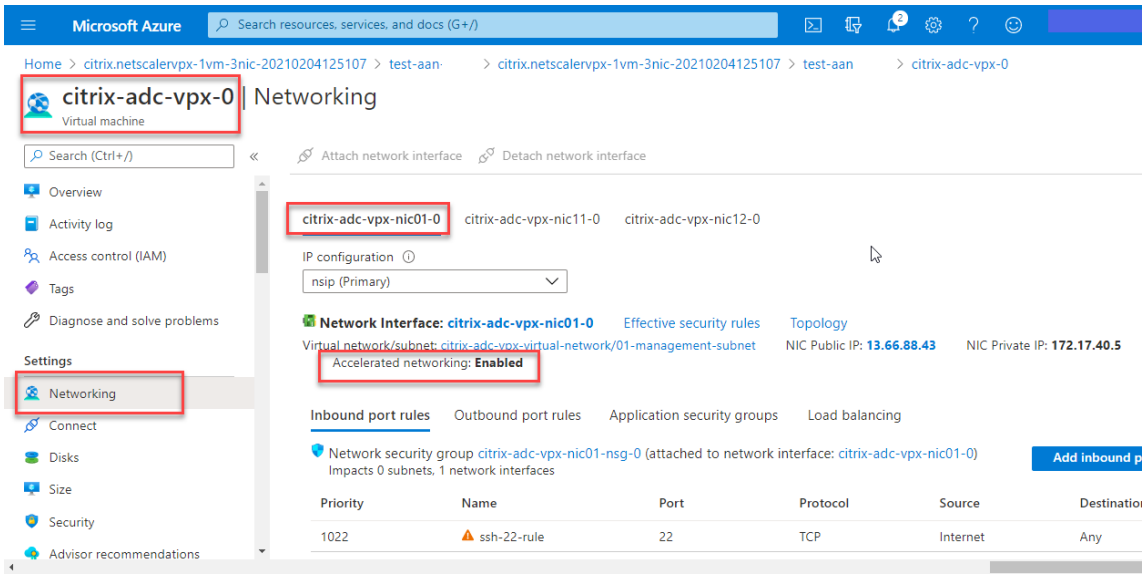
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking true --resource-group rsgp1-aan
<!--NeedCopy-->
```

To disable accelerated networking on a specific NIC:

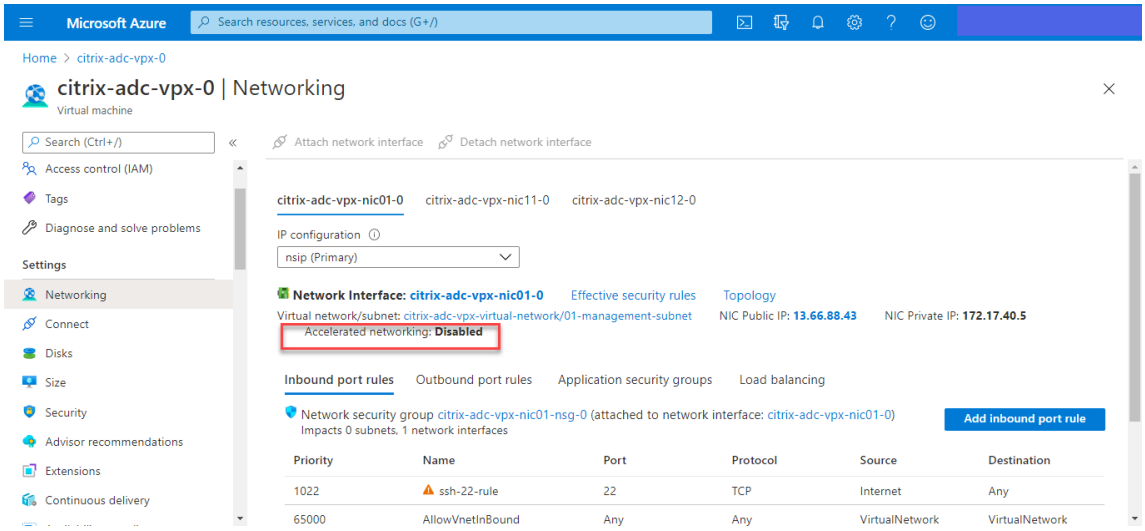
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking false --resource-group rsgp1-aan
<!--NeedCopy-->
```


3. To verify the Accelerated Networking status after the deployment is completed, Navigate to **VM > Networking**.

   In the following example, you can see that Accelerated Networking is **Enabled**.



   In the following example, you can see that Accelerated Networking is **Disabled**.



To verify accelerated networking on an interface by using FreeBSD Shell of Citrix ADC

You can log in to FreeBSD shell of Citrix ADC, and run the following commands to verify the accelerated networking status.

Example for ConnectX3 NIC:

The following example shows the “ifconfig” command output of the Mellanox ConnectX3 NIC. The “50/n” indicates the VF interfaces of the Mellanox ConnectX3 NICs. 0/1 and 1/1 indicates the PV inter-

faces of the Citrix ADC VPX instance. You can observe that both PV interface (1/1) and CX3 VF interface (50/1) have the same MAC addresses (00:22:48:1c:99:3e). This indicates that the two interfaces are bundled together.

```
root@nvr-us-cx3# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
0/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:0d:3a:98:71:be
        inet 172.16.27.11 netmask 0xffffff00 broadcast 172.16.27.255
        inet6 fe80::20d:3aff:fe98:71be%0/1 prefixlen 64 autoconf scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
1/1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
50/1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

options=900b8<VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,VLAN_HWFILTER,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (<unknown subtype>)
        status: active
```

Example for ConnectX4 NIC:

The following example shows the “ifconfig” command output of the Mellanox ConnectX4 NIC. The “100/n” indicates the VF interfaces of the Mellanox ConnectX4 NICs. 0/1, 1/1, and 1/2 indicates the PV interfaces of Citrix ADC VPX instance.

You can observe that both PV interface (1/1) and CX4 VF interface (100/1) have the same MAC addresses (00:0d:3a:9b:f2:1d). This indicates that the two interfaces are bundled together. Similarly, the PV interface (1/2) and CX4 VF interface (100/2) have the same MAC addresses (00:0d:3a:1e:d2:23).

```
root@SmartNIC-CX4-NS-DUT-NEW1# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
1/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>

        ether 00:0d:3a:9b:f2:1d

        inet 10.0.1.29 netmask 0xffffff00 broadcast 10.0.1.255
        inet6 fe80::20d:3aff:fe9b:f21d%0/1 prefixlen 64 autoconf scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active

1/2: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>

        ether 00:0d:3a:1e:d2:23

        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active

100/1: flags=8a03<UP,BROADCAST,ALLMULTI,SIMPLEX,MULTICAST> metric 0 mtu 1500

        ether 00:0d:3a:9b:f2:1d

        media: Ethernet autoselect <full-duplex,rxpause,txpause> (autoselect
<full-duplex,rxpause>)
        status: active

100/2: flags=8a03<UP,BROADCAST,ALLMULTI,SIMPLEX,MULTICAST> metric 0 mtu 1500

        ether 00:0d:3a:1e:d2:23

        media: Ethernet autoselect <full-duplex,rxpause,txpause> (autoselect
<full-duplex,rxpause>)
        status: active
```

To verify accelerated networking on an interface by using ADC CLI

Example for ConnectX3 NIC:

The following show interface command output indicates that the PV interface 1/1 is bundled with virtual function 50/1, which is an SR-IOV VF NIC. The MAC addresses of both 1/1 and 50/1 NICs are the same. After accelerated networking is enabled, the data of the 1/1 interface is sent through datapath of the 50/1 interface, which is a ConnectX3 interface. You can see that the “show interface” output of PV interface (1/1) points to the VF (50/1). Similarly, the “show interface” output of VF interface (50/1) points to the PV interface (1/1).


```
> show interface 1/1

        Interface 1/1 (NetScaler Virtual Interface, SmartNIC, VF 50/1 Datapath 50/1) #1
        flags=0xe060 <ENABLED, UP, UP, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m07s
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.

Done

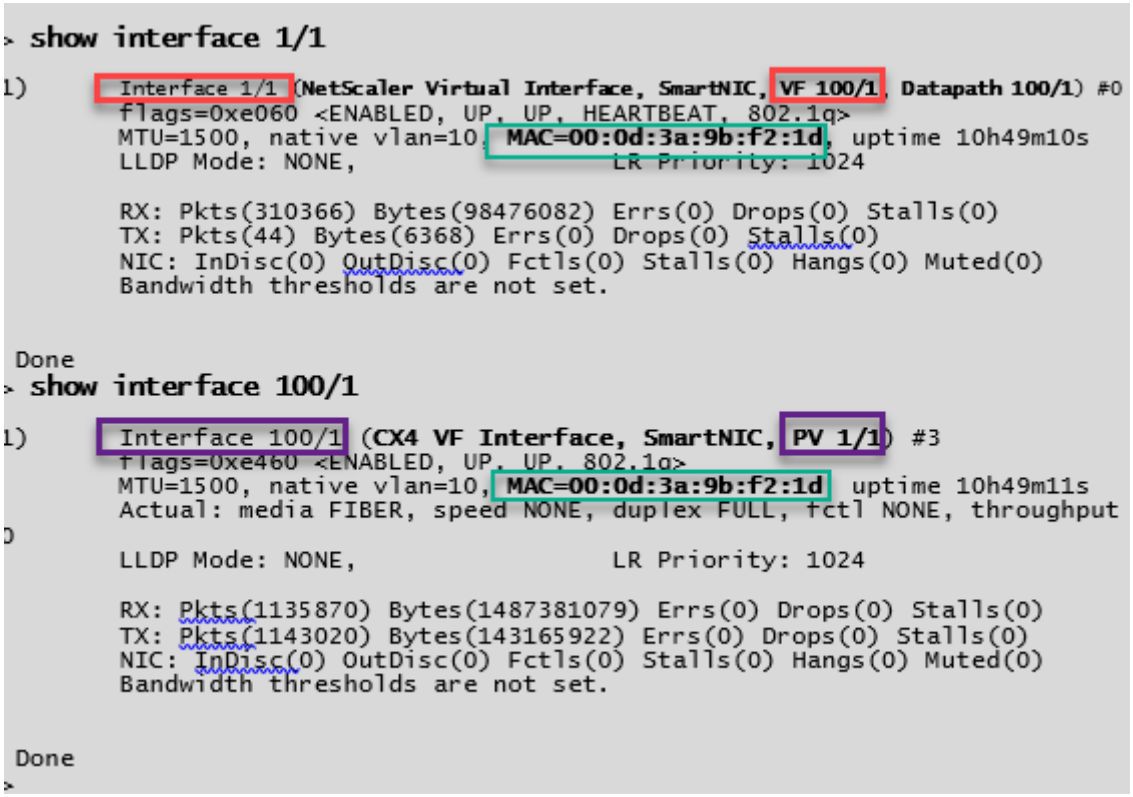
> show interface 50/1

      Interface 50/1 (CX3 VF Interface, SmartNIC, PV 1/1) #2
        flags=0xe460 <ENABLED, UP, UP, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m08s
        Actual: media NONE, speed 50000, duplex FULL, fctl NONE, throughput 50000
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
```

Example for ConnectX4 NIC:

The following show interface command output indicates that the PV interface 1/1 is bundled with virtual function 100/1, which is an SR-IOV VF NIC. The MAC addresses of both 1/1 and 100/1 NICs are the same. After accelerated networking is enabled, the data of 1/1 interface is sent through the data path of 100/1 interface, which is a ConnectX4 interface. You can see that the “show interface” output of PV interface (1/1) points to the VF (100/1). Similarly, the “show interface” output of VF interface (100/1) points to the PV interface (1/1).



Points to note in Citrix ADC

- PV interface is considered as the primary or main interface for all the necessary operations. Configurations must be performed on PV interfaces only.
- All the 'set' operations on a VF interface are blocked except the following:
  - enable interface
  - disable interface
  - reset interface
  - clear stats

  > Note:
  >
  > Citrix recommends that you do not perform any operations on the VF interface.

- You can verify the binding of PV interface with VF interface using the `show interface` command.

Configure a VLAN to a PV interface

When a PV interface is bound to a VLAN, the associated accelerated VF interface is also bound to the same VLAN as the PV interface. In this example, the PV interface (1/1) is bound to VLAN (20). The VF interface (100/1) that is bundled with the PV interface (1/1) is also bound to VLAN 20.

Example:

1. Create a VLAN.

```
add vlan 20
<!--NeedCopy-->
```

2. Bind a VLAN to the PV interface.

```
bind vlan 20  - ifnum 1/1

show vlan

1)  VLAN ID: 1
    Link-local IPv6 addr: fe80::20d:3aff:fe9b:f21d/64
    Interfaces : LO/1

2)  VLAN ID: 10     VLAN Alias Name:
    Interfaces : 0/1 100/1
    IPs : 10.0.1.29  Mask: 255.255.255.0

3)  VLAN ID: 20     VLAN Alias Name:
    Interfaces :  1/1 100/2

<!--NeedCopy-->
```

> Note
>
> VLAN binding operation is not permitted on an accelerated VF interface.

```
bind vlan 1 -ifnum 100/1
ERROR: Operation not permitted
<!--NeedCopy-->
```

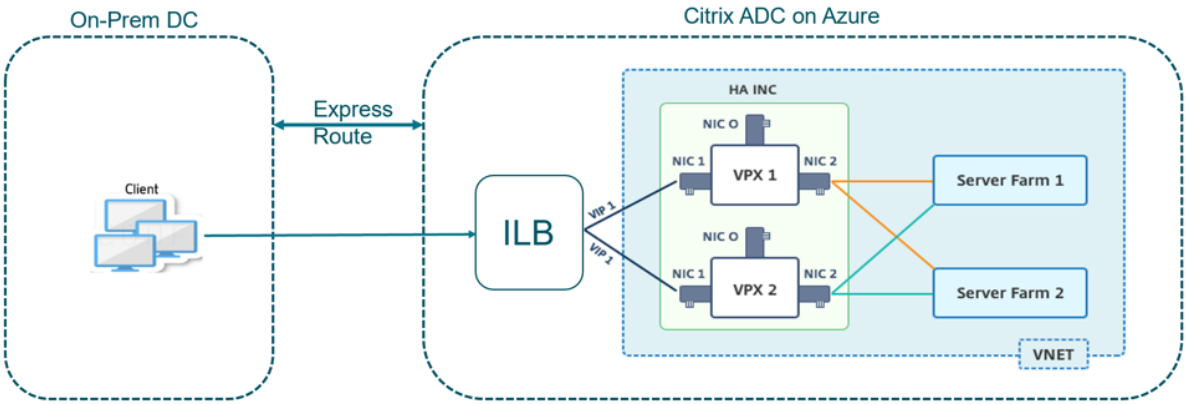
Configure HA-INC nodes by using the Citrix high availability template with Azure ILB

November 10, 2021

You can quickly and efficiently deploy a pair of VPX instances in HA-INC mode by using the standard template for intranet applications. The Azure internal load balancer (ILB) uses an internal or private IP address for the front end as shown in Figure 1. The template creates two nodes, with three sub-

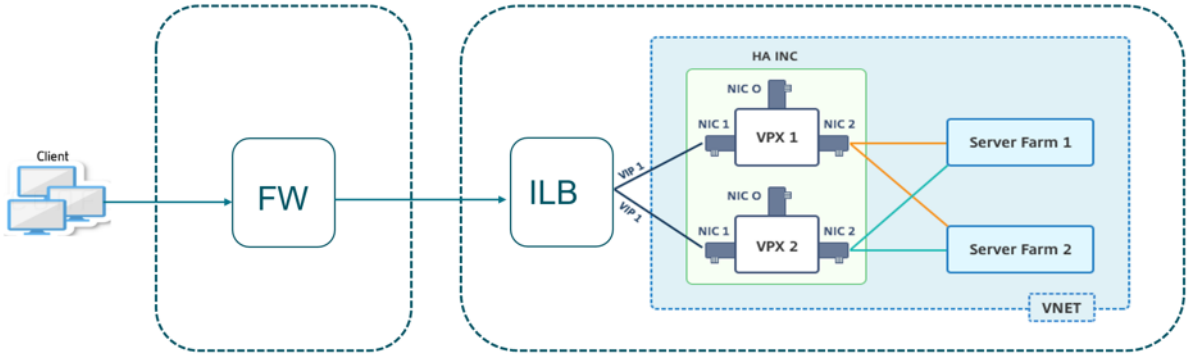
nets and six NICs. The subnets are for management, client, and server-side traffic with each subnet belonging to a different NIC on each device.

Figure 1: Citrix ADC HA pair for clients in an internal network



You can also use this deployment when the Citrix ADC HA pair is behind a firewall as shown in Figure 2. The public IP address belongs to the firewall and is NAT'd to the front-end IP address of the ILB.

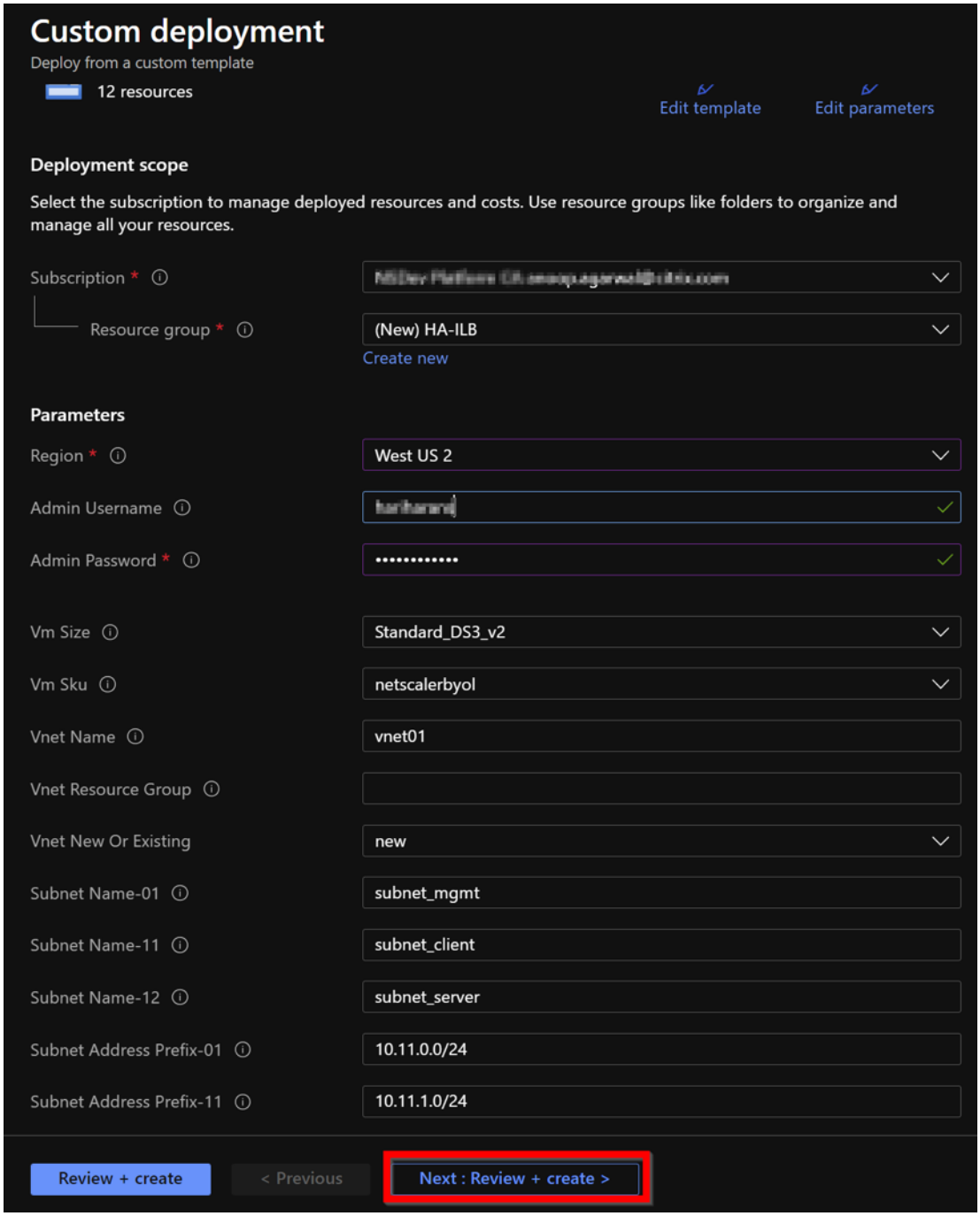
Figure 2: Citrix ADC HA pair with firewall having public IP address



You can get the Citrix ADC HA pair template for intranet applications at the Azure portal.

Complete the following steps to launch the template and deploy a high availability VPX pair by using Azure Availability Sets.

1. From the Azure portal, navigate to the **Custom deployment** page.
2. The **Basics** page appears. Create a Resource Group. Under the **Parameters** tab, enter details for the Region, Admin user name, Admin Password, license type (`VM sku`), and other fields.

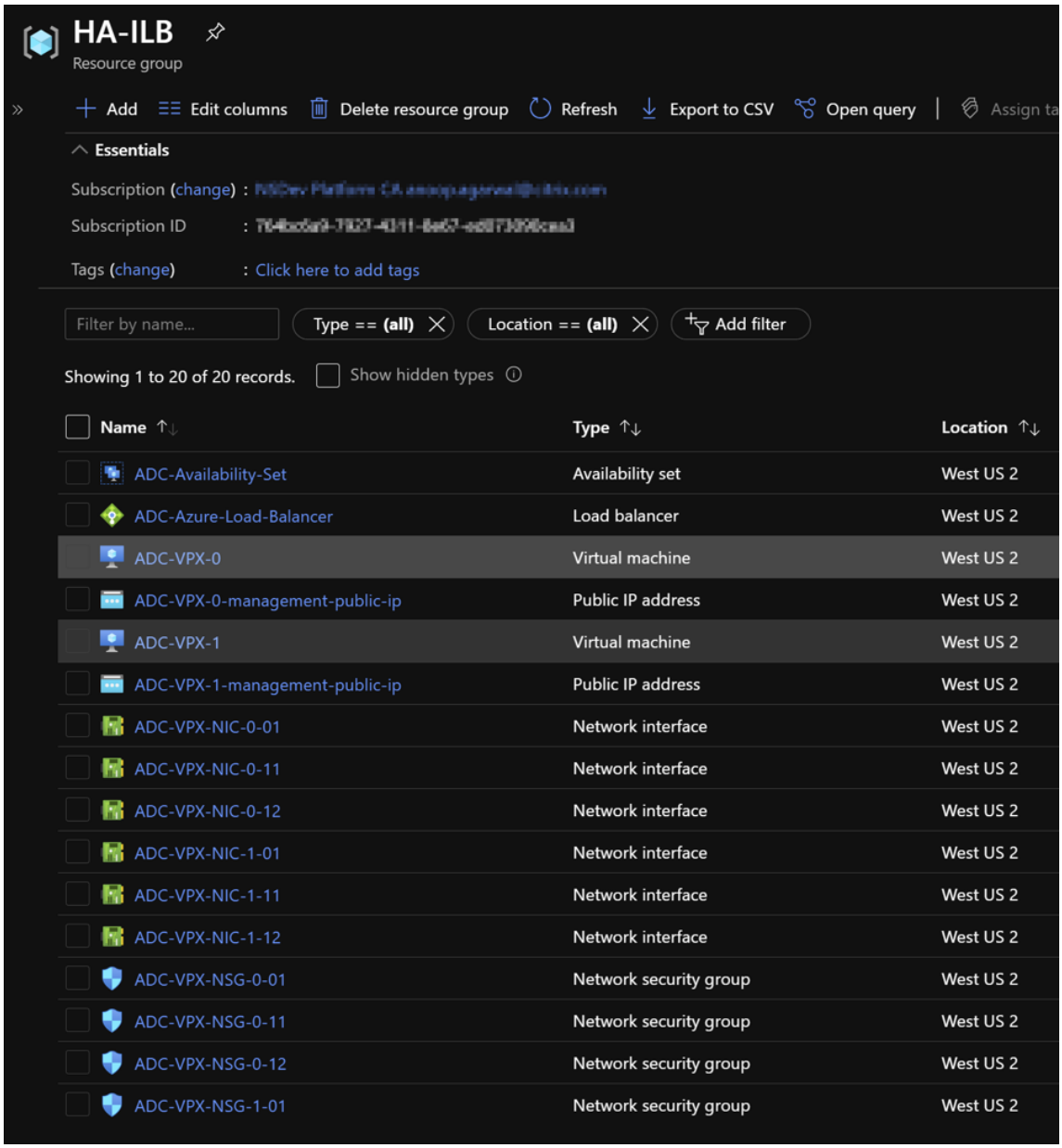


3. Click **Next : Review + create >**.

   It might take a moment for the Azure Resource Group to be created with the required configurations. After completion, select the Resource Group in the Azure portal to see the configuration details, such as LB rules, back-end pools, health probes. The high availability pair appears as ADC-VPX-0 and ADC-VPX-1.

   If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.

Once the required configuration is complete, the following resources are created.



4. You must log on to **ADC-VPX-0** and **ADC-VPX-1** nodes to validate the following configuration:

   - NSIP addresses for both nodes must be in the management subnet.
   - On the primary (ADC-VPX-0) and secondary (ADC-VPX-1) nodes, you must see two SNIP addresses. One SNIP (client subnet) is used for responding to ILB probes and the other SNIP (server subnet) is used for back-end server communication.

   > Note
   >
   > In the HA-INC mode, the SNIP address of the ADC-VPX-0 and ADC-VPX-1 VMs are different while in the same subnet, unlike with the classic on-premises ADC HA deployment where

> both are the same.
> To support deployments when the VPX pair SNIP is in different subnets, or anytime the VIP is not in the same subnet as a SNIP, you must either enable Mac-Based Forwarding (MBF), or add a static host route for each VIP to each VPX node.

On the primary node (ADC-VPX-0)

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.5        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

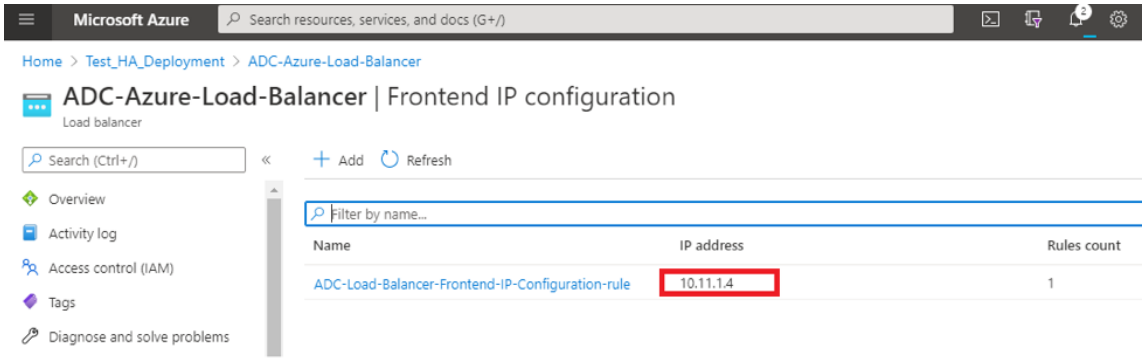
```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.5 (ADC-VPX-0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:20:26 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.4
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

On the secondary node (ADC-VPX-1)

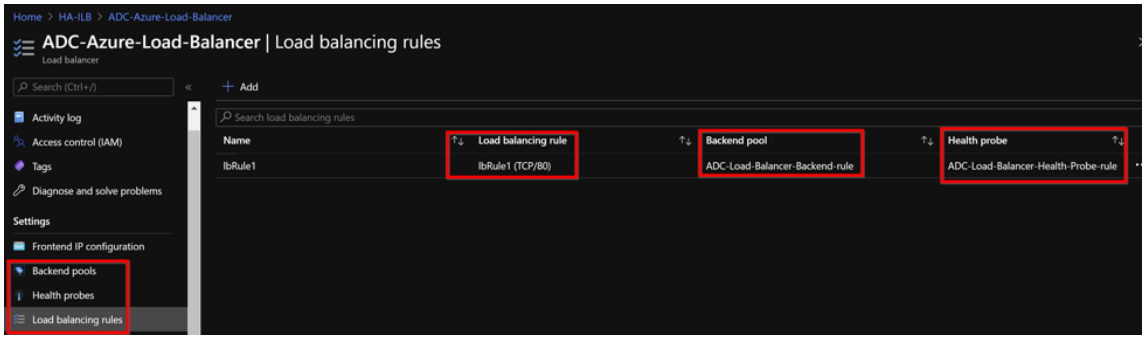

```
> sh ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.11.0.4        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.6        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.4 (ADC-VPX-1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:24:18 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.5
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

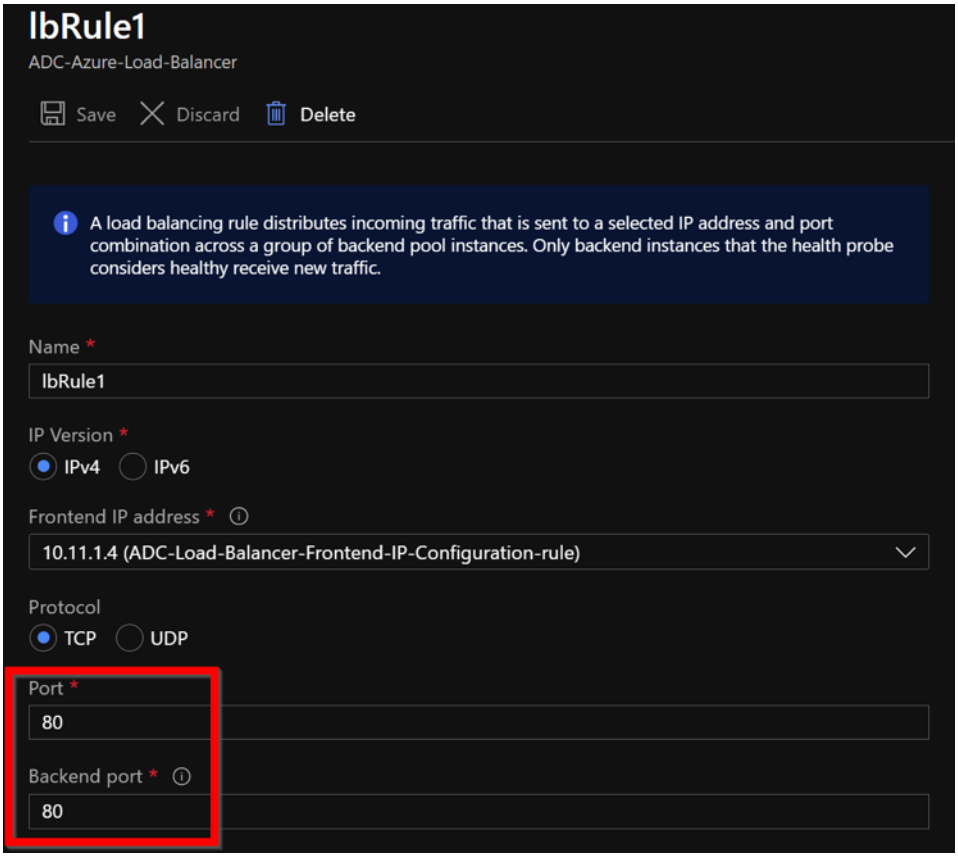
5. After the primary and secondary nodes are UP and the Synchronization status is **SUCCESS**, you must configure the load balancing virtual server or the gateway virtual server on the primary node (ADC-VPX-0) with the private floating IP (FIP) address of the ADC Azure load balancer. For more information, see the Sample configuration section.
6. To find the private IP address of ADC Azure load balancer, navigate to **Azure portal > ADC Azure Load Balancer > Frontend IP configuration**.



7. In the **Azure Load Balancer** configuration page, the ARM template deployment helps create the LB rule, back-end pools, and health probes.

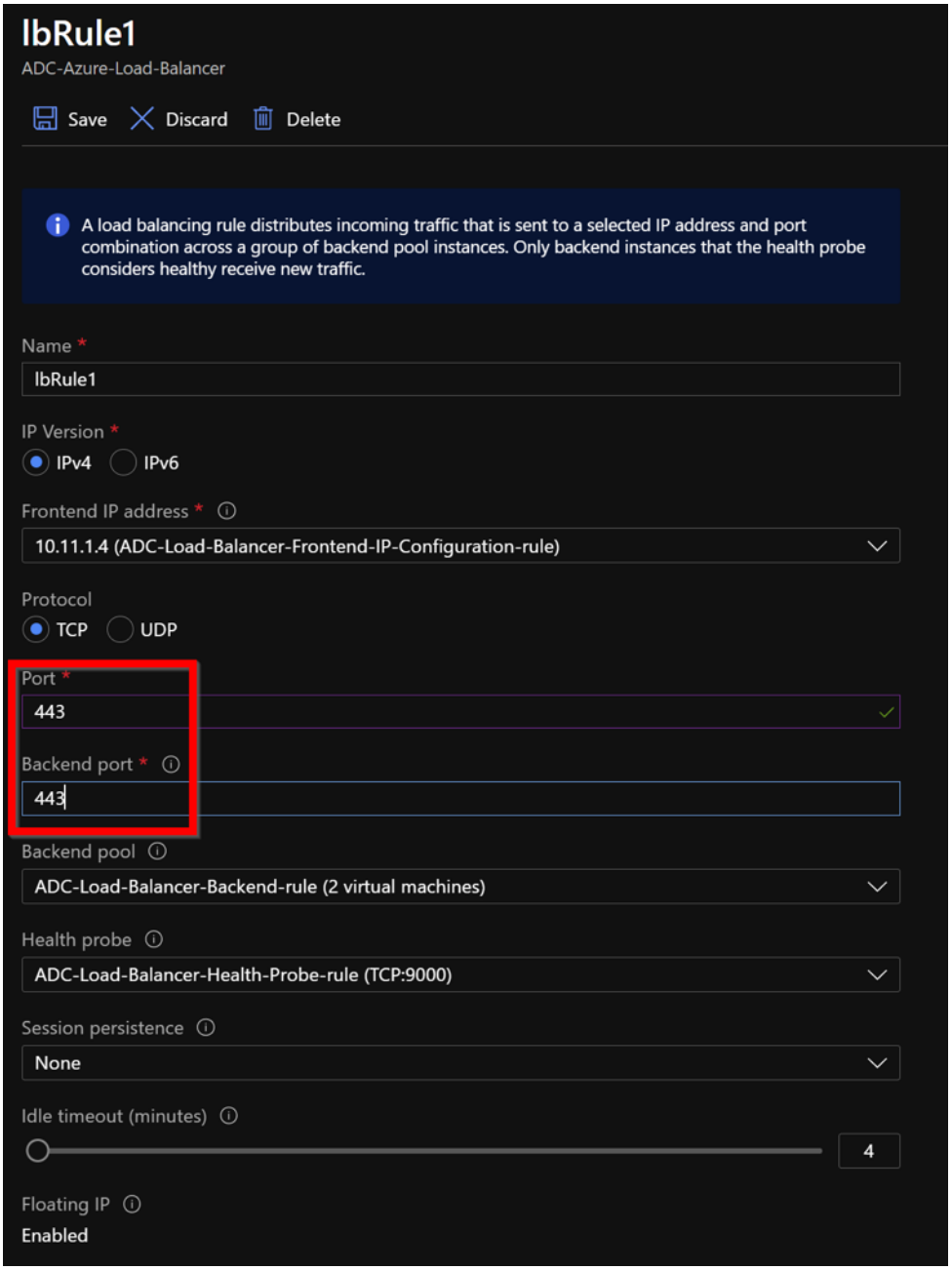


   - The LB Rule (LbRule1) uses port 80, by default.



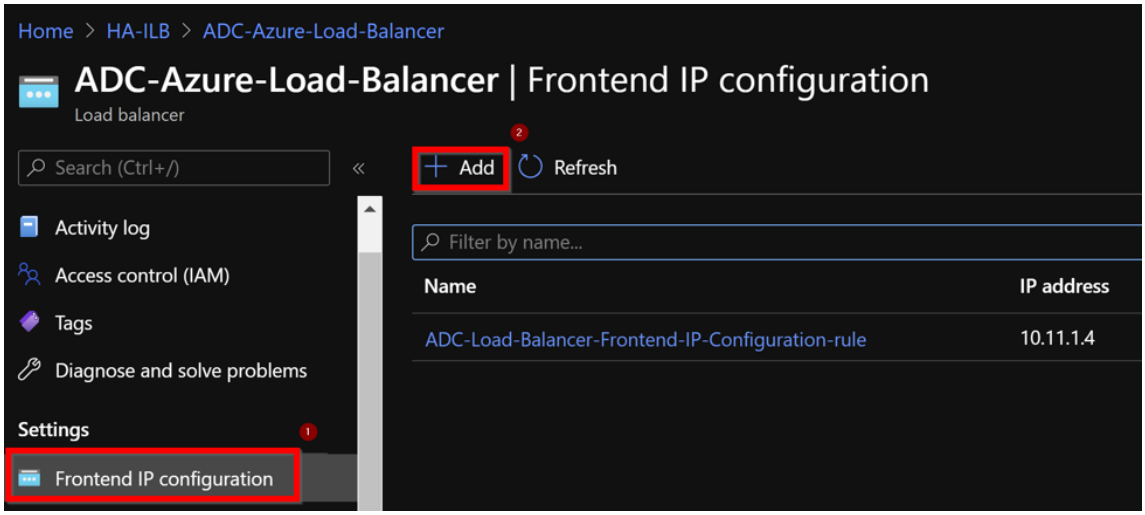
- Edit the rule to use port 443, and save the changes.

> Note
>
> For enhanced security, Citrix recommends you to use SSL port 443 for LB virtual server or Gateway virtual server.

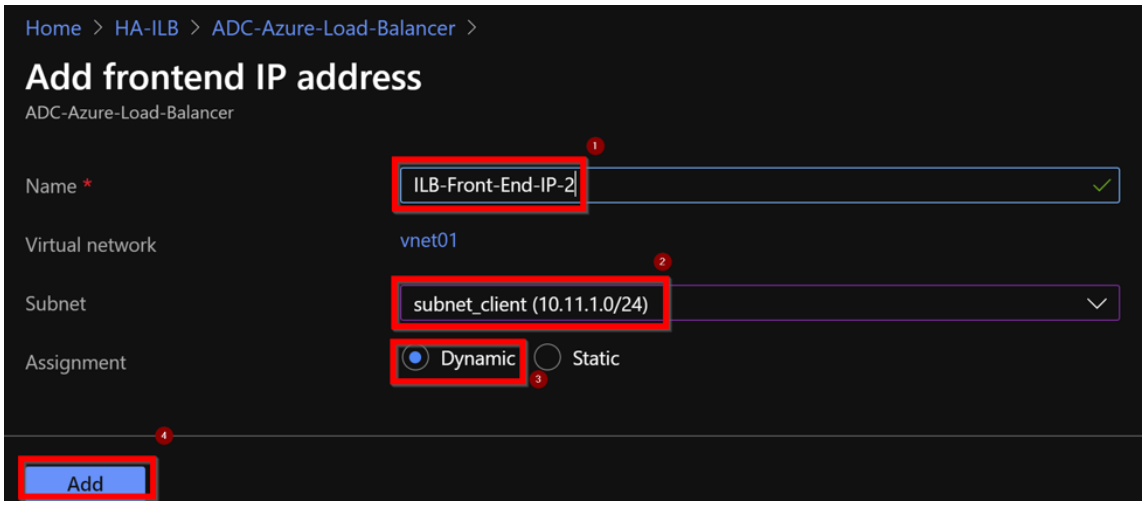


To add more VIP addresses on the ADC, perform the following steps:

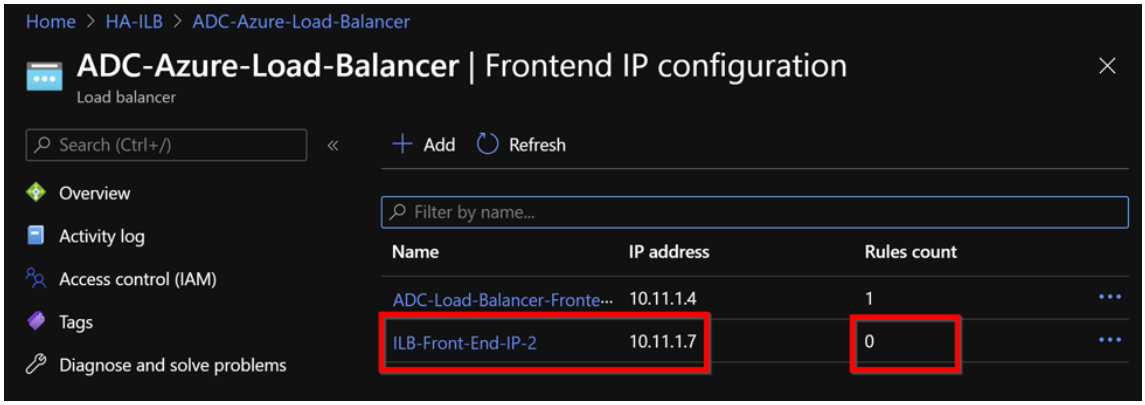
1. Navigate to **Azure Load Balancer > Frontend IP configuration**, and click **Add** to create a new internal load balancer IP address.



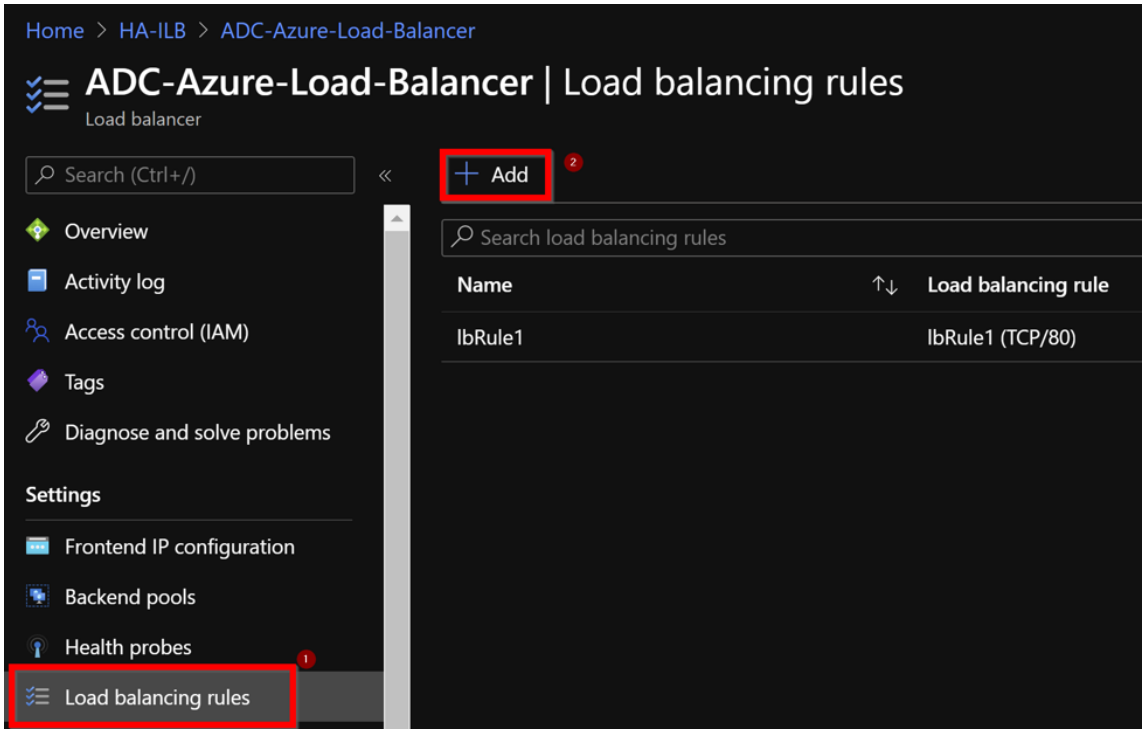
2. In the **Add frontend IP address** page, enter a name, choose the client subnet, assign either dynamic or static IP address, and click **Add**.



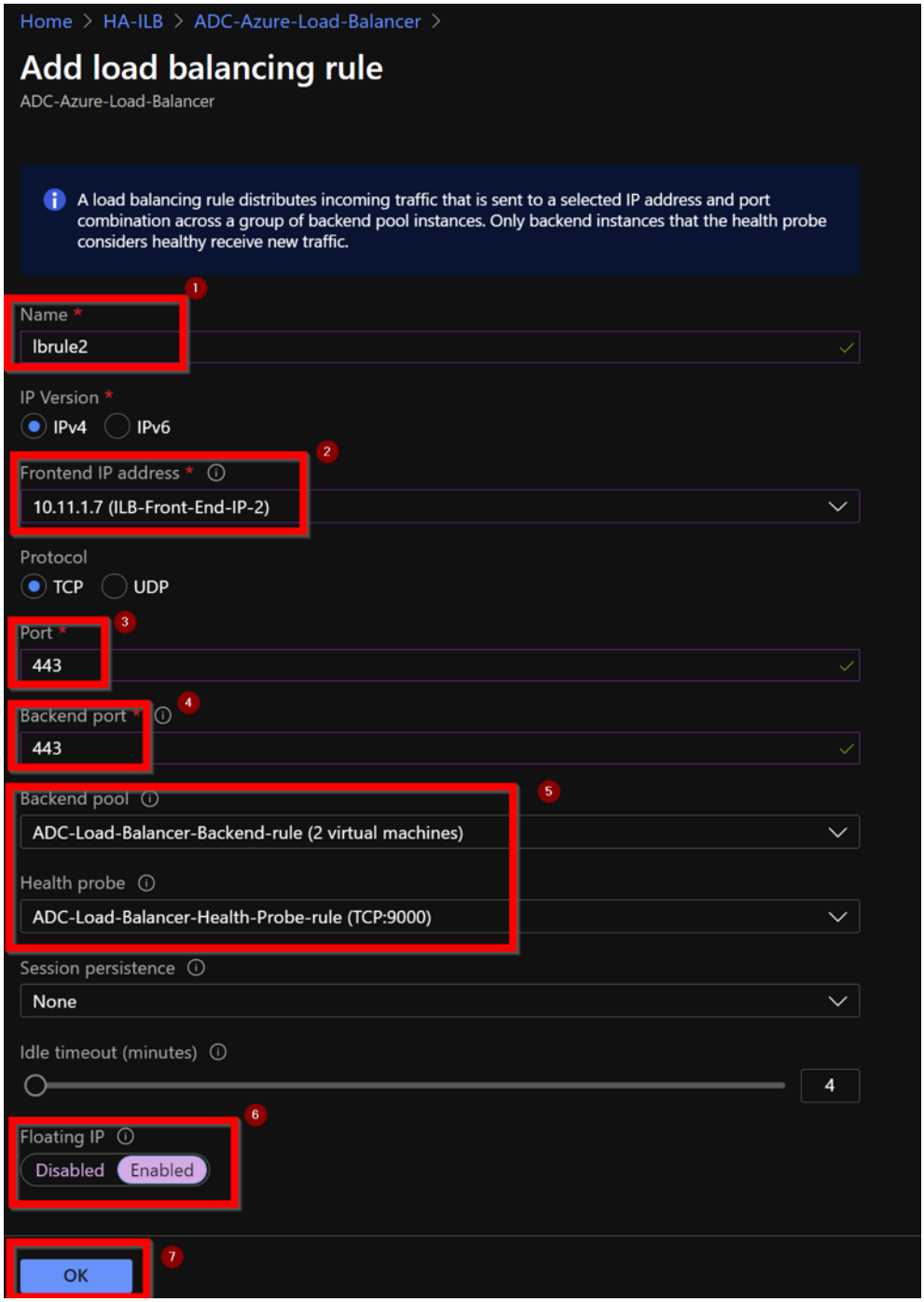
3. The front-end IP address is created but an LB Rule is not associated. Create a new load balancing rule, and associate it with the front-end IP address.



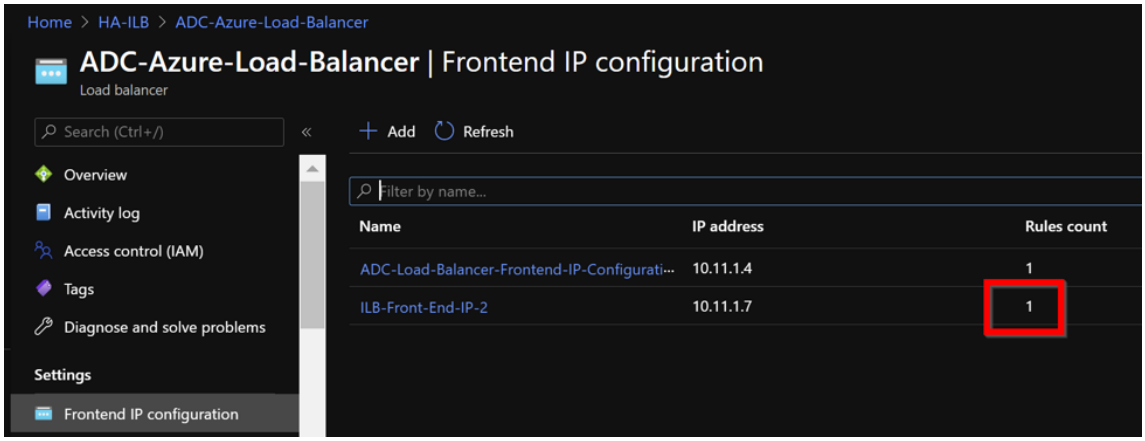
4. In the **Azure Load Balancer** page, select **Load balancing rules**, and then click **Add**.



5. Create a new LB Rule by choosing the new front-end IP address and the port. **Floating IP** field must be set to **Enabled**.



6. Now the **Frontend IP configuration** shows the LB rule that is applied.



Sample configuration

To configure a gateway VPN virtual server and load balancing virtual server, run the following commands on the primary node (ADC-VPX-0). The configuration auto synchronizes to the secondary node (ADC-VPX-1).

Gateway sample configuration

```
enable feature aaa LB SSL SSLVPN
enable ns mode MBF
add vpn vserver vpn_ssl SSL 10.11.1.4 443
add ssl certKey ckp -cert wild-cgwsanity.cer -key wild-cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

Load balancing sample configuration

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 10.11.1.7 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

You can now access the load balancing or VPN virtual server using the fully qualified domain name (FQDN) associated with the internal IP address of ILB.

See the **Resources** section for more information about how to configure the load-balancing virtual server.

Resources:

The following links provide additional information related to HA deployment and virtual server configuration:

- Configuring high availability nodes in different subnets
- Set up basic load balancing

Related resources:

- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands
- Configuring GSLB on Active-Standby HA Deployment on Azure

Configure HA-INC nodes by using the Citrix high availability template for internet-facing applications

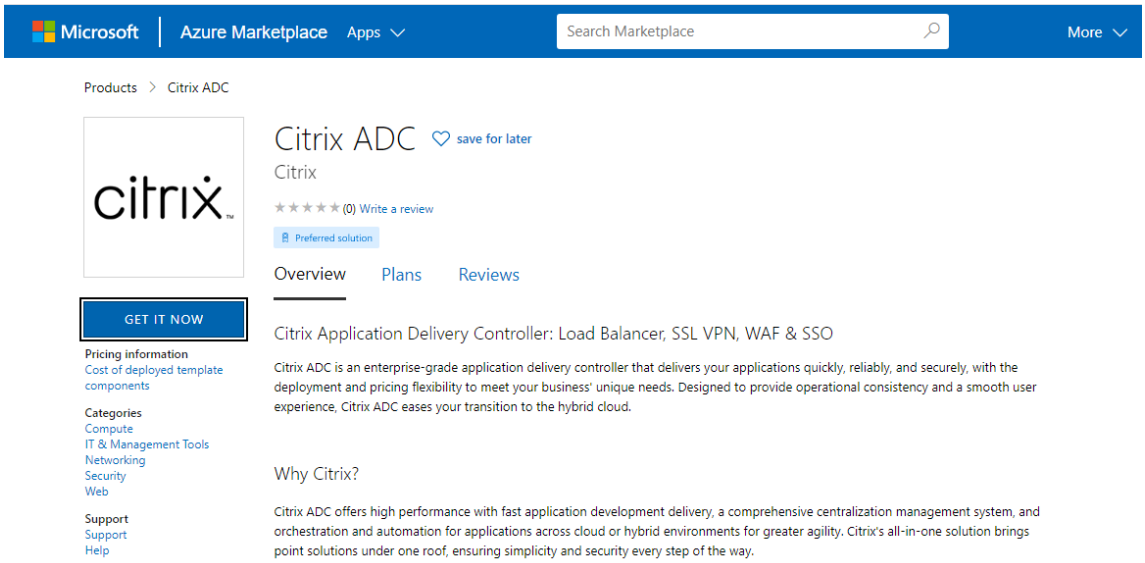
September 14, 2021

You can quickly and efficiently deploy a pair of VPX instances in HA-INC mode by using the standard template for internet-facing applications. The Azure load balancer (ALB) uses a public IP address for the front end. The template creates two nodes, with three subnets and six NICs. The subnets are for management, client, and server-side traffic. Each subnet has two NICs for both the VPX instances.

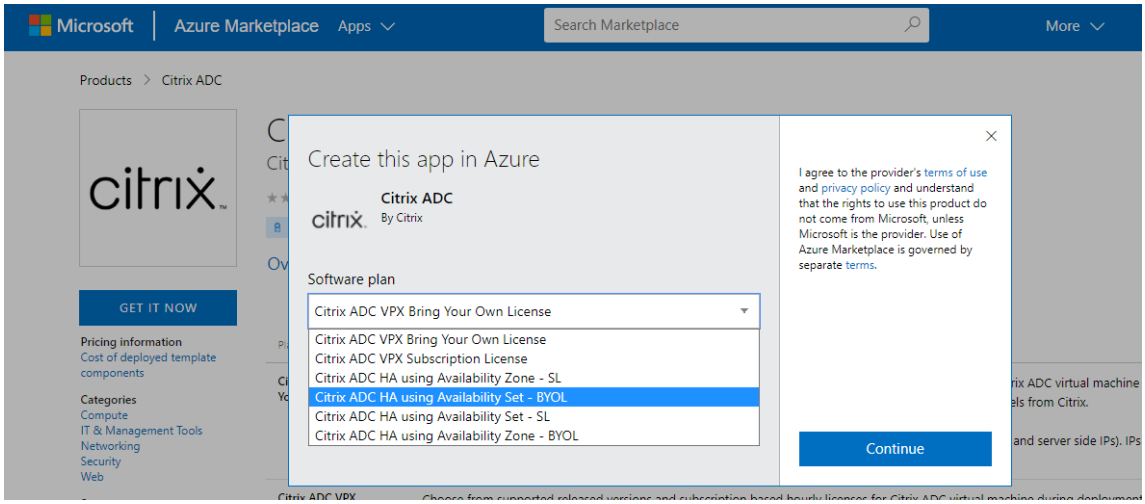
You can get the Citrix ADC HA pair template for internet-facing applications at the Azure Marketplace.

Complete the following steps to launch the template and deploy a high availability VPX pair by using Azure availability sets or availability zone.

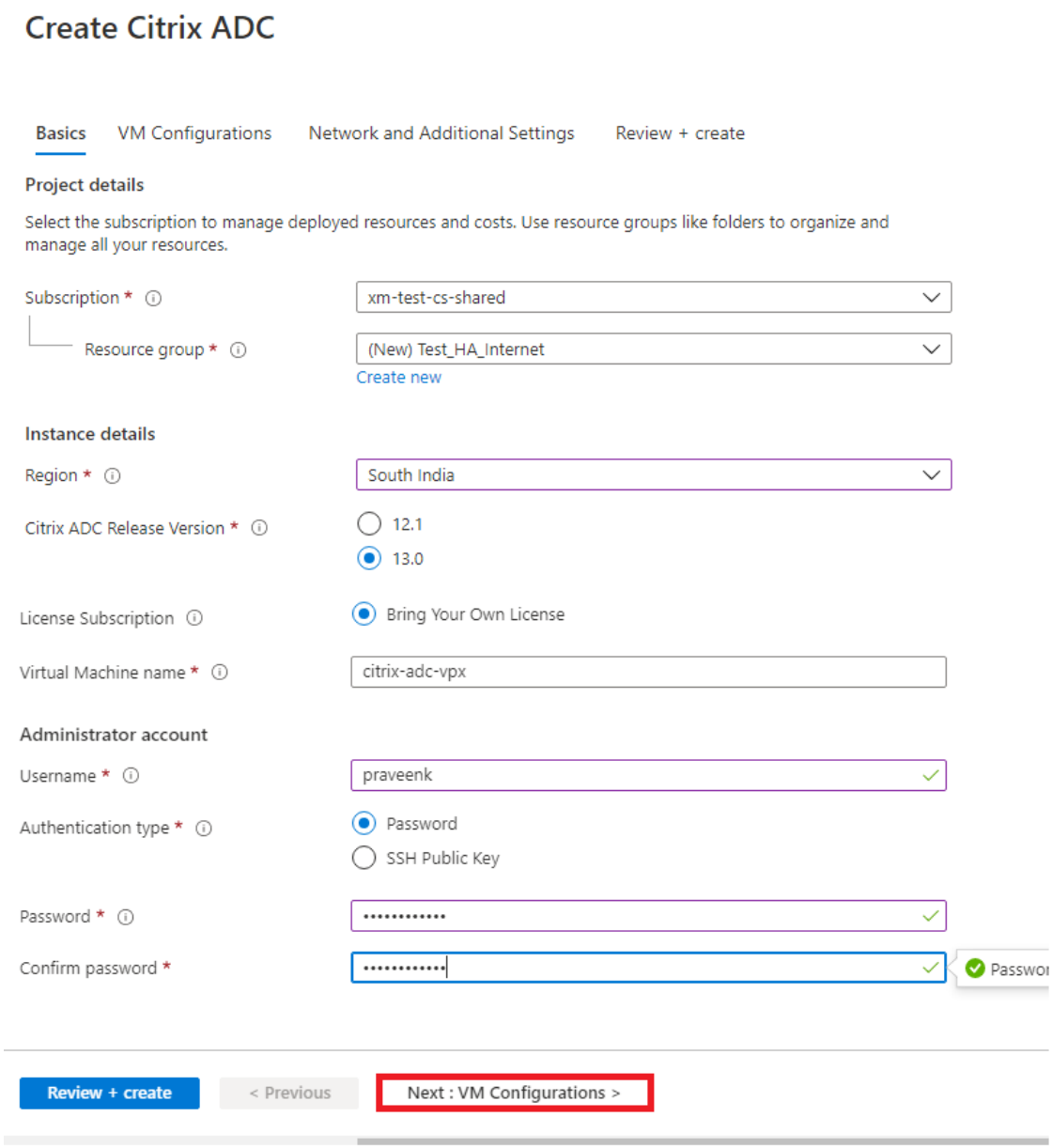
1. From the Azure Marketplace, search **Citrix ADC**.
2. Click **GET IT NOW**.



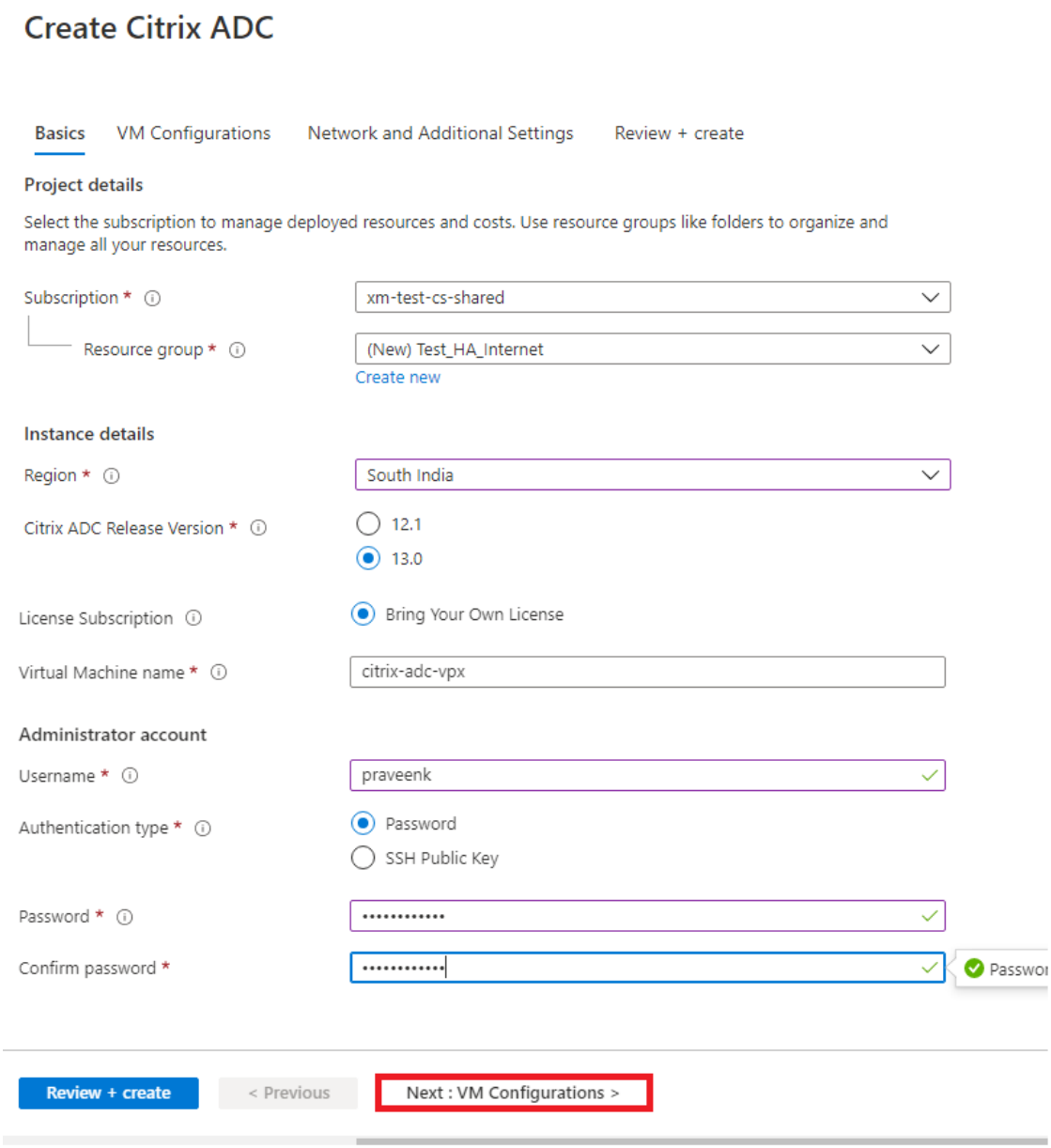
3. Select the required HA deployment along with license, and click **Continue**.



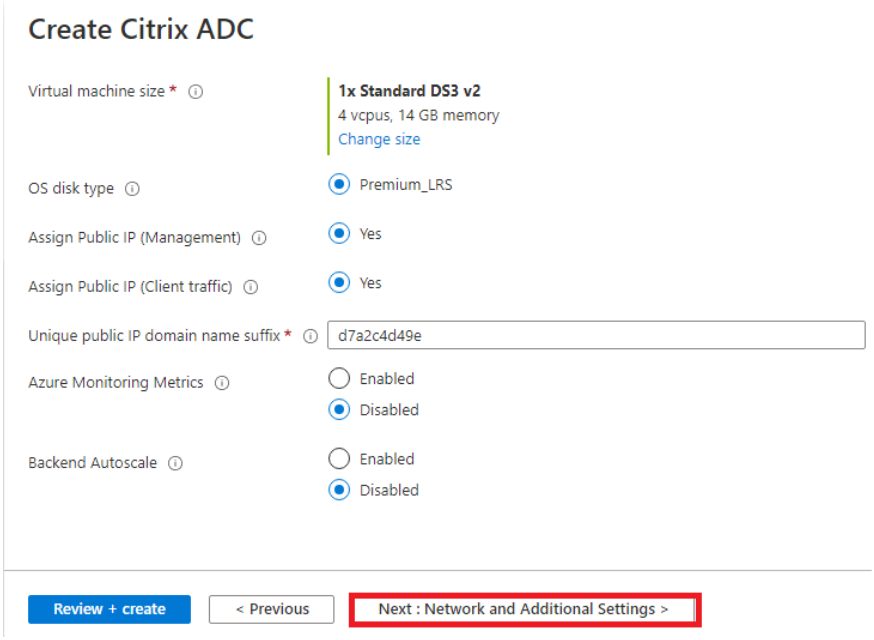
4. The **Basics** page appears. Create a Resource Group. Under the **Parameters** tab, enter details for the Region, Admin user name, Admin Password, license type (VM SKU), and other fields.



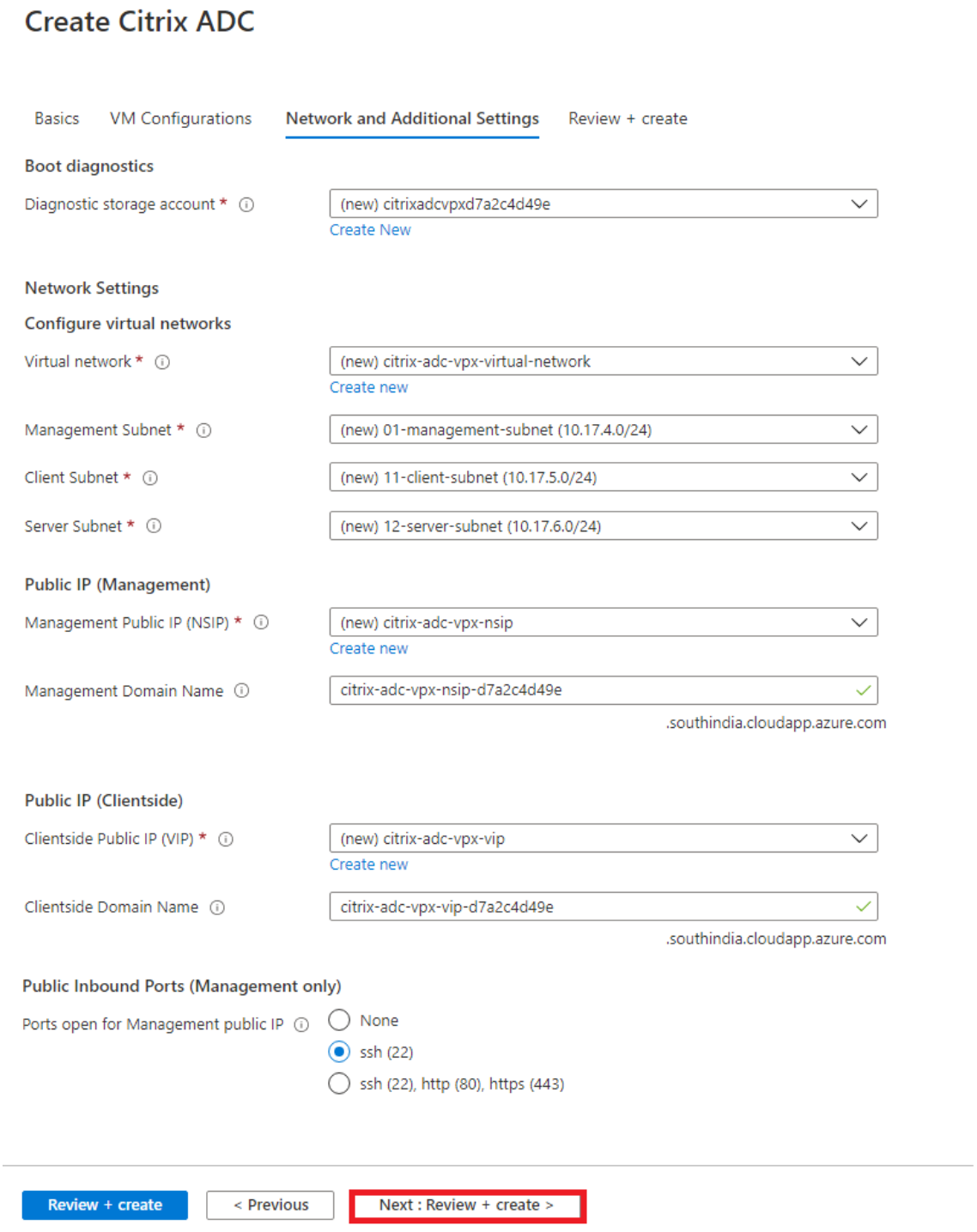
5. Click **Next : VM Configurations >**.



6. On the **VM Configurations** page, perform the following:
    - Configure public IP domain name suffix
    - Enable or disable **Azure Monitoring Metrics**
    - Enable or disable **Backend Autoscale**
7. Click **Next: Network and Additional settings >**



8. On **Network and Additional Settings** page, create Boot diagnostics account and configure the network settings.



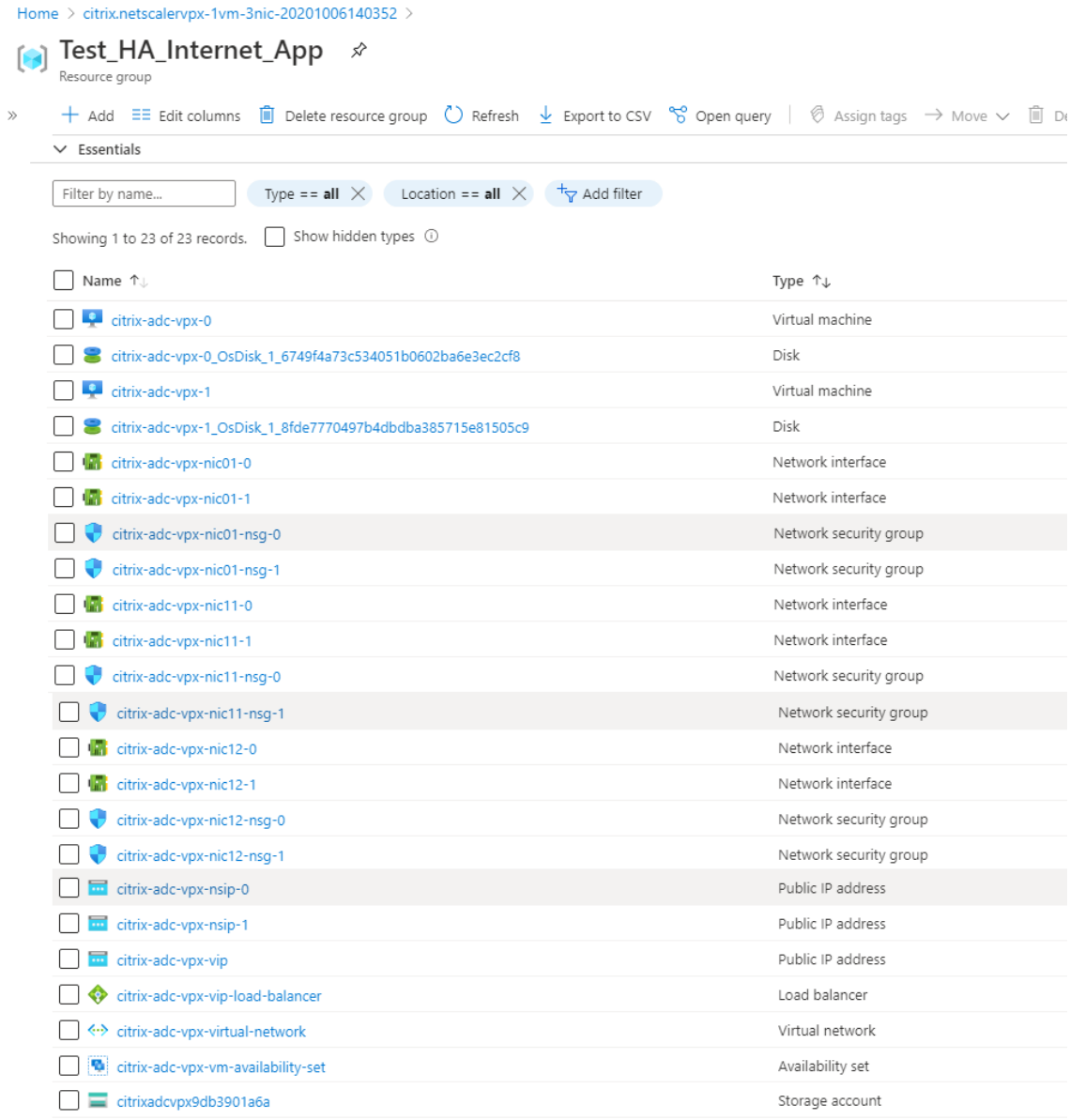
9. Click **Next: Review + create >**.
10. Review the basic settings, VM configuration, network and additional settings, and click **Create**.

    It might take a moment for the Azure Resource Group to be created with the required configura-

tions. After completion, select the Resource Group in the Azure portal to see the configuration details, such as LB rules, back-end pools, and health probes. The high availability pair appears as **citrix-adc-vpx-0** and **citrix-adc-vpx-1**.

If further modifications are required for your HA setup, such as creating more security rules and ports, you can do that from the Azure portal.

Once the required configuration is complete, the following resources are created.



11. You must log on to **citrix-adc-vpx-0** and **citrix-adc-vpx-1** nodes to validate the following configuration:

    - NSIP addresses for both nodes must be in the management subnet.
    - On the primary (citrix-adc-vpx-0) and secondary (citrix-adc-vpx-1) nodes, you must see

two SNIP addresses. One SNIP (client subnet) is used for responding to the ALB probes and the other SNIP (server subnet) is used for back-end server communication.

> Note
>
> In the HA-INC mode, the SNIP addresses of the citrix-adc-vpx-0 and citrix-adc-vpx-1 VMs are different, unlike with the classic on-premises ADC high availability deployment where both are the same.

On the primary node (citrix-adc-vpx-0)

```
> sh ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.18.0.4        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.4 (ns-vpx0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:34:21 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.5
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
```

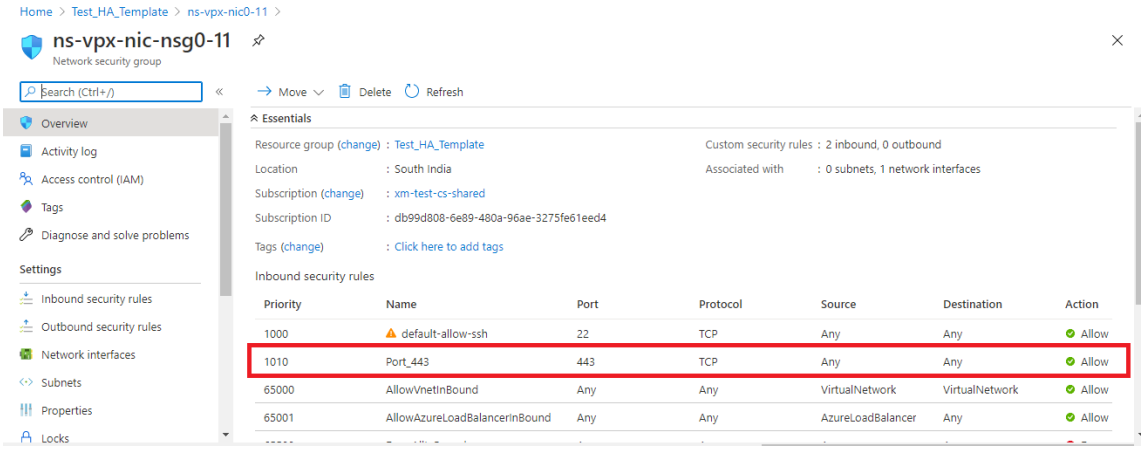
On the secondary node (citrix-adc-vpx-1)

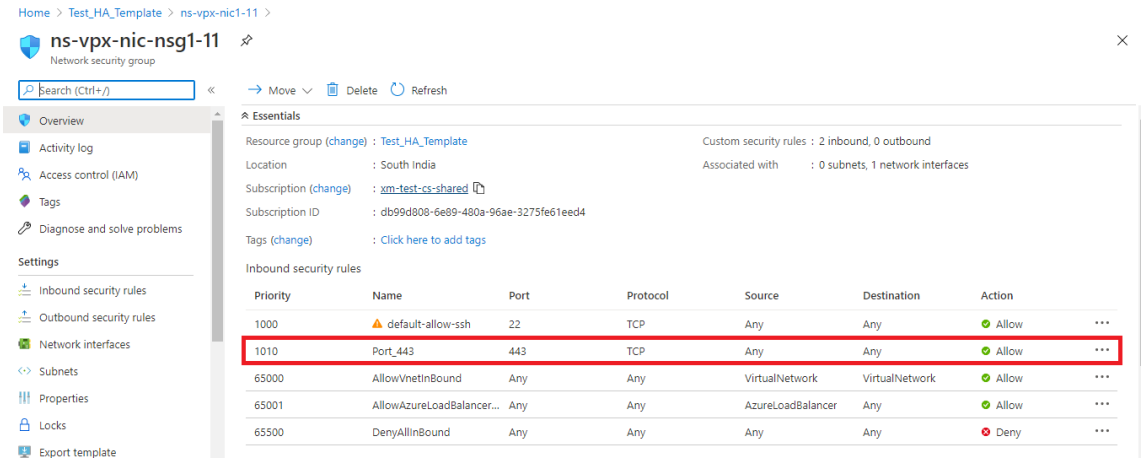

```
> show ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.18.0.5        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
>
```

```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.5 (ns-vpx1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:23:51 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.4
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

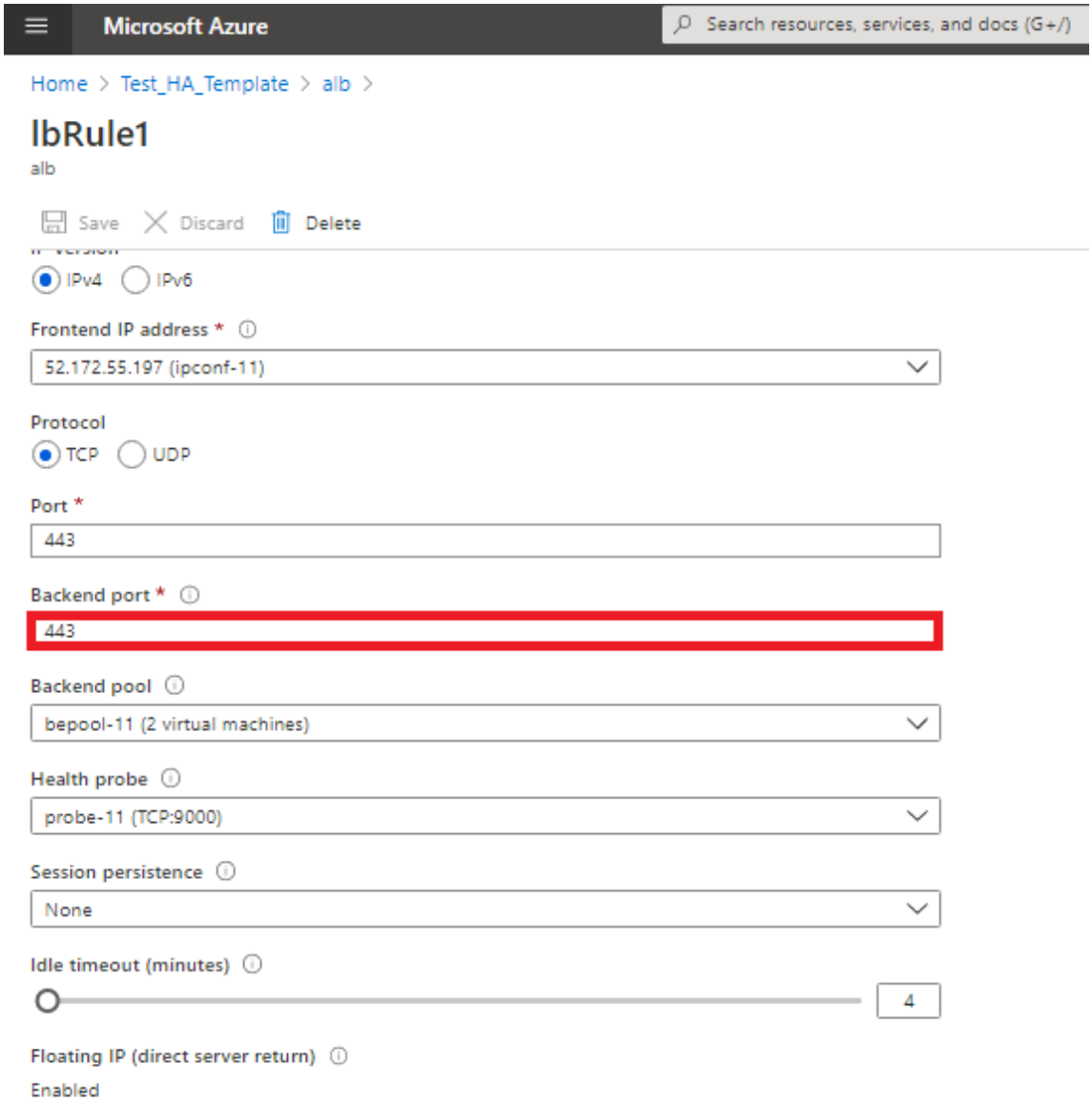
12. After the primary and secondary nodes are UP and the Synchronization status is **SUCCESS**, you must configure the load balancing virtual server or the gateway virtual server on the primary node (citrix-adc-vpx-0) with the public IP address of the ALB virtual server. For more information, see the Sample configuration section.
13. To find the public IP address of ALB virtual server, navigate to **Azure portal > Azure Load Balancer > Frontend IP configuration**.

14. Add the inbound security rule for virtual server port 443 on the network security group of both the client interfaces.

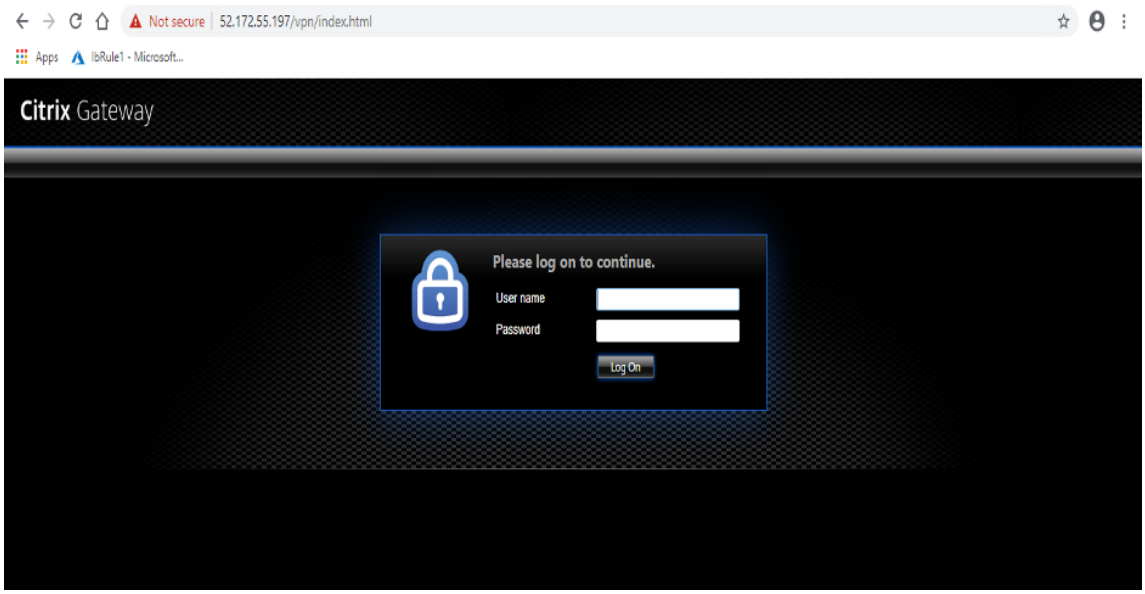




15. Configure the ALB port that you want to access, and create inbound security rule for the specified port. The Backend port is your load balancing virtual server port or the VPN virtual server port.



16. Now, you can access the load balancing virtual server or the VPN virtual server using FQDN associated with the ALB public IP address.



Sample configuration

To configure a gateway VPN virtual server and load balancing virtual server, run the following commands on the primary node (ADC-VPX-0). The configuration auto synchronizes to the secondary node (ADC-VPX-1).

Gateway sample configuration

```
enable feature aaa LB SSL SSLVPN
add ip 52.172.55.197 255.255.255.0 -type VIP
add vpn vserver vpn_ssl SSL 52.172.55.197 443
add ssl certKey ckp -cert cgwsanity.cer -key cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

Load balancing sample configuration

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 52.172.55.197 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

You can now access the load balancing or VPN virtual server using the fully qualified domain name (FQDN) associated with the internal IP address of ILB.

See the **Resources** section for more information about how to configure the load balancing virtual server.

Resources:

The following links provide additional information related to HA deployment and virtual server configuration:

- Create virtual servers
- Set up basic load balancing

Configure a high-availability setup with Azure external and internal load balancers simultaneously

September 14, 2021

The high availability pair on Azure supports both external and internal load balancers simultaneously.

You have the following two options to configure a high availability pair using both Azure external and internal load balancers:

- Using two LB virtual servers on the Citrix ADC appliance.
- Using one LB virtual server and an IP set. The single LB virtual server serves traffic to multiple IPs, which are defined by the IPset.

Perform the following steps to configure a high availability pair on Azure using both the external and internal load balancers simultaneously:

For Steps 1 and 2, use the Azure portal. For Steps 3 and 4, use the Citrix ADC VPX GUI or the CLI.

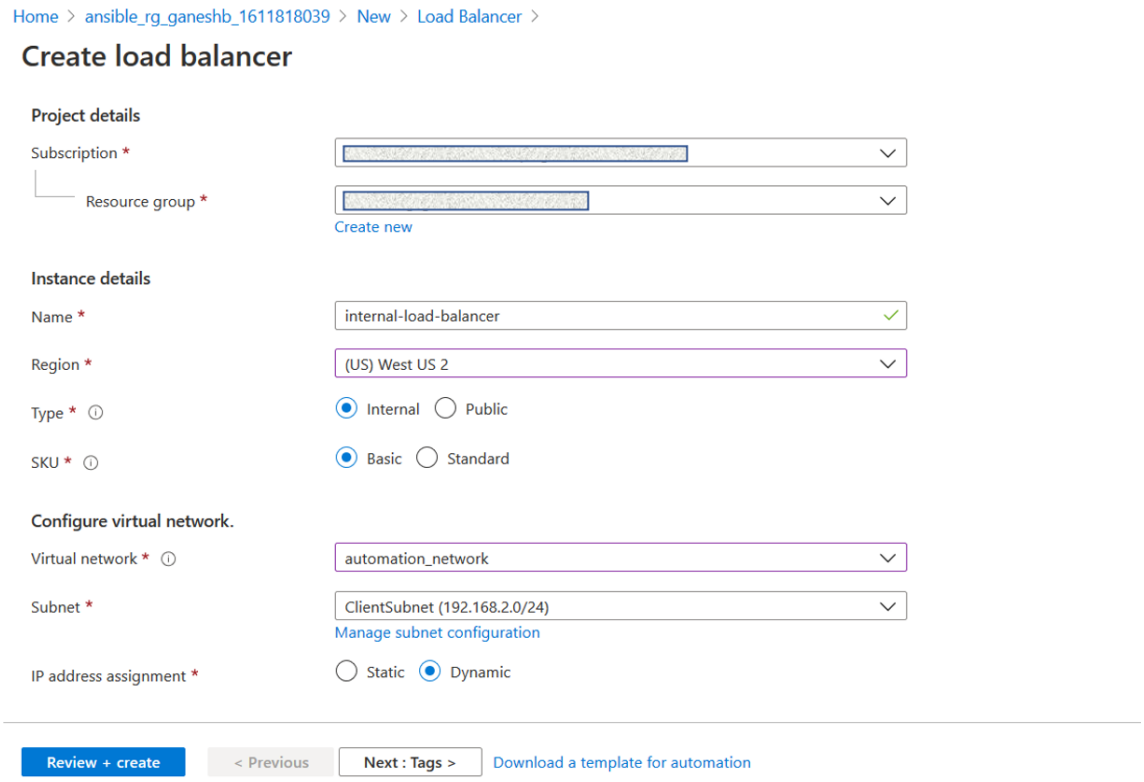
Step 1. Configure an Azure load balancer, either an external load balancer or an internal load balancer.

For more information on configuring high-availability setup with Azure external load balancers, see Configure a high-availability setup with multiple IP addresses and NIC.

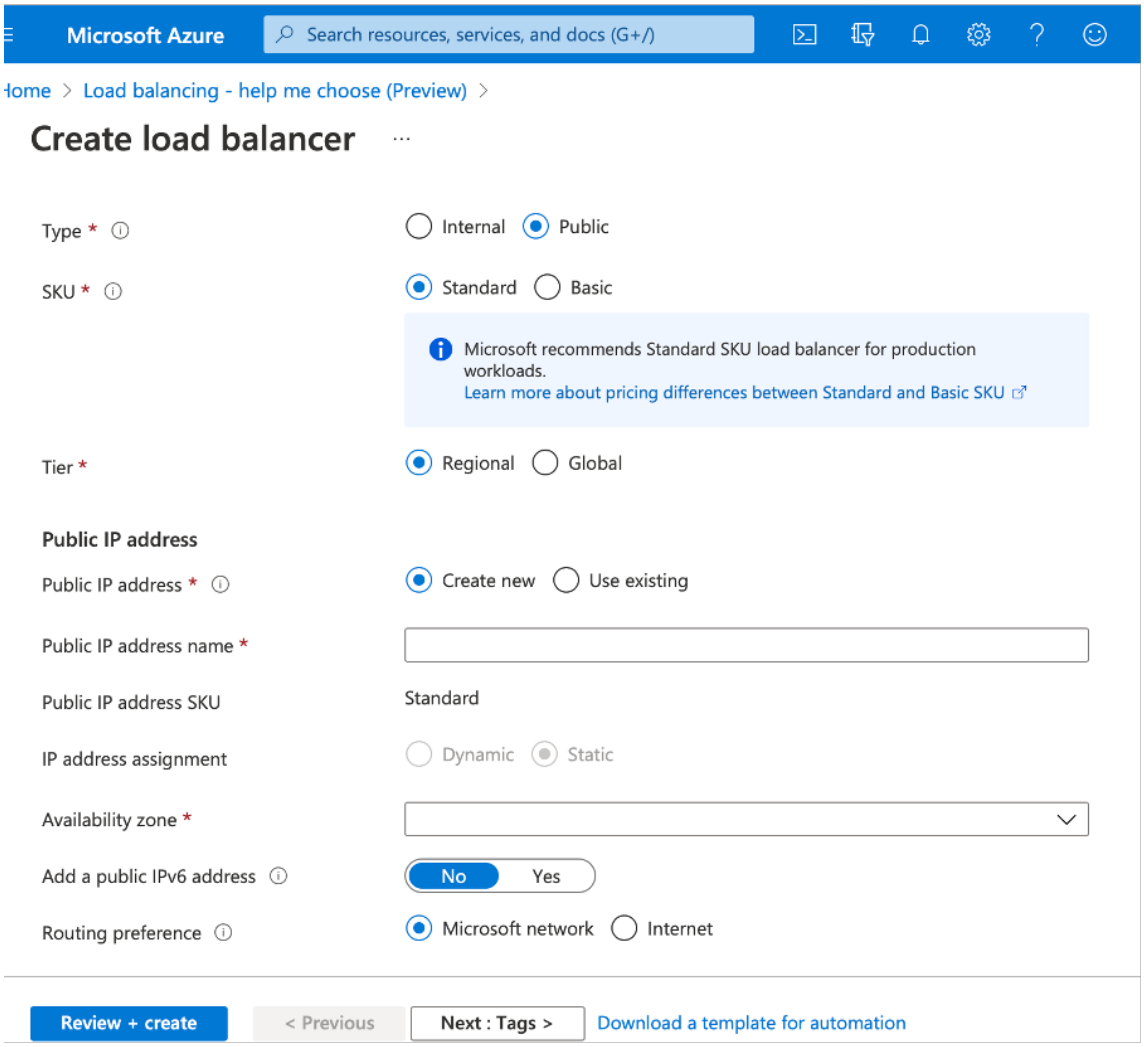
For more information on configuring high-availability setup with Azure internal load balancers, see Configure HA-INC nodes by using the Citrix high availability template with Azure ILB.

Step 2. Create an extra load balancer (ILB) in your resource group. In Step 1, if you have created an external load balancer, you now create an internal load balancer and conversely.

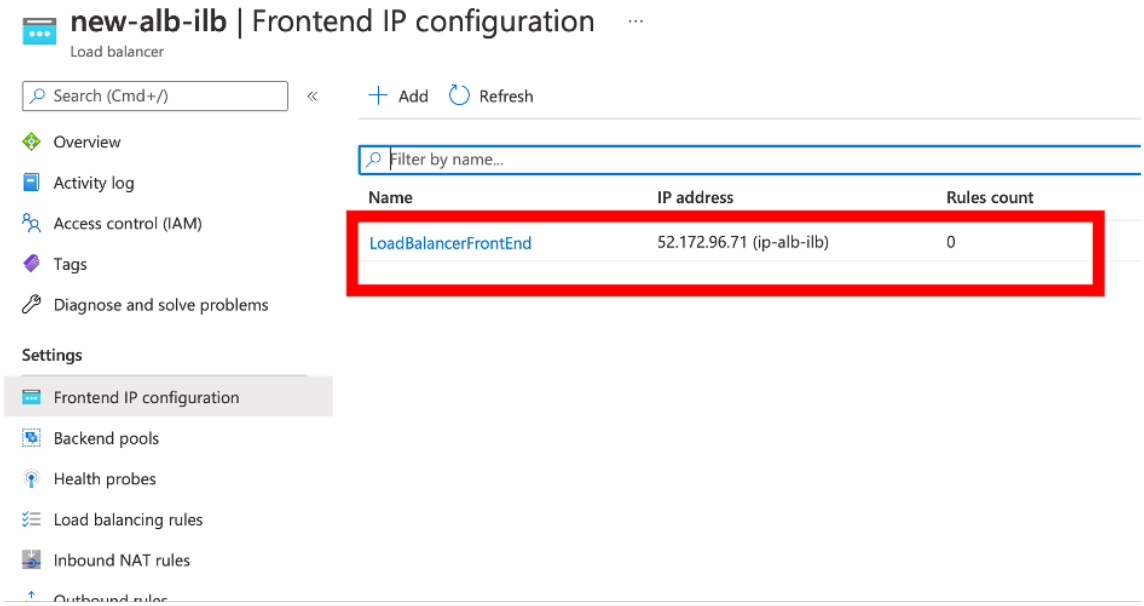
- To create an internal load balancer, choose the load balancer type as **Internal**. For the **Subnet** field, you must choose your Citrix ADC client subnet. You can choose to provide a static IP address in that subnet, provided there are no conflicts. Otherwise, choose the dynamic IP address.



- To create an external load balancer, choose the load balancer type as **Public** and create the public IP address here.



1. After you have created the Azure Load Balancer, navigate to **Frontend IP configuration** and note down the IP address shown here. You must use this IP address while creating the ADC load balancing virtual server as in Step 3.



2. In the **Azure Load Balancer configuration** page, the ARM template deployment helps create the LB rule, back-end pools, and health probes.
3. Add the high availability pair client NICs to the backend pool for the ILB.
4. Create a health probe (TCP, 9000 port)
5. Create two load balancing rules:
    - One LB rule for HTTP traffic (webapp use case) on port 80. The rule must also use the backend port 80. Select the created backend pool and the health probe. Floating IP must be enabled.
    - Another LB rule for HTTPS or CVAD traffic on port 443. The process is the same as the HTTP traffic.

Step 3. On the primary node of Citrix ADC appliance, create a load balancing virtual server for ILB.

1. Add a load balancing virtual server.

```
add lb vserver <name> <serviceType> [<ILB Frontend IP address>] [<
    port>]
<!--NeedCopy-->
```

Example:

```
add lb vserver vserver_name HTTP 52.172.96.71 80
<!--NeedCopy-->
```


> Note:
>
> Use the load balancer frontend IP address, which is associated with the additional Load balancer that you create in Step 2.

2. Bind a service to a load balancing virtual server.

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

For more information, see Set up basic load balancing

Step 4: As an alternative to Step 3, you can create a load balancing virtual server for ILB using IPsets.

1. Add an IP address of type virtual server IP (VIP).

```
add nsip <ILB Frontend IP address> -type <type>
<!--NeedCopy-->
```

Example:

```
add nsip 52.172.96.71 -type vip
<!--NeedCopy-->
```

2. Add an IPset on both primary and secondary nodes.

```
add ipset <name>
<!--NeedCopy-->
```

Example:

```
add ipset ipset1
<!--NeedCopy-->
```

3. Bind IP addresses to the IP set.

```
bind ipset <name> <ILB Frontend IP address>
<!--NeedCopy-->
```

Example:

```
bind ipset ipset1 52.172.96.71
<!--NeedCopy-->
```


4. Set the existing LB virtual server to use the IPset.

```
set lb vserver <vserver name> -ipset <ipset name>
<!--NeedCopy-->
```

Example:

```
set lb vserver vserver_name -ipset ipset1
<!--NeedCopy-->
```

For more information, see Configure a multi-IP virtual server.

Install a Citrix ADC VPX instance on Azure VMware Solution

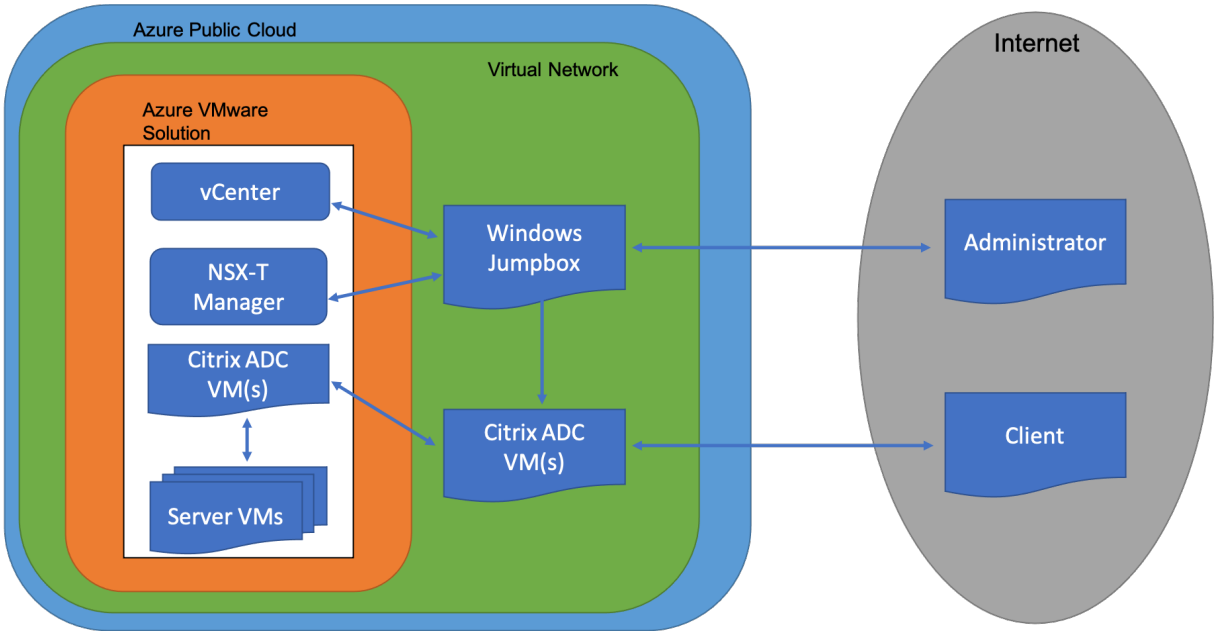
September 14, 2021

Azure VMware Solution (AVS) provides you with private clouds that contain vSphere clusters, built from dedicated bare-metal Azure infrastructure. The minimum initial deployment is three hosts, but additional hosts can be added one at a time, up to a maximum of 16 hosts per cluster. All provisioned private clouds have vCenter Server, vSAN, vSphere, and NSX-T.

The VMware Cloud (VMC) on Azure enables you to create cloud software-defined data centers (SDDC) on Azure with the number of ESX hosts that you want. The VMC on Azure supports Citrix ADC VPX deployments. VMC provides a user interface same as on-prem vCenter. It functions similar to the ESX-based Citrix ADC VPX deployments.

The following diagram shows the Azure VMware solution on the Azure public cloud that an administrator or a client can access over the internet. An administrator can create, manage, and configure workload or server VMs using Azure VMware solution. The admin can access the AVS's web-based vCenter and NSX-T Manager from a Windows Jumpbox. You can create the Citrix ADC VPX instances (standalone or high availability pair) and server VMs within Azure VMware Solution using vCenter, and manage the corresponding networking using NSX-T manager. The Citrix ADC VPX instance on AVS works similar to the on-prem VMware cluster of hosts. AVS is managed from a Windows Jumpbox that is created in the same virtual network.

A client can only access the AVS service by connecting to the VIP of ADC. Another Citrix ADC VPX instance outside Azure VMware Solution but in the same Azure virtual network helps add the VIP of the Citrix ADC VPX instance within Azure VMware Solution as a service. As per requirement, you can configure the Citrix ADC VPX instance to provide service over the internet.



Prerequisites

Before you begin installing a virtual appliance, do the following:

- For more information on Azure VMware solution and its prerequisites, see Azure VMware Solution documentation.
- For more information on deploying Azure VMware solution, see Deploy an Azure VMware Solution private cloud.
- For more information on creating a Windows Jump box VM to access and manage Azure VMware Solution, see Access an Azure VMware Solution private cloud
- In Windows Jump box VM, download the Citrix ADC VPX appliance setup files.
- Create appropriate NSX-T network segments on VMware SDDC to which the virtual machines connect. For more information, see Add a network segment in Azure VMware Solution
- Obtain VPX license files.
- Virtual machines (VMs) created or migrated to the Azure VMware Solution private cloud must be attached to a network segment.

VMware cloud hardware requirements

The following table lists the virtual computing resources that the VMware SDDC must provide for each VPX nCore virtual appliance.

Table 1. Minimum virtual computing resources required for running a Citrix ADC VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (vCPU)	2
Virtual network interfaces	In VMware SDDC, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded to version 7 or higher.
Disk space	20 GB

> Note
>
> This is in addition to any disk requirements for the hypervisor.

For production use of the VPX virtual appliance, the full memory allocation must be reserved.

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. The following table describes the system requirements for installing OVF tool.

Table 2. System requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the “OVF Tool User Guide” PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
NIC	100 Mbps or faster NIC

For information about installing OVF, search for the “OVF Tool User Guide” PDF file at http://kb.vmware.com/.

Downloading the Citrix ADC VPX setup files

The Citrix ADC VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log

on. If you do not have a Citrix account, access the home page at http://www.citrix.com. Click the **New Users link**, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

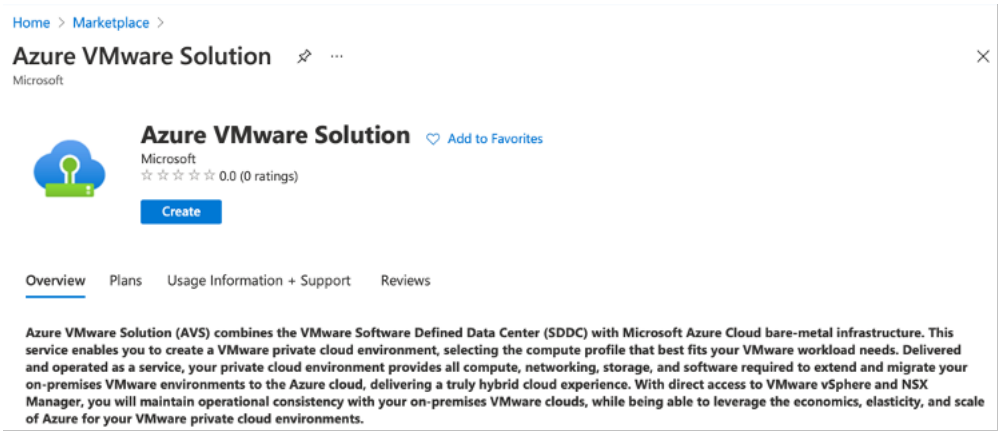
Citrix.com > **Downloads > Citrix ADC > Virtual Appliances**.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

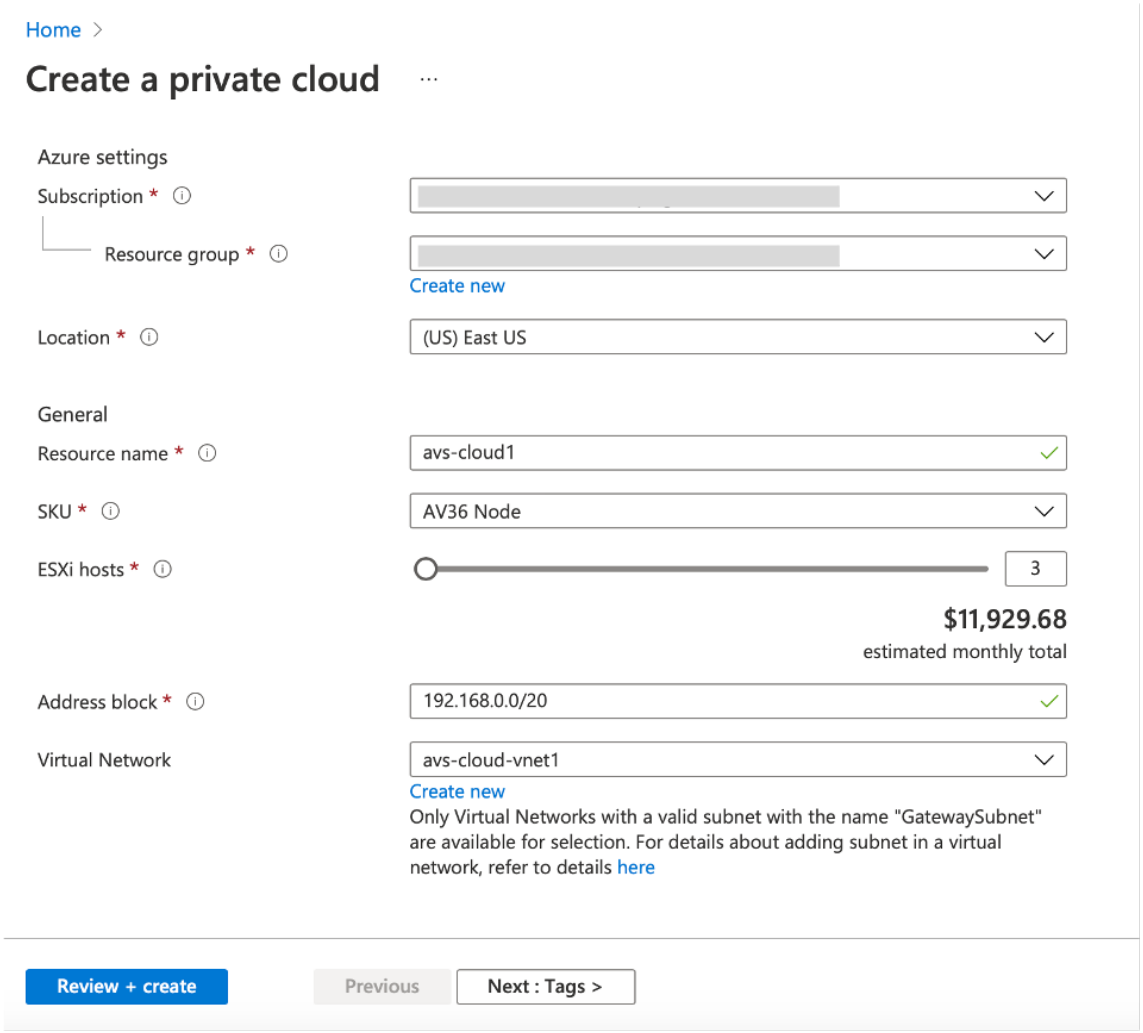
- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-13.0-79.64-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-13.0-79.64.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-13.0-79.64.mf)

Deploy Azure VMware solution

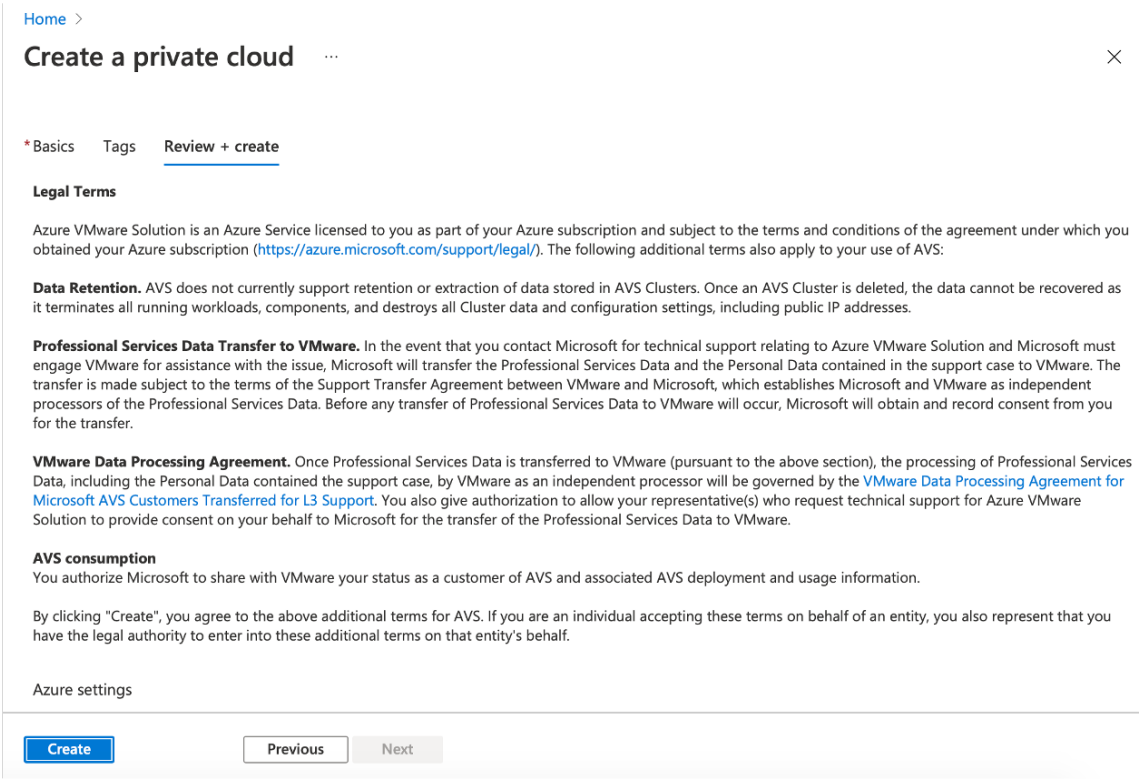
1. Log in to your Microsoft Azure portal, and navigate to **Azure Marketplace**.
2. From the **Azure Marketplace**, search **Azure VMware Solution** and click **Create**.



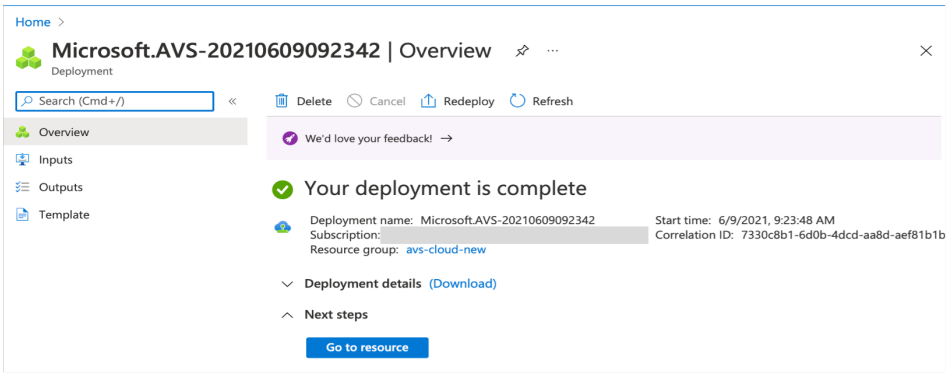
3. In the **Create a private cloud** page, enter the following details:
    - Select a minimum of 3 ESXi hosts to create the default cluster of your private cloud.
    - For the **Address block** field, use **/22** address space.
    - For the **Virtual Network**, make sure that the CIDR range doesn't overlap with any of your on-premises or other Azure subnets (virtual networks) or with the gateway subnet.
    - Gateway subnet is used to express route the connection with private cloud.



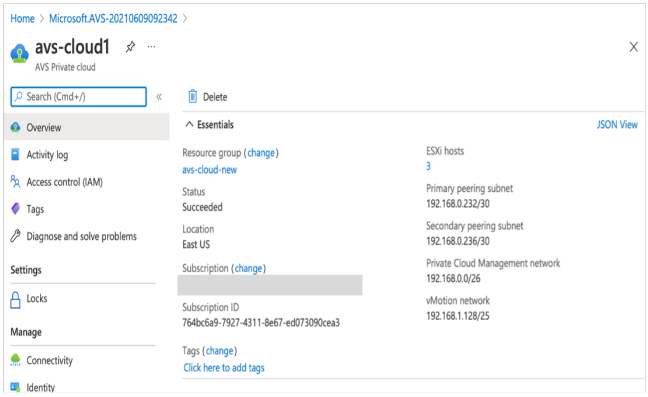
4. Click **Review + Create**.
5. Review the settings. If you must change any settings, click **Previous**.



6. Click **Create**. Private cloud provisioning process starts. It can take up to two hours for the private cloud to be provisioned.



7. Click **Go to resource**, to verify the private cloud that is created.



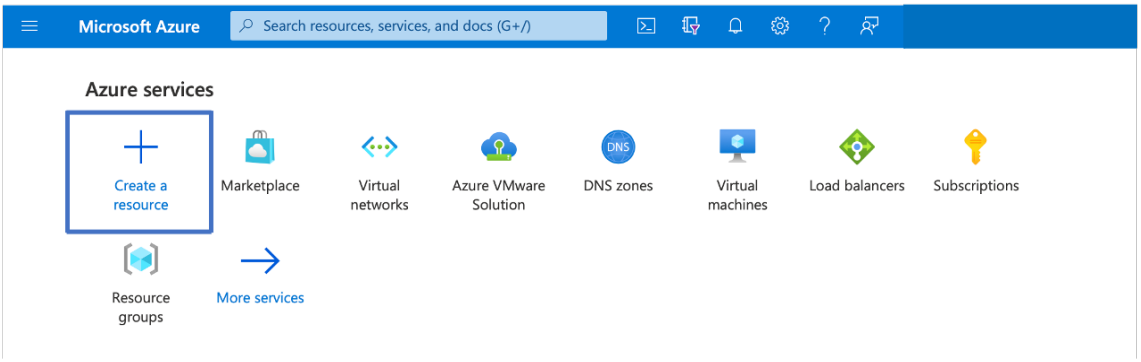
> Note
>
> To access this resource, you need a VM in Windows that acts as a Jump box.

Connect to an Azure virtual machine running Windows

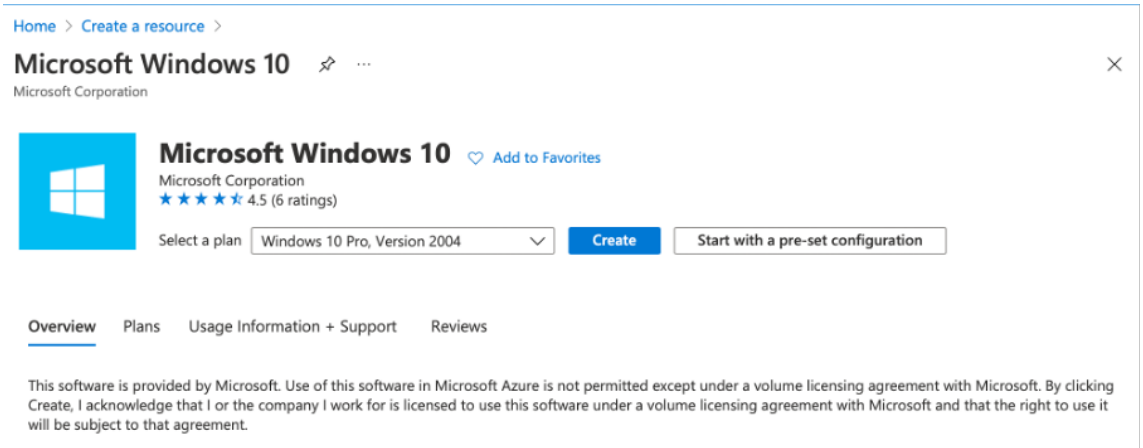
This procedure shows you how to use the Azure portal to deploy a virtual machine (VM) in Azure that runs Windows Server 2019. To see your VM in action, you then RDP to the VM and install the IIS web server.

To access the private cloud that you have created, you need to create a Windows Jump box within the same virtual network.

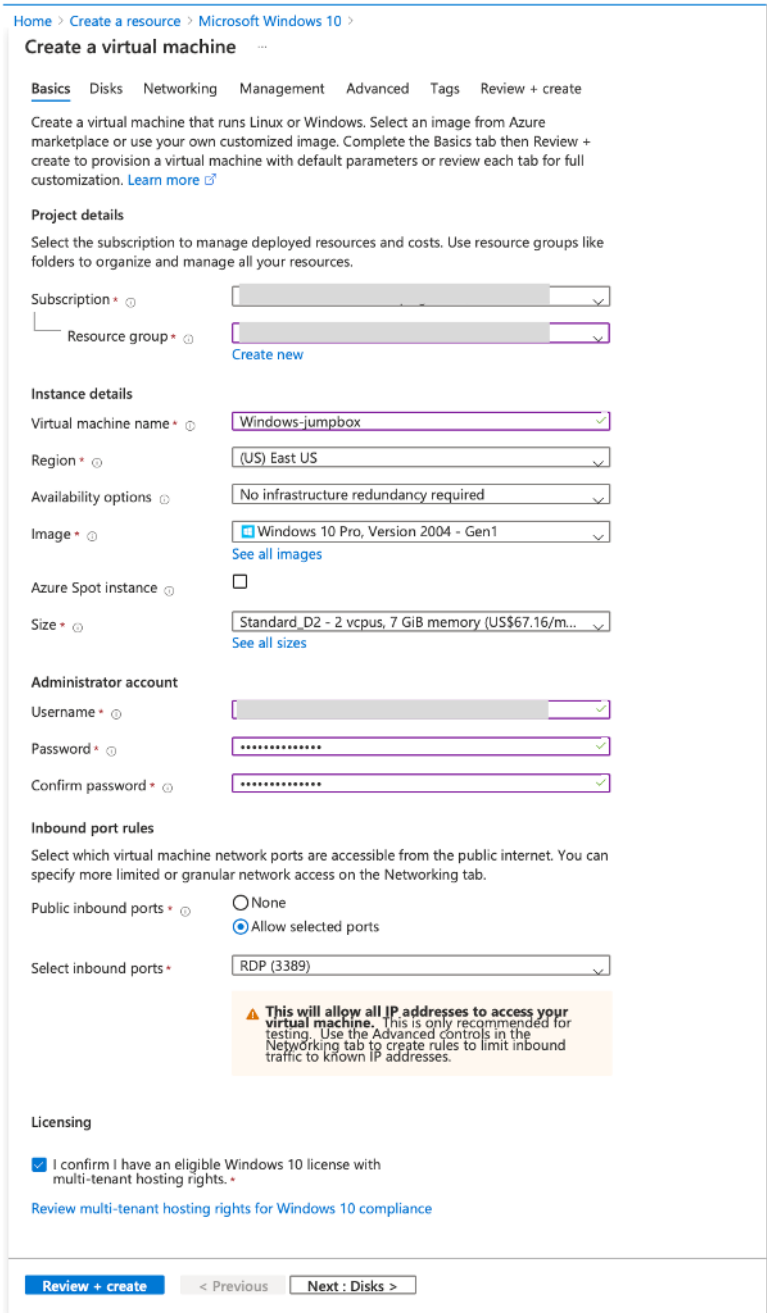
1. Go to the **Azure portal**, and click **Create a Resource**.



2. Search for **Microsoft Windows 10**, and click **Create**.



3. Create a virtual machine (VM) that runs Windows Server 2019. The **Create a virtual machine** page appears. Enter all the details in **Basics** tab, and select the **Licensing** check box. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.



4. After validation runs, select the **Create** button at the bottom of the page.
5. After the deployment is complete, select **Go to resource**.
6. Go to the Windows VM that you have created. Use the public IP address of the Windows VM and connect using RDP.

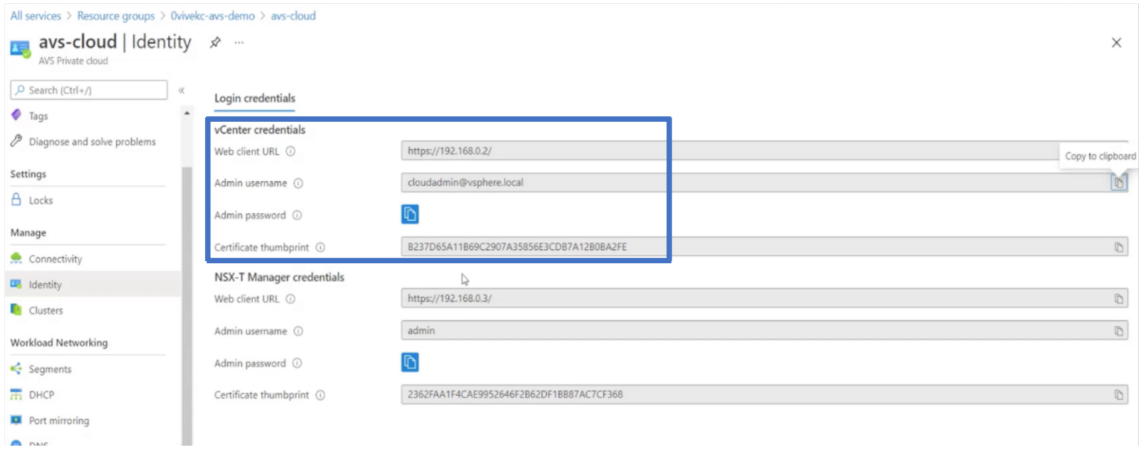
   Use the **Connect** button in the Azure portal to start a Remote Desktop (RDP) session from a Windows desktop. First you connect to the virtual machine, and then you sign on.

   To connect to a Windows VM from a Mac, you must install an RDP client for Mac such as Microsoft

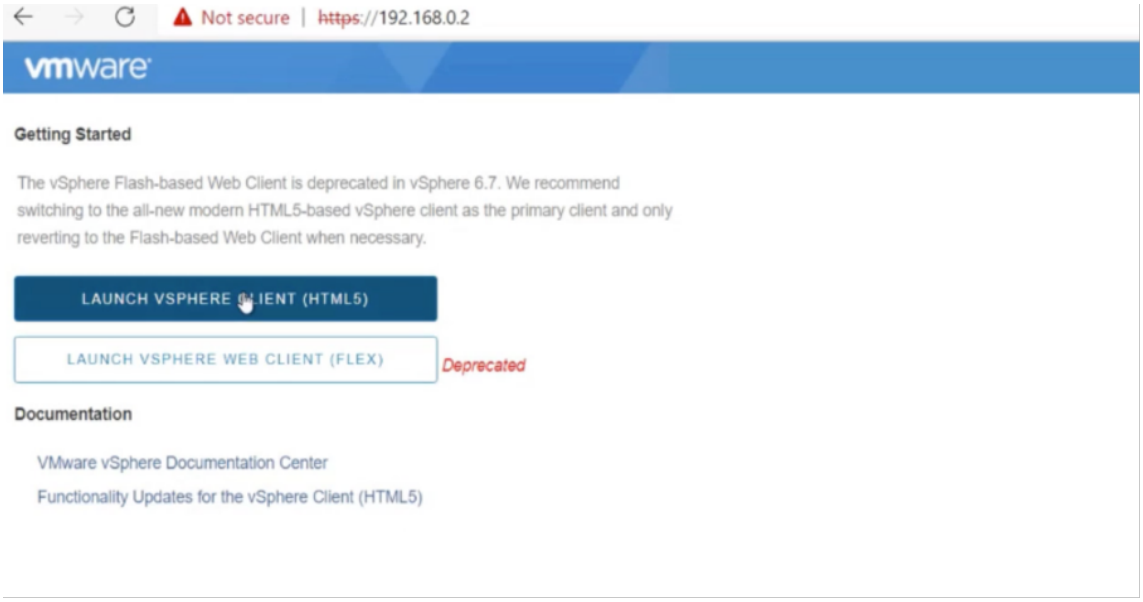
Remote Desktop. For more information, see How to connect and sign on to an Azure virtual machine running Windows.

Access your Private Cloud vCenter portal

1. In your Azure VMware Solution private cloud, under **Manage**, select **Identity**. Make note of the vCenter credentials.

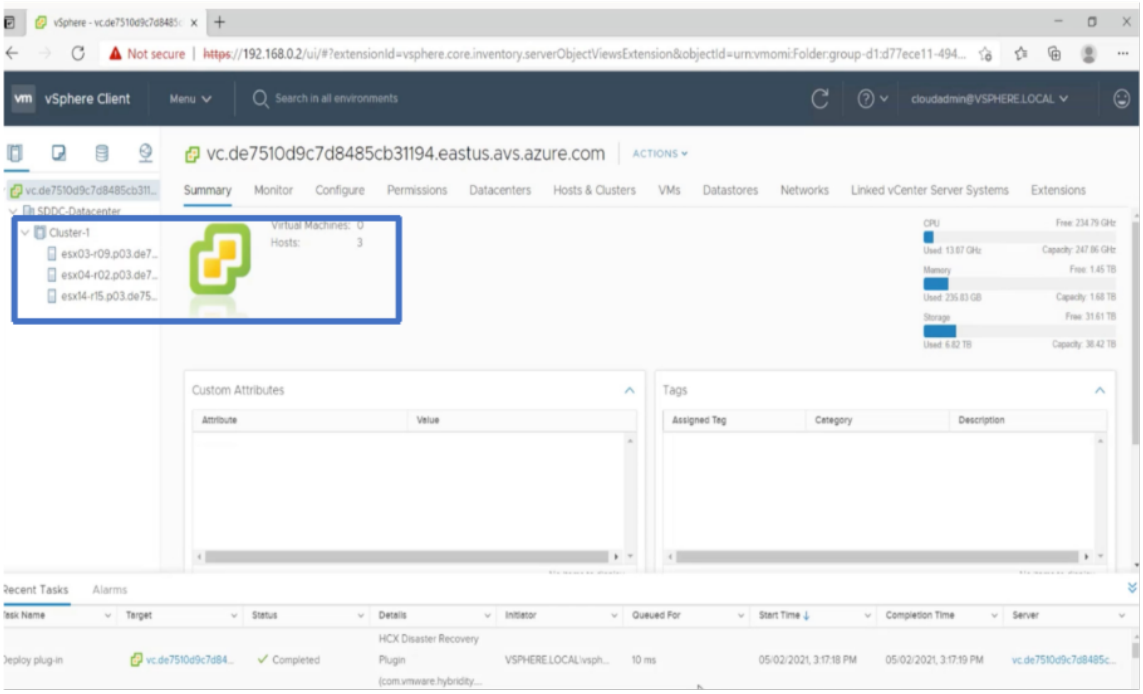


2. Launch the vSphere client by typing the vCenter web client URL.



3. Log in to VMware vSphere using the vCenter credentials of your Azure VMware Solution private cloud.

4. In the vSphere client, you can verify the ESXi hosts that you created in Azure portal.



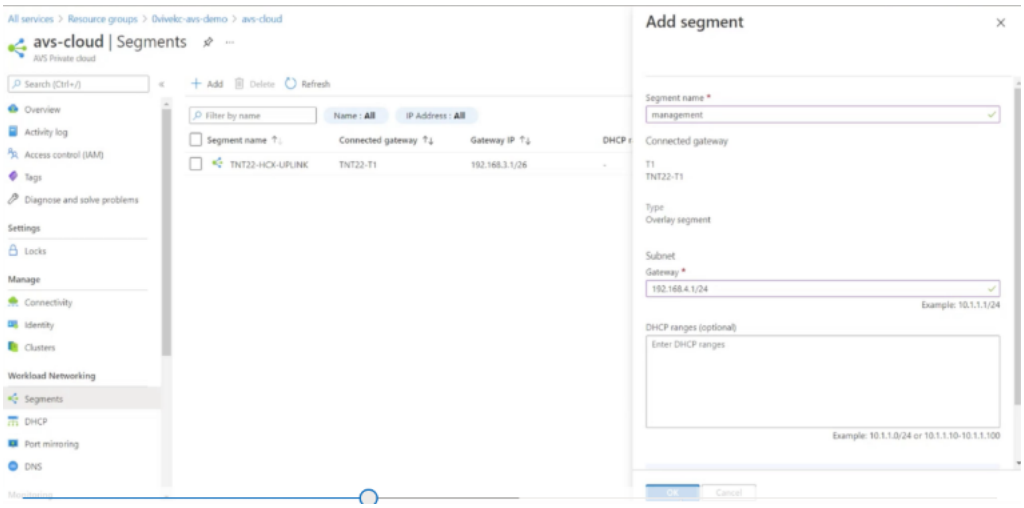
For more information, see Access your Private Cloud vCenter portal.

Create an NSX-T segment in the Azure portal

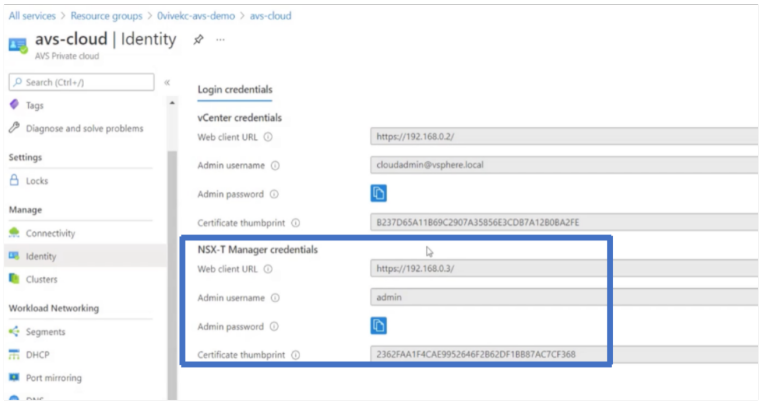
You can create and configure an NSX-T segment from the Azure VMware Solution console in the Azure portal. These segments are connected to the default Tier-1 gateway, and the workloads on these segments get East-West and North-South connectivity. Once you create the segment, it displays in NSX-T

Manager and vCenter.

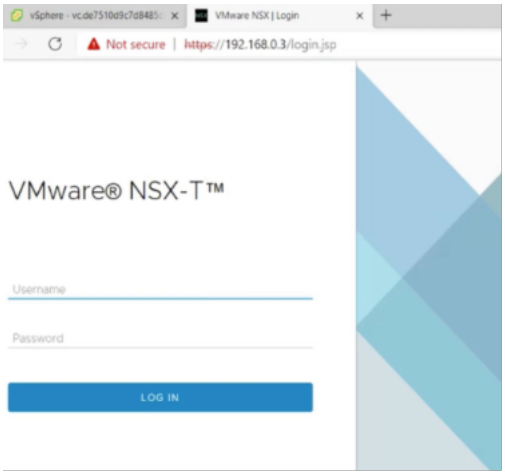
1. In your Azure VMware Solution private cloud, under **Workload Networking**, select **Segments > Add**. Provide the details for the new logical segment and select **OK**. You can create three separate segments for Client, Management, and Server interfaces.



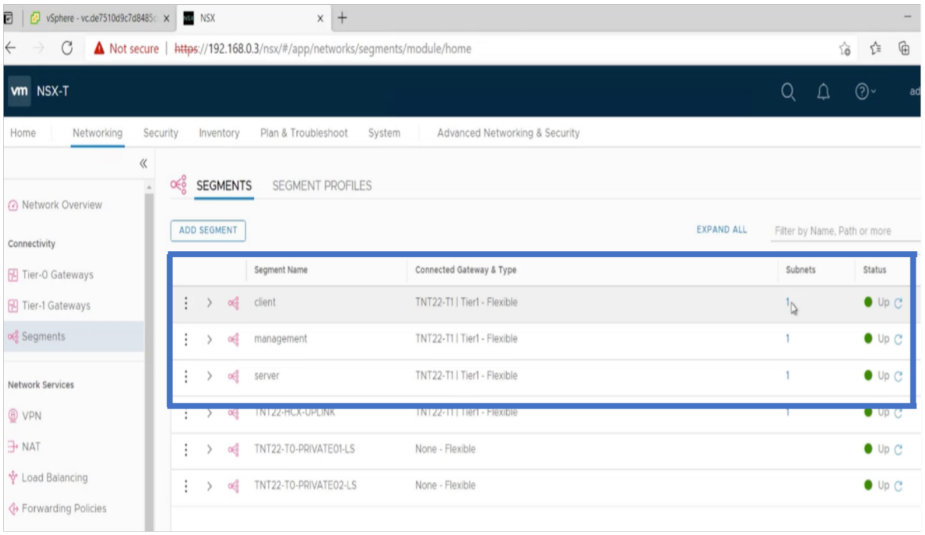
2. In your Azure VMware Solution private cloud, under **Manage**, select **Identity**. Make note of the NSX-T Manager credentials.



3. Launch the VMware NSX-T Manager by typing the NSX-T web client URL.



4. In the NSX-T manager, under **Networking > Segments**, you can see all the segments that you have created. You can also verify the subnets.



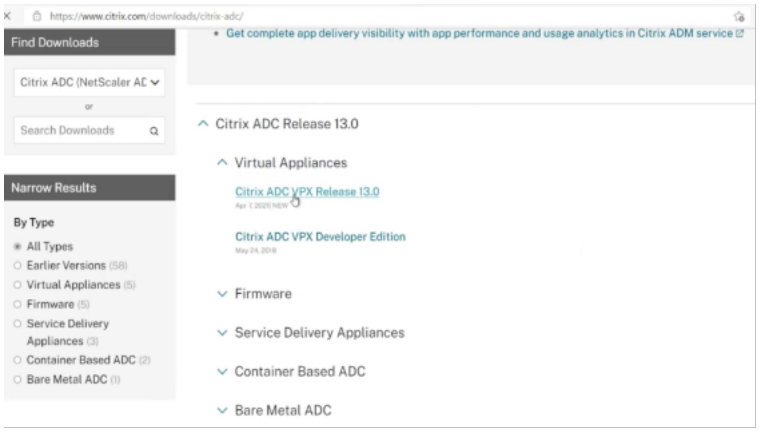
For more information, see Create an NSX-T segment in the Azure portal.

Install a Citrix ADC VPX instance on VMware cloud

After you have installed and configured VMware Software-Defined Data Center (SDDC), you can use the SDDC to install virtual appliances on the VMware cloud. The number of virtual appliances that you can install depends on the amount of memory available on the SDDC.

To install Citrix ADC VPX instances on VMware cloud, perform these steps in Windows Jumpbox VM:

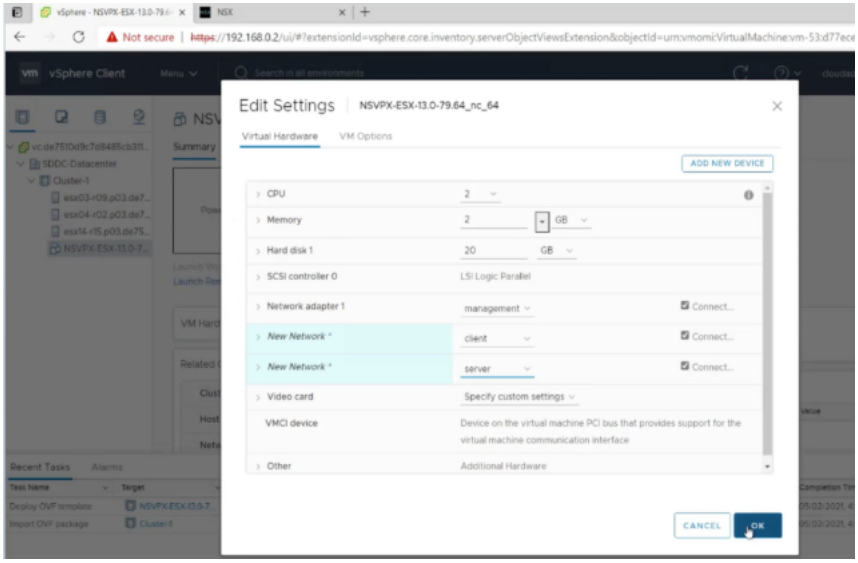
1. Download the Citrix ADC VPX instance setup files for ESXi host from the Citrix Downloads site.



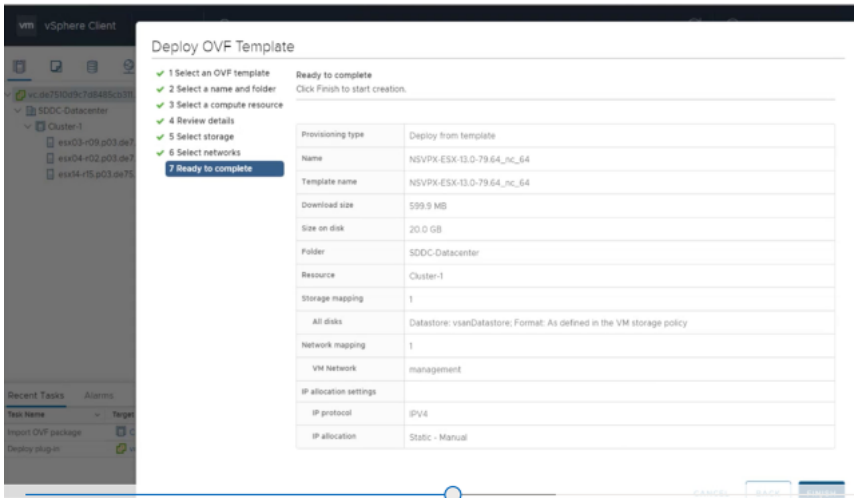
2. Open VMware SDDC in the Windows Jumpbox.
3. In the **User Name** and **Password** fields, type the administrator credentials, and then click **Login**.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file** field, browse to the location at which you saved the Citrix ADC VPX instance setup files, select the .ovf file, and click **Next**.

   > NOTE
   >
   > By default, the Citrix ADC VPX instance uses E1000 network interfaces. To deploy ADC with the VMXNET3 interface, modify the OVF to use VMXNET3 interface instead of E1000. Availability of VMXNET3 interface is limited by Azure infrastructure and might not be available in Azure VMware Solution.

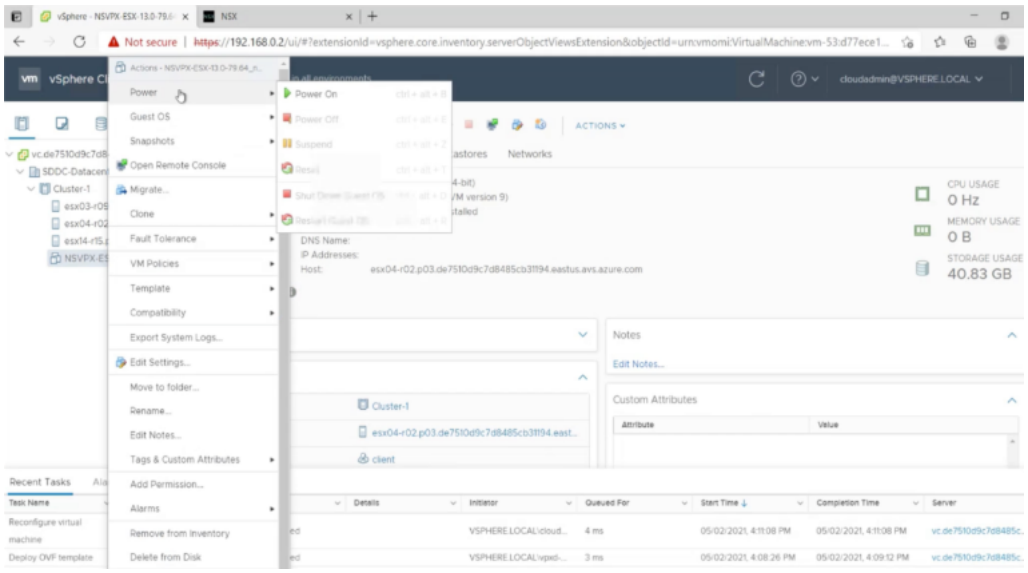
6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the VMware SDDC. Click **OK**.



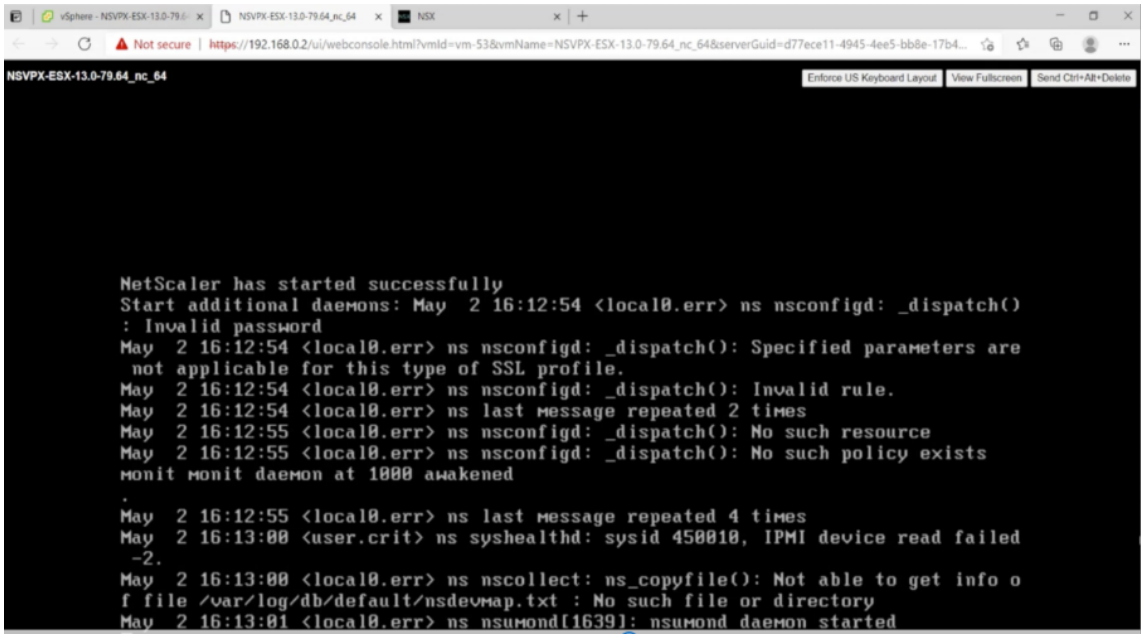
7. Click **Finish** to start installing a virtual appliance on VMware SDDC.



8. You are now ready to start the Citrix ADC VPX instance. In the navigation pane, select the Citrix ADC VPX instance that you have installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.



9. You are now connected to the Citrix ADC VM from the vSphere client.



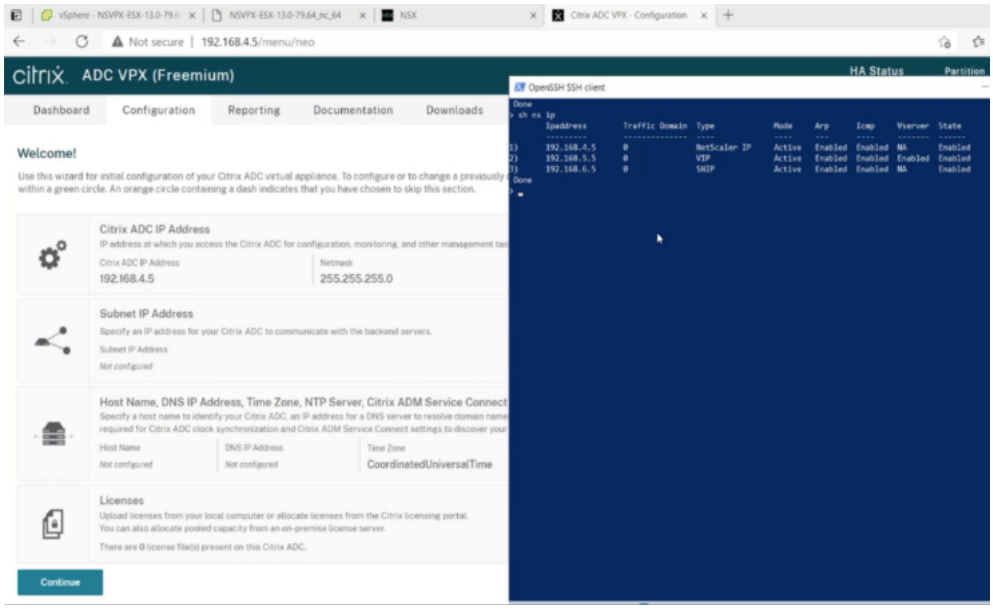
10. To access the Citrix ADC appliance by using the SSH keys, type the following command in the CLI:

```
ssh nsroot@<management IP address>
<!--NeedCopy-->
```

Example:

```
ssh nsroot@192.168.4.5
<!--NeedCopy-->
```

11. You can verify the ADC configuration by using the `show ns ip` command.



Add Azure Autoscale settings

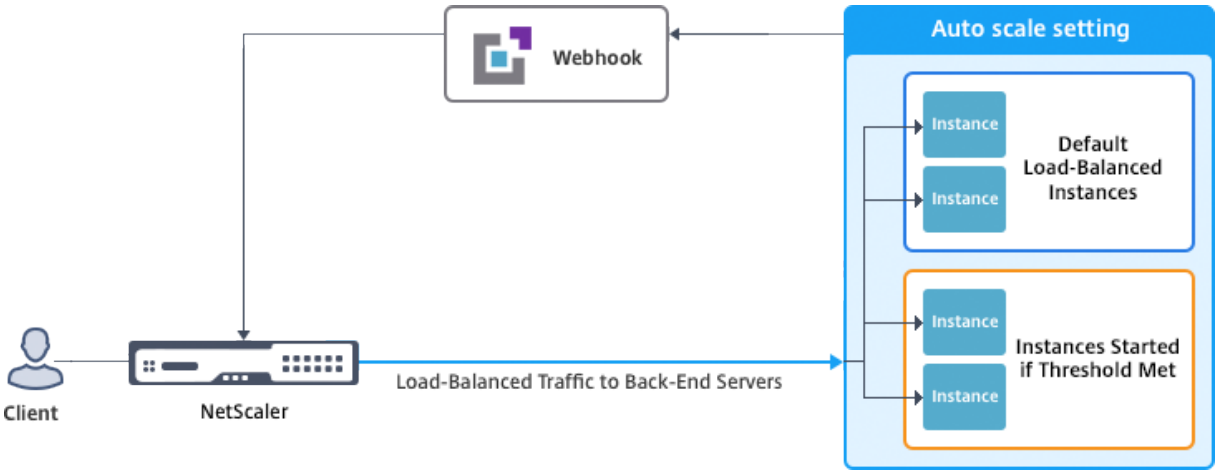
September 14, 2021

Efficient hosting of applications in a cloud involves easy and cost-effective management of resources depending on the application demand. To meet increasing demand, you have to scale network resources upward. Whether demand subsides, you must scale down to avoid the unnecessary cost of idle resources. To minimize the cost of running the application, you have to constantly monitor traffic, memory and CPU use, and so on. However, monitoring traffic manually is cumbersome. For the application environment to scale up or down dynamically, you must automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

You can use Autoscale with Azure virtual machine scale sets (VMSS) for VPX multi-IP standalone and high availability deployment on Azure.

Integrated with the Azure virtual machine scale sets (VMSS) and Autoscale feature, the Citrix ADC VPX instance provides the following advantages:

- Load balance and management: Auto configures servers to scale up and scale down, depending on demand. The VPX instance auto detects the VMSS Autoscale setting in the back-end subnet in the same resource group as the VPX instance and allows the user to select the VMSS Autoscale setting to balance the load. All of this is done by auto configuring Citrix ADC virtual and subnet IP addresses on the VPX instance.
- High availability: Detects Autoscale groups in the same resource group and load-balance servers.
- Better network availability: The VPX instance supports back-end servers on different virtual networks (VNets).



For more information, see the following Azure topic

- Virtual Machine Scale Sets Documentation
- Overview of Autoscale in Microsoft Azure Virtual Machines, Cloud Services, and Web Apps

Before you begin

1. Read Azure-related usage guidelines. For more information, see Deploy a Citrix ADC VPX instance on Microsoft Azure.
2. Create one or more Citrix ADC VPX instances with three network interfaces on Azure according to your requirement (standalone or high availability deployment).
3. Open the TCP 9001 port on the network security group of the 0/1 interface of the VPX instance. The VPX instance uses this port to receive the scale-out and scale-in notification.
4. Create an Azure virtual machine scale set (VMSS) in the same resource group. If you don't have an existing VMSS configuration, complete the following tasks:

   a) Create a VMSS

   b) Enable Autoscale on VMSS

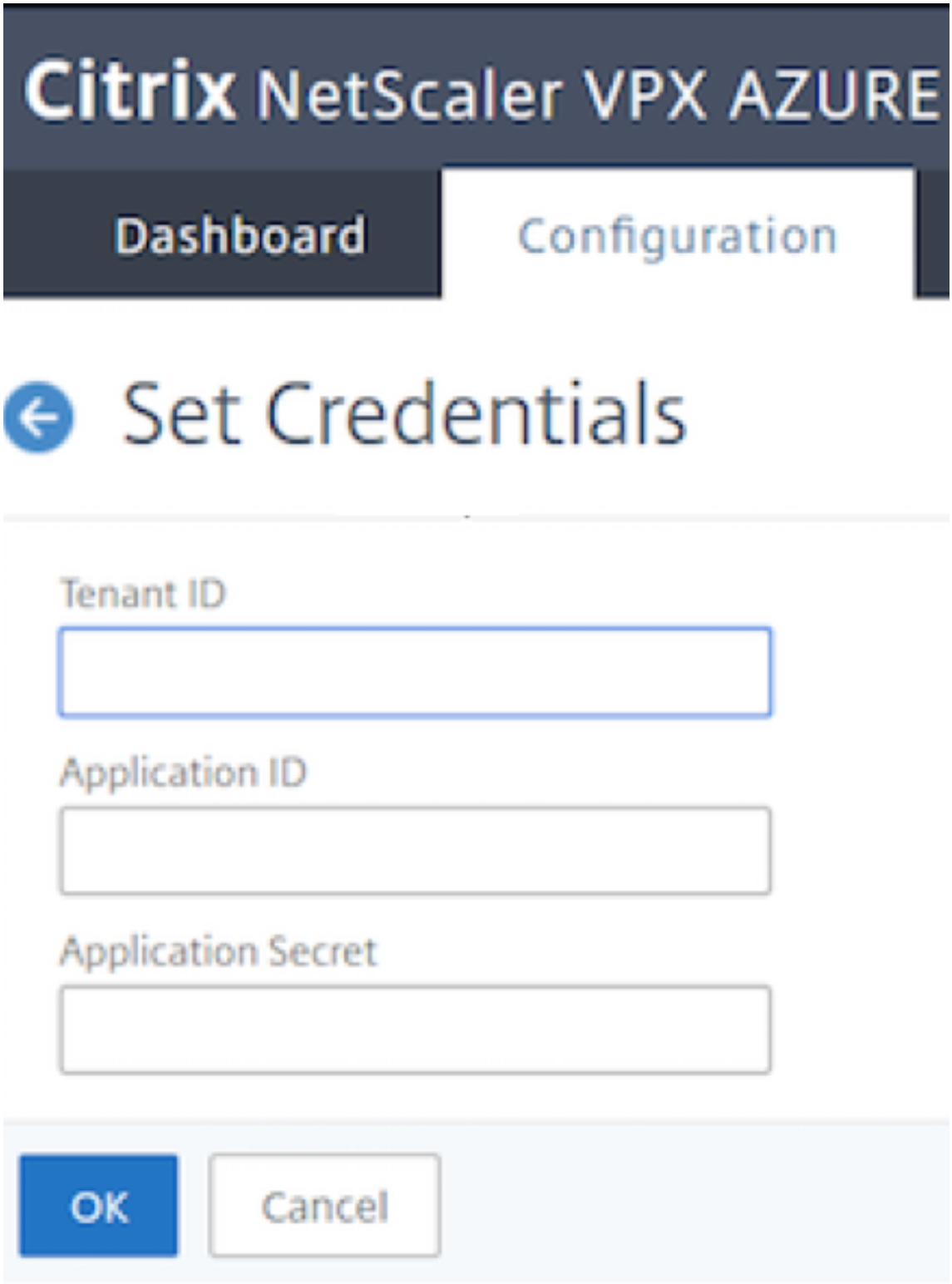
   c) Create scale-in and scale-out policy in VMSS Autoscale setting

   For more information, see Overview of Autoscale with Azure virtual machine scale sets.
5. Create an Azure Active Directory (ADD) application and service principal that can access resources. Assign contributor role to the newly created AAD application. For more information, see Use portal to create an Azure Active Directory application and service principal that can access resources.

Add VMSS to a Citrix ADC VPX instance

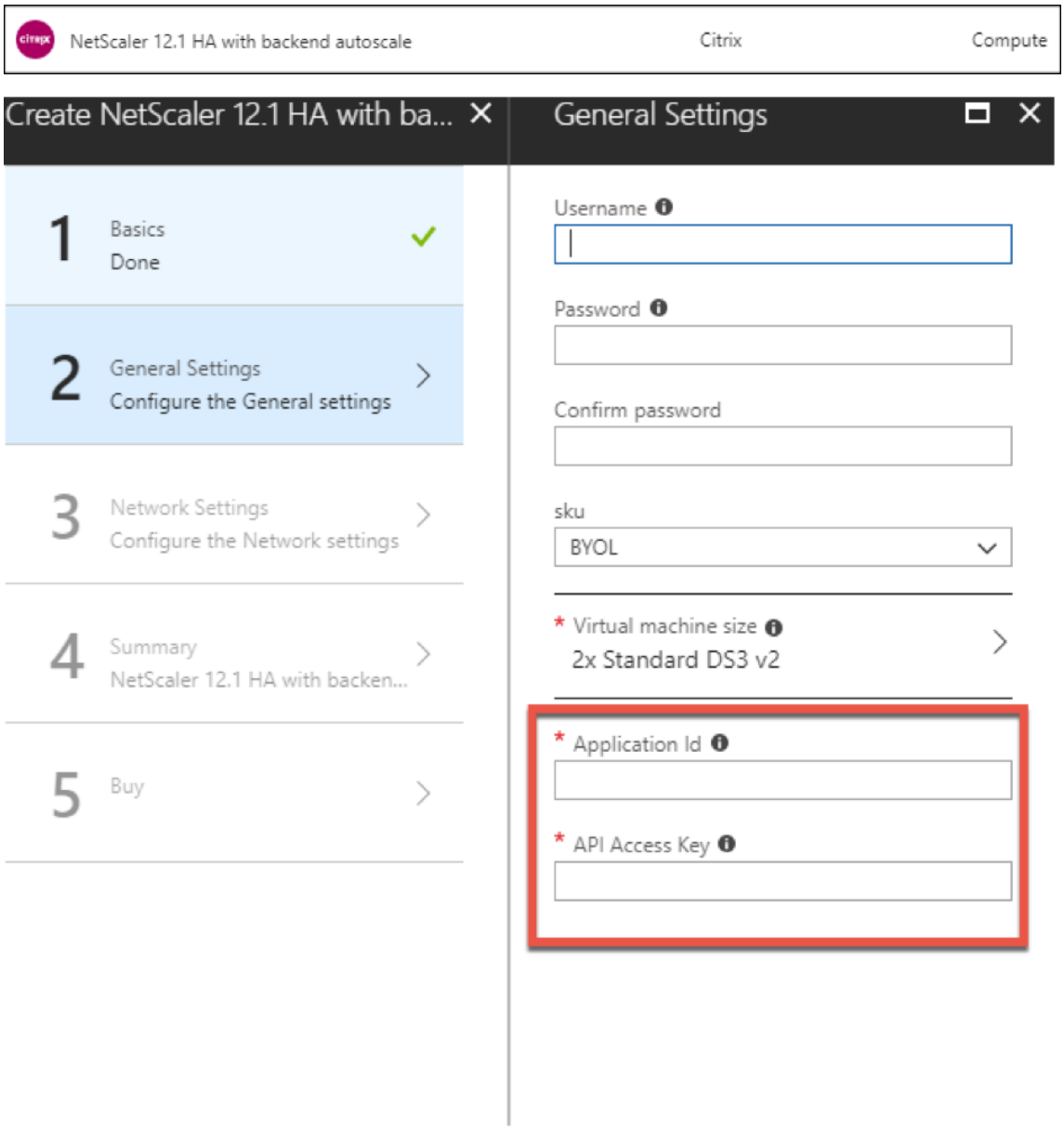
You can add the Autoscale setting to a VPX instance with a single click by using the GUI. Complete these steps to add the Autoscale setting to the VPX instance:

1. Log on to the VPX instance.
2. When you log on to the Citrix ADC VPX instance for the first time, you see the Set Credentials page. Add the required Azure credentials for the Autoscale feature to work.



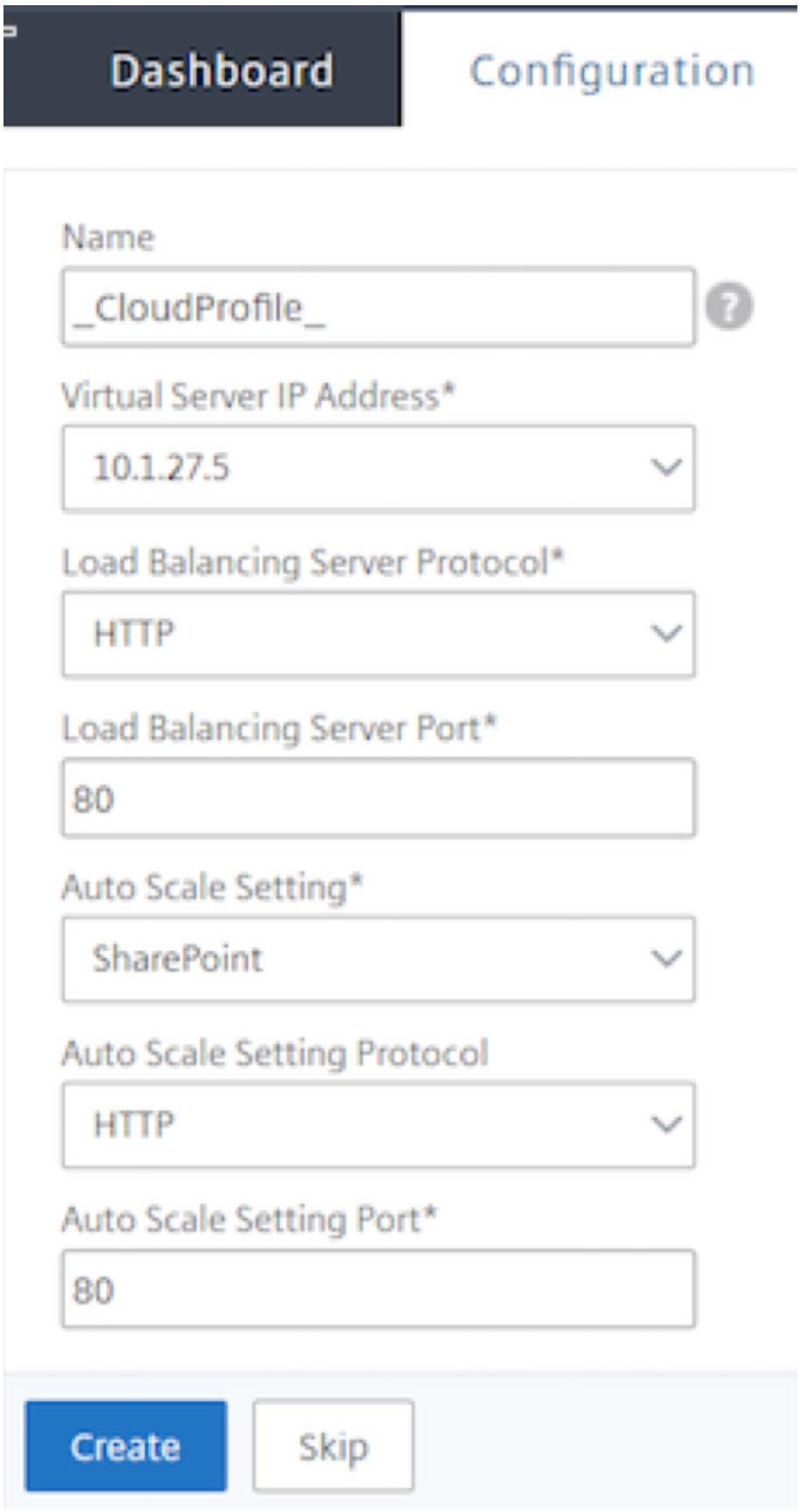
The Set Credential page appears only when the application ID and API access key are not set or the correct application ID and API access keys (same as application secret) is not set in the Azure portal.

When you deploy the “NetScaler 12.1 HA with back end Autoscale” offer from the Azure Marketplace, the Azure portal prompts for Azure service principal credentials (application ID and API access key).



For information about how to create an application ID see Adding an application and to create an access key or application secret see Configure a client application to access web APIs.

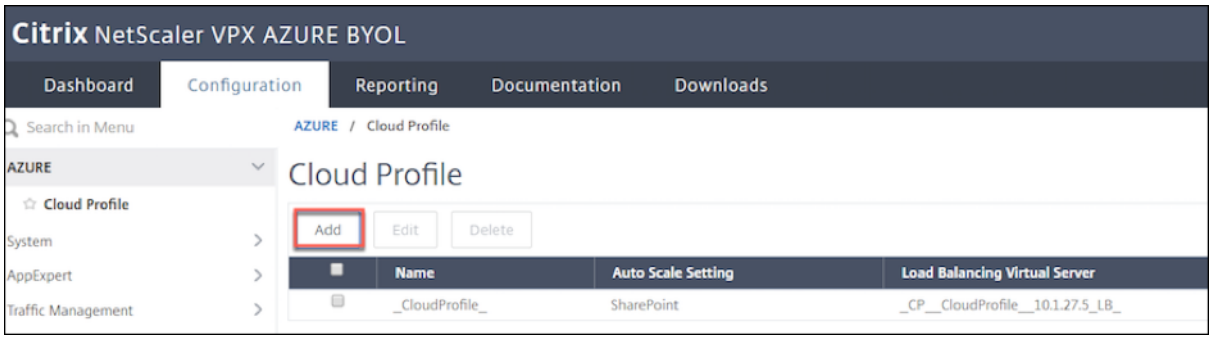
3. In the default cloud profile page, enter the details, as shown in the following example, and click Create.



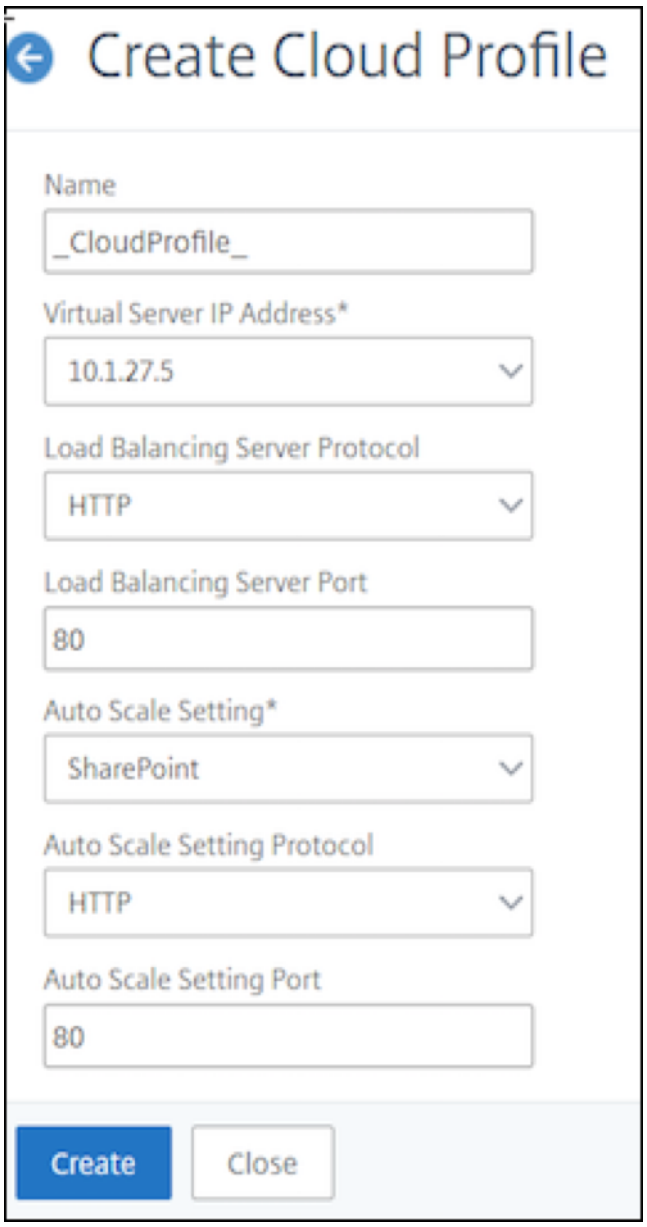
Points to keep in mind while creating a cloud profile

- The virtual server IP address is auto-populated from the free IP address available to the VPX instance. For more information, see Assign multiple IP addresses to virtual machines using the Azure portal.
- Autoscale setting is prepopulated from the VMSS Autoscale setting configured in the current resource group on your Azure account. For more information, see Overview of Autoscale with Azure virtual machine scale sets.
- While selecting the Auto Scaling Group protocol and port, ensure your servers listen on those protocol and ports and you bind the correct monitor in the service group. By default, the TCP monitor is used.
- For SSL Protocol type Autos Scaling, after you create the Cloud Profile the load balance virtual server or service group will be down because of a missing certificate. You can bind the certificate to the virtual server or service group manually.

After the first time logon, if you want to create a cloud profile, on the GUI go to System > Azure > Cloud Profile and click Add.



The Create Cloud Profile configuration page appears.



Cloud Profile creates a Citrix ADC load-balancing (LB) virtual server (virtual server) and a service group with members (servers) as the servers of the Auto Scaling Group. Your back-end servers must be reachable through the SNIP configured on the VPX instance.

To view autoscale-related information in the Azure portal, go to All service > Virtual machine scale set > Select Virtual machine scale set > Scaling.

Azure tags for Citrix ADC VPX deployment

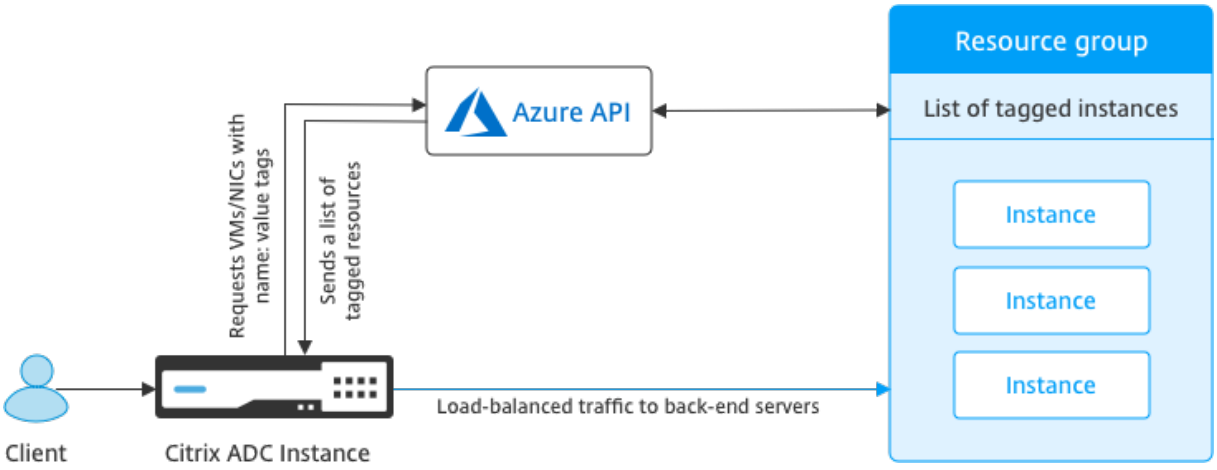
September 14, 2021

In the Azure cloud portal, you can tag resources with a name: value pair (such as Dept: Finance) to categorize and view resources across resource groups and, within the portal, across subscriptions. Tagging is helpful when you need to organize resources for billing or management or automation.

How Azure tag works for VPX deployment

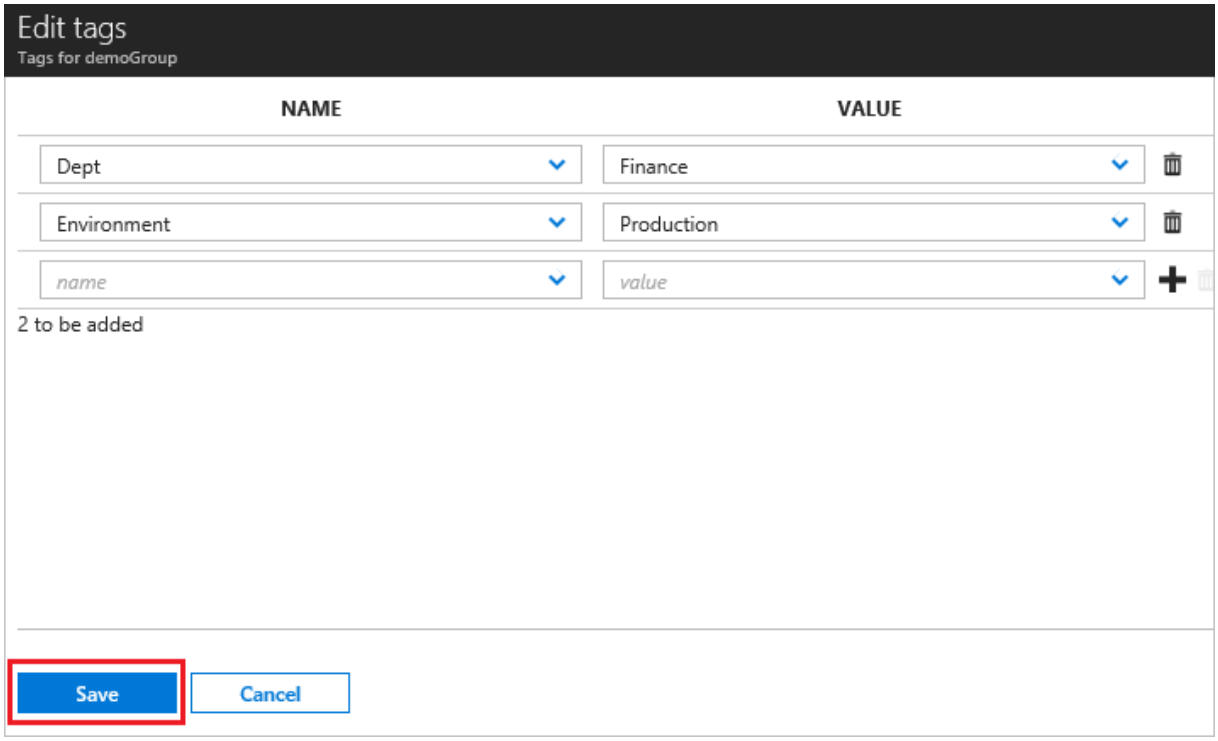
For Citrix ADC VPX standalone and high-availability instances deployed on Azure Cloud, now you can create load balancing service groups associated with an Azure tag. The VPX instance constantly monitors Azure virtual machines (back-end servers) and network interfaces (NICs), or both, with the respective tag and updates the service group accordingly.

The VPX instance creates the service group that load balances the back-end servers using tags. The instance queries the Azure API for all resources that are tagged with a particular tag name and tag value. Depending on the assigned poll period (by default 60 seconds), the VPX instance periodically polls the Azure API and retrieves the resources available with the tag name and tag values assigned in the VPX GUI. Whenever a VM or NIC with the appropriate tag is added or deleted, the ADC detects the respective change and adds or deletes the VM or NIC IP address from the service group automatically.



Before you begin

Before creating Citrix ADC load balancing service groups, add a tag to the servers in Azure. You can assign the tag to either the virtual machine or to NIC.



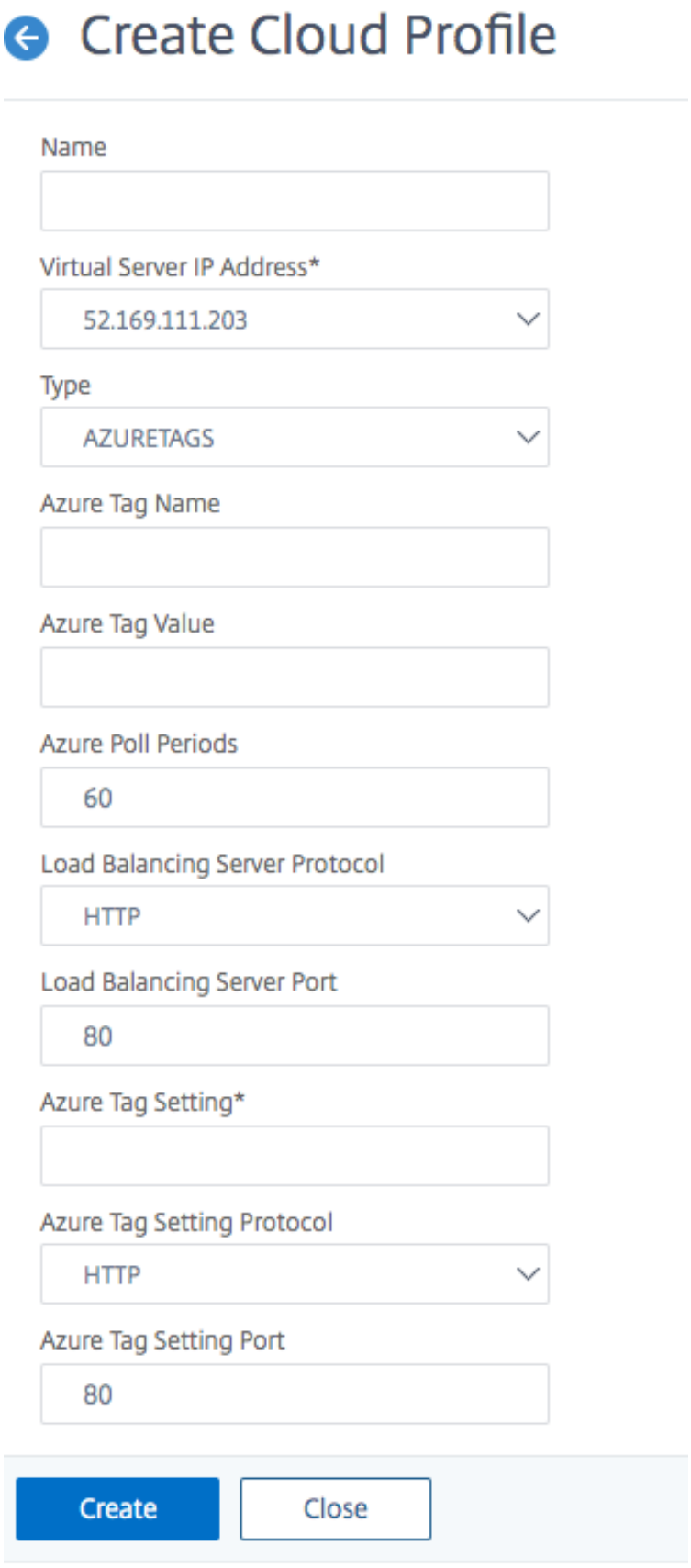
For more information about adding Azure tags, see Microsoft document Use tags to organize your Azure resources.

> Note
> ADC CLI commands to add Azure tag settings support tag names and tag values that start only with numerals or alphabets and not other keyboard characters.

How to add Azure tag settings by using VPX GUI

You can add the Azure tag cloud profile to a VPX instance by using the VPX GUI so that the instance can load balance the back-end servers using the specified tag. Follow these steps:

1. From the VPX GUI, go to **Configuration > Azure > Cloud Profile**.
2. Click Add to create a cloud profile. The cloud profile window opens.



1. Enter values for the following fields:
   - Name: Add a name for your profile
   - Virtual Server IP Address: The virtual server IP address is auto-populated from the free IP address available to the VPX instance. For more information, see Assign multiple IP addresses to virtual machines using the Azure portal.
   - Type: From the menu, select AZURETAGS.
   - Azure Tag Name: Enter the name that you have assigned to the VMs or NICs in the Azure portal.
   - Azure Tag Value: Enter the value that you have assigned to the VMs or NICs in Azure portal.
   - Azure Poll Periods: By default the poll period is 60 seconds, which is the minimum value. You can change it according to your requirement.
   - Load Balancing Server Protocol: Select the protocol that your load balancer listens on.
   - Load Balancing Server Port: Select the port that your load balancer listens on.
   - Azure tag setting: The name of the service group that will be created for this cloud profile.
   - Azure Tag Setting Protocol: Select the protocol that your back-end servers listen on.
   - Azure Tag Setting Port: Select the port that your back-end servers listen on.
2. Click **Create**.

A load-balancer virtual server and a service group are created for the tagged VMs or NICs. To see the load balancer virtual server, from the VPX GUI, navigate to **Traffic Management > Load Balancing > Virtual Servers**.

How to add Azure tag settings by using VPX CLI

Type the following command on Citrix ADC CLI to create a cloud profile for Azure tags.

```
add cloud profile `<profile name>` -type azuretags -vServerName `<
    vserver name>` -serviceType HTTP -IPAddress `<vserver IP address>` -
    port 80 -serviceGroupName `<service group name>` -
    boundServiceGroupSvcType HTTP -vsvrbindsvcport 80 -azureTagName `<
    Azure tag specified on Azure portal>` -azureTagValue `<Azure value
    specified on the Azure portal>` -azurePollPeriod 60

<!--NeedCopy-->
```

> Important
> You must save all configurations; otherwise, the configurations are lost after you restart the instance. Type `save config`.

Example 1: Here's a sample command for a cloud profile for HTTP traffic of all Azure VMs/NICs tagged with the "myTagName/myTagValue" pair:


```
add cloud profile MyTagCloudProfile -type azuretags -vServerName
    MyTagVServer -serviceType HTTP -IPAddress 40.115.116.57 -port 80 -
    serviceGroupName MyTagsServiceGroup -boundServiceGroupSvcType HTTP -
    vsvrbindsvcport 80 -azureTagName myTagName -azureTagValue myTagValue
     -azurePollPeriod 60
Done
<!--NeedCopy-->
```

To display the cloud profile, type `show cloudprofile`.

Example 2: The following CLI command prints information about the newly added cloud profile in example 1.

```
show cloudprofile
1)    Name:  MyTagCloudProfile Type:  azuretags         VServerName:
    MyTagVServer ServiceType:  HTTP       IPAddress:  52.178.209.133
    Port:  80               ServiceGroupName:  MyTagsServiceGroup
    BoundServiceGroupSvcType:  HTTP
       Vsvrbindsvcport:  80     AzureTagName:  myTagName AzureTagValue:
           myTagValue AzurePollPeriod:  60     GraceFul:  NO
          Delay:  60
<!--NeedCopy-->
```

To remove a cloud profile, type rm cloud profile <cloud profile name>

Example 3: The following command removes the cloud profile created in example 1.

```
> rm cloudprofile MyTagCloudProfile
  Done
<!--NeedCopy-->
```

Troubleshooting

Issue: In very rare cases, the “rm cloud profile” CLI command might fail to remove service group and servers associated with the deleted cloud profile. This happens when the command is issued seconds before the poll period of the cloud profile being deleted elapses.

Solution: Manually delete the remaining service groups by entering the following CLI command for each of the remaining service groups:

```
#> rm servicegroup <serviceGroupName>

<!--NeedCopy-->
```


Also remove each of the remain servers by entering the following CLI command for each of the remaining servers:

```
#> rm server <name>
<!--NeedCopy-->
```

Issue: If you add an Azure tag setting to a VPX instance by using CLI, the rain_tags process continues to run on an HA pair node after a warm reboot.

Solution: Manually terminate the process on the secondary node after a warm reboot. From the CLI of the secondary HA node exit to the shell prompt:

```
#> shell

<!--NeedCopy-->
```

Use the following command to kill the rain_tags process:

```
# PID=`ps -aux | grep rain_tags | awk '{
 print $2 }
 '`; kill -9 $PID

<!--NeedCopy-->
```

Issue: Back-end servers might not be reachable and reported as DOWN by the VPX instance, in spite of being healthy.
Solution: Make sure that the VPX instance can reach the tagged IP address corresponding to the back-end server. For a tagged NIC, this is the NIC IP address; whereas for a tagged VM, this is the VM's primary IP address. If the VM/NIC resides on a different Azure VNet, make sure that VNet peering is enabled.

Configure GSLB on Citrix ADC VPX instances

September 14, 2021

Citrix ADC appliances configured for global server load balancing (GSLB) provide disaster recovery and continuous availability of applications by protecting against points of failure in a WAN. GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers if there is an outage.

This section describes how to enable GSLB on VPX instances on two sites in a Microsoft Azure environment, by using Windows PowerShell commands.

> **Note**
>
> For more information about GSLB, see Global Server Load Balancing.

You can configure GSLB on a Citrix ADC VPX instance on Azure, in two steps:

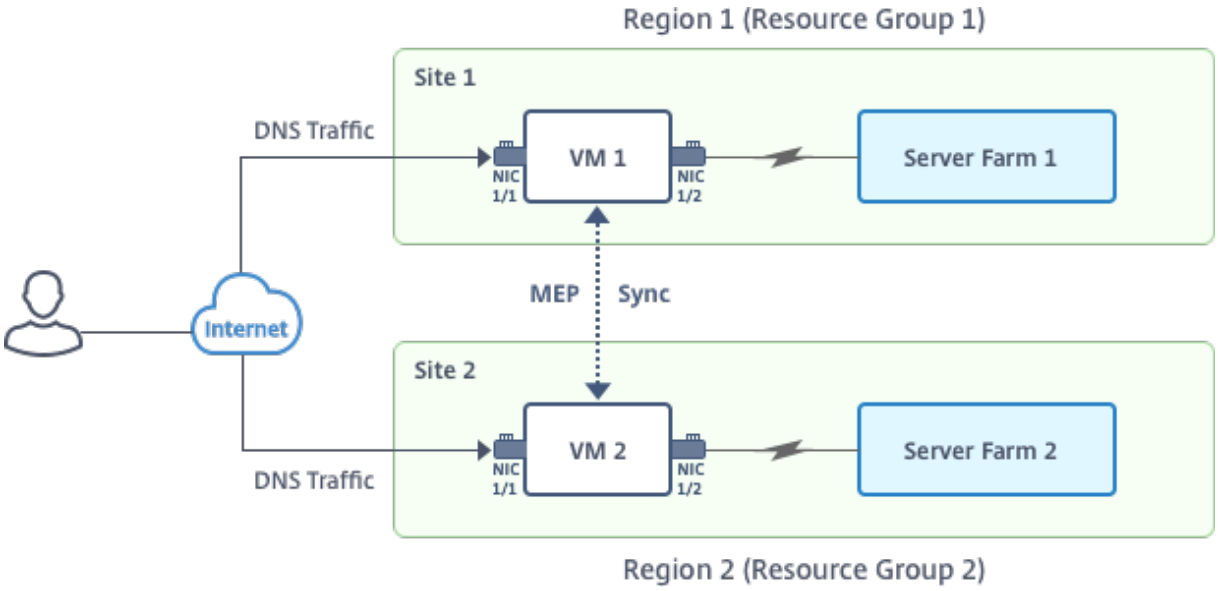
1. Create a VPX instance with multiple NICs and multiple IP addresses, on each site.
2. Enable GSLB on the VPX instances.

> **Note**
>
> For more information about configuring multiple NICs and IP addresses see: Configure multiple IP addresses for a Citrix ADC VPX instance in standalone mode by using PowerShell commands

Scenario

This scenario includes two sites - Site 1 and Site 2. Each site has a VM (VM1 and VM2) configured with multiple NICs, multiple IP addresses, and GSLB.

Figure. GSLB setup implemented across two sites - Site 1 and Site 2.



In this scenario, each VM has three NICs - NIC 0/1, 1/1, and 1/2. Each NIC can have multiple private and public IP addresses. The NICs are configured for the following purposes.

- NIC 0/1: to serve management traffic
- NIC 1/1: to serve client-side traffic
- NIC 1/2: to communicate with back-end servers

For information about the IP addresses configured on each NIC in this scenario, see the IP configuration details section.

Parameters

Following are sample parameters settings for this scenario in this document. You can use different settings if you want.

```
$location="West Central US"

$vnetName="NSVPX-vnet"

$RGName="multiIP-RG"

$prmStorageAccountName="multiipstorageaccnt"

$avSetName="MultiIP-avset"

$vmSize="Standard\_DS3\_V2"
<!--NeedCopy-->
```

Note: The minimum requirement for a VPX instance is 2 vCPUs and 2 GB RAM.

```
$publisher="citrix"

$offer="netscalervpx111"

$sku="netscalerbyol"

$version="latest"

$vmNamePrefix="MultiIPVPX"

$nicNamePrefix="MultiipVPX"

$osDiskSuffix="osdiskdb"

$numberOfVMs=1

$ipAddressPrefix="10.0.0."

$ipAddressPrefix1="10.0.1."

$ipAddressPrefix2="10.0.2."

$pubIPName1="MultiIP-pip1"

$pubIPName2="MultiIP-pip2"
```



```
26
27 $IpConfigName1="IPConfig1"
28
29 $IPConfigName2="IPConfig-2"
30
31 $IPConfigName3="IPConfig-3"
32
33 $IPConfigName4="IPConfig-4"
34
35 $frontendSubnetName="default"
36
37 $backendSubnetName1="subnet\_1"
38
39 $backendSubnetName2="subnet\_2"
40
41 $suffixNumber=10
42 <!--NeedCopy-->
```

Create a VM

Follow steps 1–10 to create VM1 with multiple NICs and multiple IP addresses, by using PowerShell commands:

1. Create resource group
2. Create storage account
3. Create availability set
4. Create virtual network
5. Create public IP address
6. Create NICs
7. Create VM config object
8. Get credentials and set OS properties for the VM
9. Add NICs
10. Specify OS disk and create VM

After you complete all the steps and commands to create VM1, repeat these steps to create VM2 with parameters specific to it.

Create resource group


```
New-AzureRMResourceGroup -Name $RGName -Location $location
<!--NeedCopy-->
```

Create storage account

```
$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $RGName -Type Standard_LRS
     -Location $location
<!--NeedCopy-->
```

Create availability set

```
$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $RGName -Location $location
<!--NeedCopy-->
```

Create virtual network

1. Add subnets.

```
$subnet1=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontendSubnetName -AddressPrefix "10.0.0.0/24"
$subnet2=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName1 -AddressPrefix "10.0.1.0/24"
$subnet3=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName2 -AddressPrefix "10.0.2.0/24"
<!--NeedCopy-->
```

2. Add virtual network object.

```
$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $RGName -Location $location -AddressPrefix 10.0.0.0/16 -Subnet
     $subnet1, $subnet2, $subnet3
<!--NeedCopy-->
```

3. Retrieve subnets.

```
$frontendSubnet=$vnet.Subnets|?{
 $_.Name -eq $frontendSubnetName }

$backendSubnet1=$vnet.Subnets|?{
```



```
 $_.Name -eq $backendSubnetName1 }

$backendSubnet2=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName2 }

<!--NeedCopy-->
```

Create public IP address

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
<!--NeedCopy-->
```

Create NICs

Create NIC 0/1

```
$nic1Name=$nicNamePrefix + $suffixNumber + "-Mgmnt"
$ipAddress1=$ipAddressPrefix + $suffixNumber
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    SubnetId $frontendSubnet.Id -PublicIpAddress $pip1 -PrivateIpAddress
     $ipAddress1 -Primary
$nic1=New-AzureRMNetworkInterface -Name $nic1Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig1
<!--NeedCopy-->
```

Create NIC 1/1

```
$nic2Name $nicNamePrefix + $suffixNumber + "-frontend"
$ipAddress2=$ipAddressPrefix1 + ($suffixNumber)
$ipAddress3=$ipAddressPrefix1 + ($suffixNumber + 1)
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    PublicIpAddress $pip2 -SubnetId $backendSubnet1.Id  -
    PrivateIpAddress $ipAddress2  -Primary
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress3
nic2=New-AzureRMNetworkInterface -Name $nic2Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig2, $IpConfig3
<!--NeedCopy-->
```

Create NIC 1/2


```
$nic3Name=$nicNamePrefix + $suffixNumber + "-backend"
$ipAddress4=$ipAddressPrefix2 + ($suffixNumber)
$IPConfig4=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    SubnetId $backendSubnet2.Id -PrivateIpAddress $ipAddress4 -Primary
$nic3=New-AzureRMNetworkInterface -Name $nic3Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig4
<!--NeedCopy-->
```

Create VM config object

```
$vmName=$vmNamePrefix
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id
<!--NeedCopy-->
```

Get credentials and set OS properties

```
$cred=Get-Credential -Message "Type the name and password for VPX login
    ."
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version
<!--NeedCopy-->
```

Add NICs

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
    Primary
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id
<!--NeedCopy-->
```

Specify OS disk and create VM

```
$osDiskName=$vmName + "-" + $osDiskSuffix
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     +$osDiskName + ".vhd"
```



```
3 $vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
4 Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
5 New-AzureRMVM -VM $vmConfig -ResourceGroupName $RGName -Location
    $location
6 <!--NeedCopy-->
```

> **Note**
>
> Repeat steps 1–10 listed in "Create Multi-NIC VMs by Using PowerShell Commands" to create VM2 with parameters specific to VM2.

IP configuration details

The following IP addresses are used.

Table 1. IP addresses used in VM1

NIC	Private IP	Public IP (PIP)	Description
0/1	10.0.0.10	PIP1	Configured as NSIP (management IP)
1/1	10.0.1.10	PIP2	Configured as SNIP/GSLB Site IP
-	10.0.1.11	-	Configured as LB server IP. Public IP is not mandatory
1/2	10.0.2.10	-	Configured as SNIP for sending monitor probes to services; public IP is not mandatory

Table 2. IP addresses used in VM2

NIC	Internal IP	Public IP (PIP)	Description
0/1	20.0.0.10	PIP4	Configured as NSIP (management IP)

NIC	Internal IP	Public IP (PIP)	Description
1/1	20.0.1.10	PIP5	Configured as SNIP/GSLB Site IP
-	20.0.1.11	-	Configured as LB server IP. Public IP is not mandatory
1/2	20.0.2.10	-	Configured as SNIP for sending monitor probes to services; public IP is not mandatory

Here are sample configurations for this scenario, showing the IP addresses and initial LB configurations as created through the Citrix ADC VPX CLI for VM1 and VM2.

Here's an example configuration on VM1.

```
add ns ip 10.0.1.10 255.255.255.0 -mgmtAccess ENABLED
Add nsip 10.0.2.10 255.255.255.0
add service svc1 10.0.1.10 ADNS 53
add lb vserver v1 HTTP 10.0.1.11 80
add service s1 10.0.2.120 http 80
Add service s2 10.0.2.121 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

Here's an example configuration on VM2.

```
add ns ip 20.0.1.10 255.255.255.0  -mgmtAccess ENABLED
Add nsip 20.0.2.10 255.255.255.0
add service svc1 20.0.1.10 ADNS 53
add lb vserver v1 HTTP 20.0.1.11 80
Add service s1 20.0.2.90 http 80
Add service s2 20.0.2.91 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

Configure GSLB sites and other settings

Perform the tasks described in the following topic to configure the two GSLB sites and other necessary settings:

Global Server Load Balancing

For more information, see this support article: https://support.citrix.com/article/CTX110348

Here's an example GSLB configuration on VM1 and VM2.

```
enable ns feature LB GSLB
add gslb site site1 10.0.1.10 -publicIP PIP2
add gslb site site2 20.0.1.10 -publicIP PIP5
add gslb service site1_gslb_http_svc1 10.0.1.11 HTTP 80 -publicIP PIP3
   -publicPort 80 -siteName site1
add gslb service site2_gslb_http_svc1 20.0.1.11 HTTP 80 -publicIP PIP6
   -publicPort 80 -siteName site2
add gslb vserver gslb_http_vip1 HTTP
bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
<!--NeedCopy-->
```

You've configured GSLB on Citrix ADC VPX instances running on Azure.

For additional information about how to configure GSLB on Citrix ADC VPX instances, click the following image to watch the video about Configuring Citrix ADC GSLB in Microsoft Azure.

Configure GSLB on an active-standby high-availability setup

September 14, 2021

You can configure global server load balancing (GSLB) on active-standby HA deployment on Azure in three steps:

1. Create a VPX HA pair on each GSLB site. See Configure a high-availability setup with multiple IP addresses and NICs for information about how to create an HA pair.
2. Configure the Azure Load Balancer (ALB) with the front-end IP address and rules to allow GSLB and DNS traffic.

   This step involves the following substeps. See the scenario in this section for the PowerShell commands used to complete these substeps.

a. Create a front-end `IPconfig` for GSLB site.

b. Create a back-end address pool with IP address of NIC 1/1 of nodes in HA.

c. Create load-balancing rules for following:

```
TCP/3011  -  gslb communication
TCP/3010 - gslb communication
UDP/53 - DNS communication
```

d. Associate back-end address pool with the LB rules created in step c.

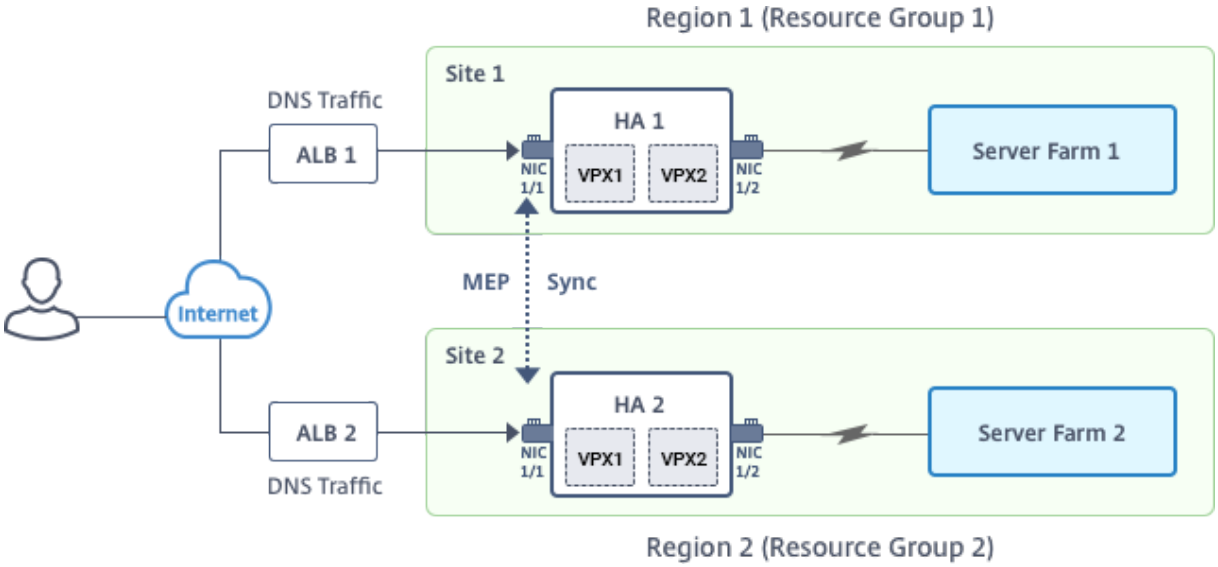
e. Update the network security group of NIC 1/1 of nodes in both the HA pair to allow the traffic for TCP 3010, TCP 3011 and UDP 53 ports.

3. Enable GSLB on each HA pair.

Scenario

This scenario includes two sites - Site 1 and Site 2. Each site has an HA pair (HA1 and HA2) configured with multiple NICs, multiple IP addresses, and GSLB.

Figure: GLSB on Active-Standy HA Deployment on Azure



In this scenario, each VM has three NICs - NIC 0/1, 1/1, and 1/2. The NICs are configured for the following purposes.

NIC 0/1: to serve management traffic

NIC 1/1: to serve client-side traffic

NIC 1/2: to communicate with back-end servers

Parameter Settings

Following are sample parameters settings for the ALB. You can use different settings if you want.

```
$locName="South east Asia"

$rgName="MulitIP-MultiNIC-RG"

$pubIPName4="PIPFORGSLB1"

$domName4="vpxgslbdns"

$lbName="MultiIPALB"

$frontEndConfigName2="FrontEndIP2"

$backendPoolName1="BackendPoolHttp"

$lbRuleName2="LBRuleGSLB1"

$lbRuleName3="LBRuleGSLB2"

$lbRuleName4="LBRuleDNS"

$healthProbeName="HealthProbe"
```

Configure ALB with the front-end IP address and rules to allow GSLB and DNS traffic

Step 1. **Create a public IP for GSLB site IP**

```
$pip4=New-AzureRmPublicIpAddress -Name $pubIPName4 -ResourceGroupName
    $rgName -DomainNameLabel $domName4 -Location $locName -
    AllocationMethod Dynamic


Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName | Add-
    AzureRmLoadBalancerFrontendIpConfig -Name $frontEndConfigName2 -
    PublicIpAddress $pip4 | Set-AzureRmLoadBalancer
```

Step 2. **Create LB rules and update the existing ALB**.

```
$alb = get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName


```



```
$frontendipconfig2=Get-AzureRmLoadBalancerFrontendIpConfig -
    LoadBalancer $alb -Name $frontEndConfigName2


$backendPool=Get-AzureRmLoadBalancerBackendAddressPoolConfig -
    LoadBalancer $alb -Name $backendPoolName1


$healthprobe=Get-AzureRmLoadBalancerProbeConfig -LoadBalancer $alb -
    Name $healthProbeName


$alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName2 -
    BackendAddressPool $backendPool -FrontendIPConfiguration
    $frontendipconfig2 -Protocol "Tcp" -FrontendPort 3011 -BackendPort
    3011 -Probe $healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer


$alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName3 -
    BackendAddressPool $backendPool -FrontendIPConfiguration
    $frontendipconfig2 -Protocol "Tcp" -FrontendPort 3010 -BackendPort
    3010 -Probe $healthprobe -EnableFloatingIP  | Set-
    AzureRmLoadBalancer


$alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName4 -
    BackendAddressPool $backendPool -FrontendIPConfiguration
    $frontendipconfig2 -Protocol "Udp" -FrontendPort 53 -BackendPort 53
    -Probe $healthprobe -EnableFloatingIP  | Set-AzureRmLoadBalancer
```

Enable GSLB on each high availability pair

Now you've two front-end IP addresses for each ALB: ALB 1 and ALB 2. One IP address is for the LB virtual server and the other for the GSLB site IP.

HA 1 has the following front-end IP addresses:

- FrontEndIPofALB1 (for LB virtual server)
- PIPFORGSLB1 (GSLB IP)

HA 2 has the following front-end IP addresses:

- FrontEndIPofALB2 (for LB virtual server)
- PIPFORGSLB2 (GSLB IP)

The following commands are used for this scenario.

```
enable ns feature LB GSLB

add service dnssvc PIPFORGSLB1 ADNS 53

add gslb site site1 PIPFORGSLB1 -publicIP PIPFORGSLB1

add gslb site site2 PIPFORGSLB2 -publicIP PIPFORGSLB2

add gslb service site1_gslb_http_svc1 FrontEndIPofALB1 HTTP 80 -
   publicIP FrontEndIPofALB1 -publicPort 80 -siteName site1

add gslb service site2_gslb_http_svc1 FrontEndIPofALB2 HTTP 80 -
   publicIP FrontEndIPofALB2 -publicPort 80 -siteName site2

add gslb vserver gslb_http_vip1 HTTP

bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

Related resources:

Configure GSLB on Citrix ADC VPX instances

Global Server Load Balancing

Configure address pools intranet IP for a Citrix Gateway appliance

September 14, 2021

In some situations, users who connect with the Citrix Gateway Plug-in need a unique IP address for a Citrix ADC Gateway appliance. When you enable address pools (also known as IP pooling) for a group, the Citrix Gateway appliance can assign a unique IP address alias to each user. You configure address pools by using intranet IP (IIP) addresses.

You can configure address pools on a Citrix Gateway appliance deployed on Azure by following this 2-step procedure:

- Registering the private IP addresses that are used in the address pool, in Azure
- Configuring address pools in the Citrix Gateway appliance

Register a private IP address in the Azure portal

In Azure, you can deploy a Citrix ADC VPX instance with multiple IP addresses. You can add IP addresses to a VPX instance in two ways:

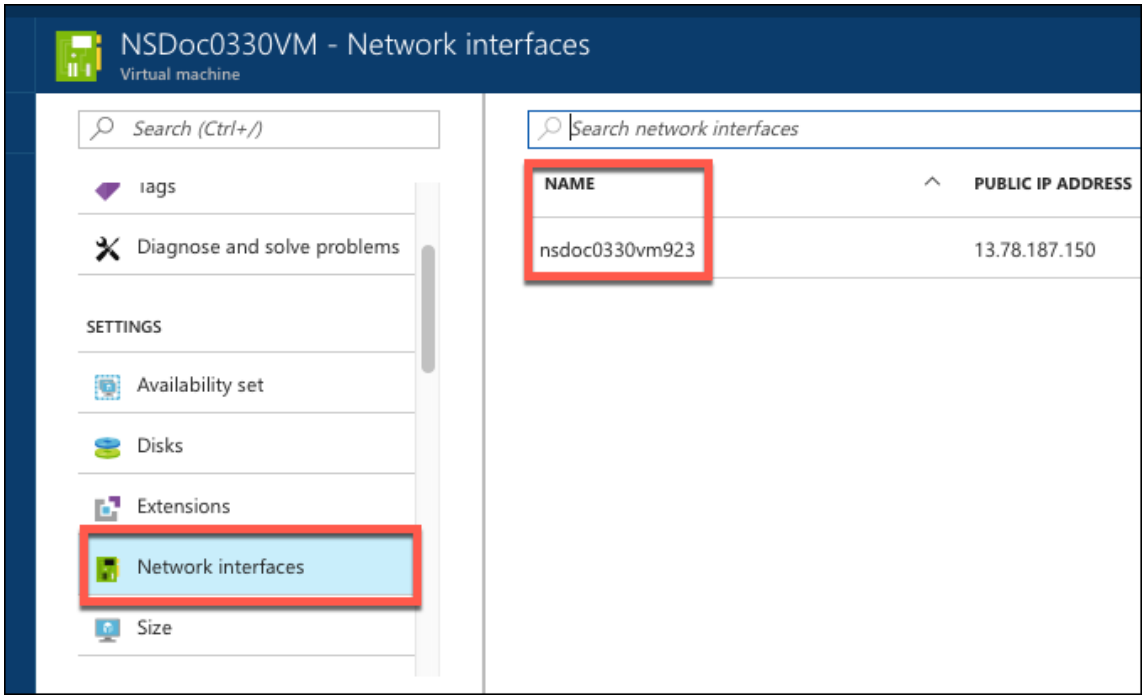
a. While provisioning a VPX instance

For more information about how to add multiple IP addresses while provisioning a VPX instance, see Configure multiple IP addresses for a Citrix ADC standalone instance. To add IP addresses by using PowerShell commands while provisioning a VPX instance, see Configure multiple IP addresses for a Citrix ADC VPX instance in standalone mode by using PowerShell commands.

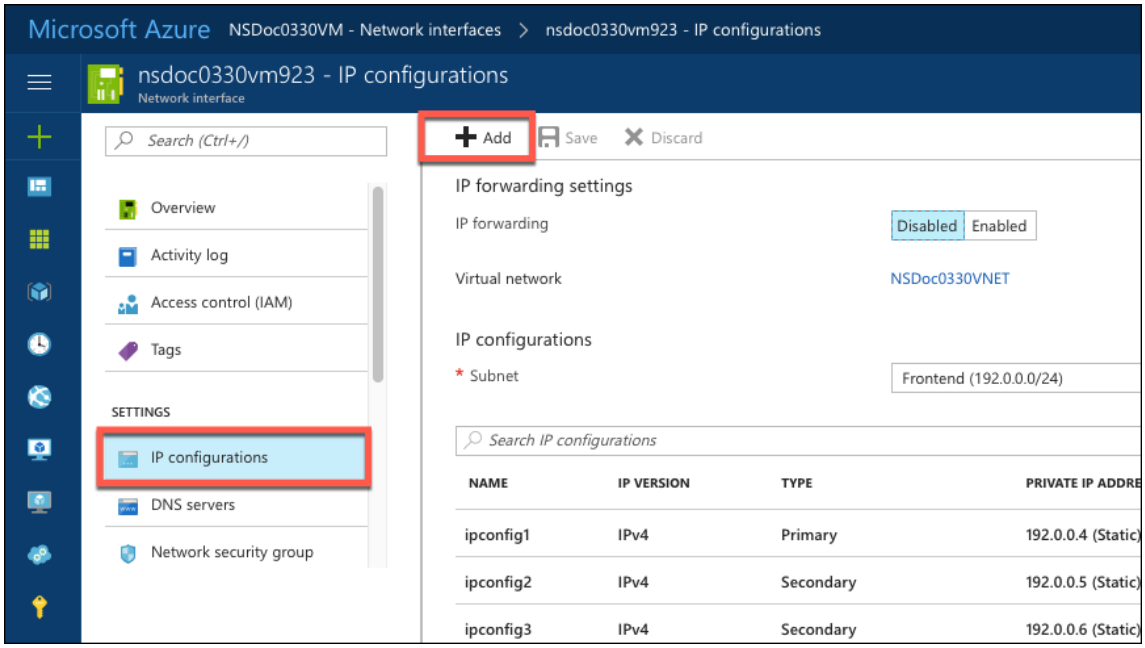
b. After provisioning a VPX instance

After you've provisioned a VPX instance, follow these steps to register a private IP address in the Azure portal, which you configure as an address pool in the Citrix Gateway appliance.

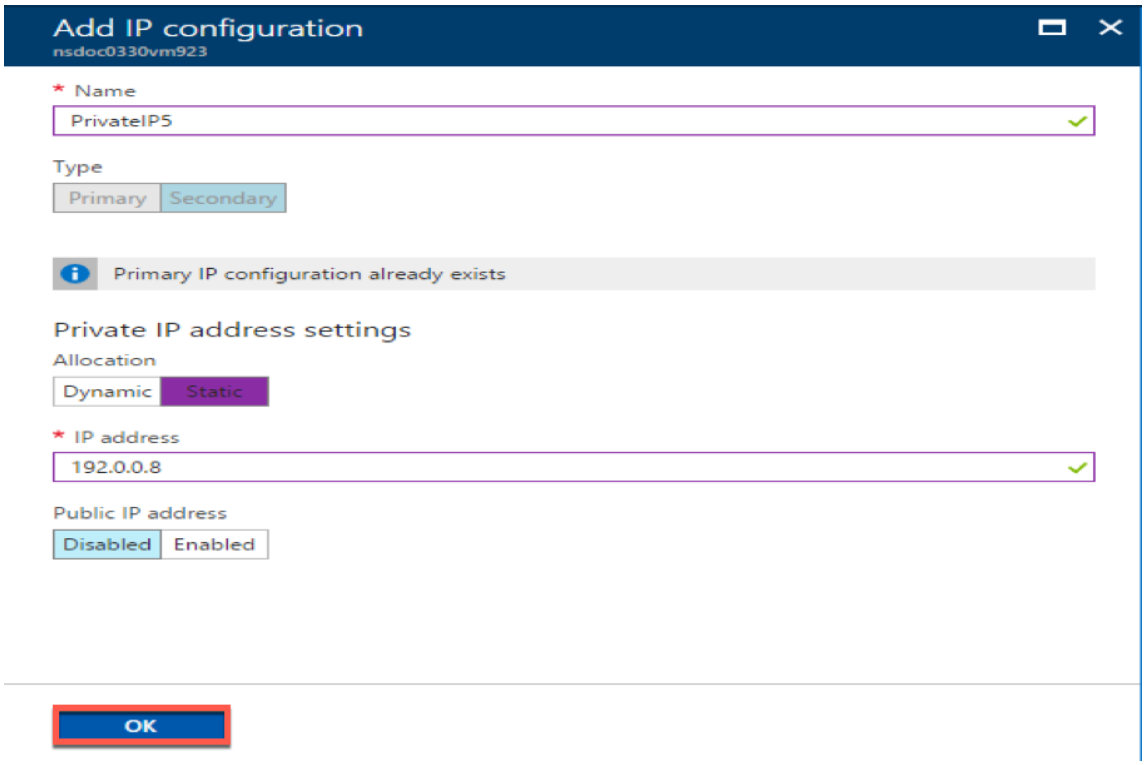
1. From Azure Resource Manager (ARM), go to the already created Citrix ADC VPX instance > **Network interfaces**. Choose the network interface which is bound to a subnet to which the IIP that you want to register belongs.



2. Click **IP Configurations**, and then click **Add**.



3. Provide the required details as shown in the example below and click **OK**.



Configure address pools in the Citrix Gateway appliance

For more information about how to configure address pools on the Citrix Gateway, see this Configuring Address Pools.

Limitation: You cannot bind a range of IIP addresses to users. Every IIP address that is used in an address pool must be registered.

Configure multiple IP addresses for a Citrix ADC VPX standalone instance by using PowerShell commands

September 14, 2021

In an Azure environment, a Citrix ADC VPX virtual appliance can be deployed with multiple NICs. Each NIC can have multiple IP addresses. This section describes how to deploy a Citrix ADC VPX instance with a single NIC and multiple IP addresses, by using PowerShell commands. You can use the same script for multi-NIC and multi-IP deployment.

> **Note**
>
> In this document, IP-Config refers to a pair of IP addresses, public IP, and private IP, that is associated with an individual NIC. For more information, see the Azure terminology section.

Use case

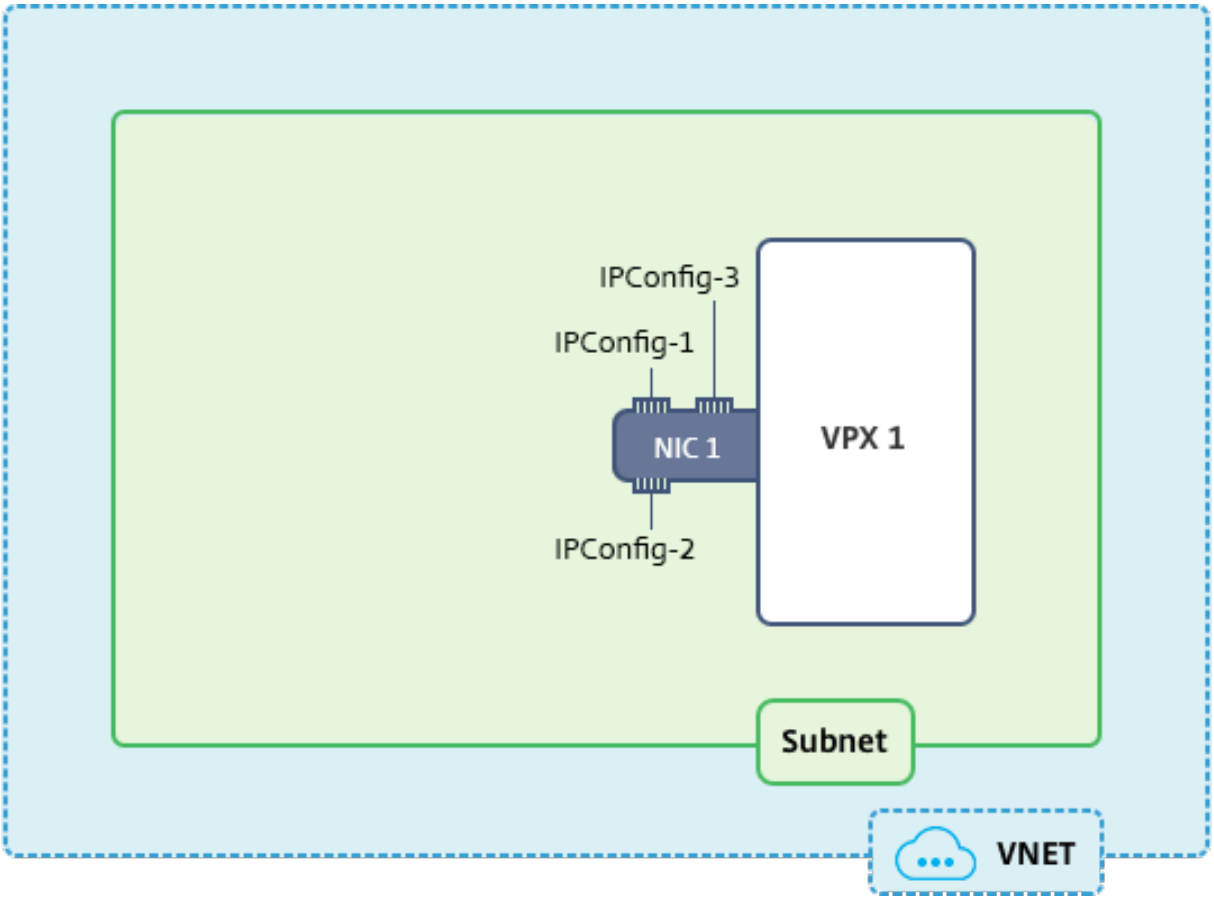
In this use case, a single NIC is connected to a virtual network (VNET). The NIC is associated with three IP configurations, as shown in the following table.

IP Config	Associated with
IPConfig-1	Static public IP address; static private IP address
IPConfig-2	Static public IP address; static private address
IPConfig-3	Static private IP address

> **Note**
>
> IPConfig-3 is not associated with any public IP address.

Diagram: Topology

Here is the visual representation of the use case.



> **Note**
>
> In a multi-NIC, multi-IP Azure Citrix ADC VPX deployment, the private IP address associated with the primary (first) `IPConfig` of the primary (first) NIC is automatically added as the management NSIP address of the appliance. The remaining private IP addresses associated with `IPConfigs` must be added in the VPX instance as VIPs or SNIPs by using the `add ns ip` command, as determined by your requirements.

Here is the summary of the steps required for configuring multiple IP addresses for a Citrix ADC VPX virtual appliance in standalone mode:

1. Create Resource Group
2. Create Storage Account
3. Create Availability Set
4. Create Network service group
5. Create Virtual Network
6. Create Public IP Address
7. Assign IP Configuration
8. Create NIC
9. Create Citrix ADC VPX Instance

10. Check NIC Configurations
11. Check VPX-side Configurations

Script

Parameters

Following are sample parameters settings for the use case in this document. You can use different settings if you want.

$locName=”westcentralus”

$rgName=”Azure-MultiIP”

$nicName1=”VM1-NIC1”

$vNetName=”Azure-MultiIP-vnet”

$vNetAddressRange=”11.6.0.0/16”

$frontEndSubnetName=”frontEndSubnet”

$frontEndSubnetRange=”11.6.1.0/24”

$prmStorageAccountName=”multiipstorage”

$avSetName=”multiip-avSet”

$vmSize=”Standard_DS4_V2” (This parameter creates a VM with up to four NICs.)

Note: The minimum requirement for a VPX instance is 2 vCPUs and 2 GB RAM.

$publisher=”Citrix”

$offer=”netscalervpx110-6531” (You can use different offers.)

$sku=”netscalerbyol” (According to your offer, the SKU can be different.)

$version=”latest”

$pubIPName1=”PIP1”

$pubIPName2=”PIP2”

$domName1=”multiipvpx1”

$domName2=”multiipvpx2”

$vmNamePrefix=”VPXMultiIP”

$osDiskSuffix=”osmultiipalbdiskdb1”

Network Security Group (NSG)-related information:

$nsgName=”NSG-MultiIP”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

$IpConfigName1=”IPConfig1”

$IPConfigName2=”IPConfig-2”

$IPConfigName3=”IPConfig-3”

1. Create Resource Group

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. Create Storage Account

```
$prmStorageAccount = New-AzureRMStorageAccount -Name $prmStorageAccountName
 -ResourceGroupName $rgName -Type Standard_LRS -Location $locName
```

3. Create Availability Set

```
$avSet = New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
$rgName -Location $locName
```

4. Create Network Security Group

1. Add rules. You must add a rule to the network security group for any port that serves traffic.

```
$rule1=New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -Description
 "Allow HTTP"-Access Allow -Protocol Tcp -Direction Inbound -Priority
101 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
 * -DestinationPortRange 80
$rule2=New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -Description
 "Allow HTTPS"-Access Allow -Protocol Tcp -Direction Inbound -Priority
110 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
 * -DestinationPortRange 443
$rule3=New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -Description
 "Allow SSH"-Access Allow -Protocol Tcp -Direction Inbound -Priority
120 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
 * -DestinationPortRange 22
```


2. Create network security group object.

```
$nsg=New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

5. Create Virtual Network

1. Add subnets.

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $frontEndSubnetName
 -AddressPrefix $frontEndSubnetRange
```

2. Add virtual network object.

```
$vnet=New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
$rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
$frontendSubnet
```

3. Retrieve subnets.

```
$subnetName="frontEndSubnet"
$subnet1=$vnet.Subnets|?{ $_.Name -eq $subnetName }
```

6. Create Public IP Address

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
$rgName -DomainNameLabel $domName1 -Location $locName -AllocationMethod
Static
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
$rgName -DomainNameLabel $domName2 -Location $locName -AllocationMethod
Static
```

> **Note**
>
> Check availability of domain names before using.
>
> Allocation method for IP addresses can be dynamic or static.

7. Assign IP Configuration

In this use case, consider the following points before assigning IP addresses:

- IPConfig-1 belongs to subnet1 of VPX1.
- IPConfig-2 belongs to subnet 1 of VPX1.
- IPConfig-3 belongs to subnet 1 of VPX1.

> **Note**
>
> When you assign multiple IP configurations to a NIC, one configuration must be assigned as primary.

```
$IPAddress1="11.6.1.27"
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -Subnet $subnet1 -PrivateIpAddress $IPAddress1 -PublicIpAddress $pip1 -Primary
$IPAddress2="11.6.1.28"
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -Subnet $subnet1 -PrivateIpAddress $IPAddress2 -PublicIpAddress $pip2
$IPAddress3="11.6.1.29"
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -Subnet $subnet1 -PrivateIpAddress $IPAddress3 -Primary
```

Use a valid IP address that meets your subnet requirements and check its availability.

8. Create NIC

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName $rgName -Location $locName -IpConfiguration $IpConfig1,$IpConfig2,$IPConfig3 -NetworkSecurityGroupId $nsg.Id
```

9. Create Citrix ADC VPX Instance

1. Initialize variables.

   ```
   $suffixNumber = 1
   $vmName = $vmNamePrefix + $suffixNumber
   ```

2. Create VM config object.

   ```
   $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avSet.Id
   ```

3. Set credentials, OS, and image.

   ```
   $cred=Get-Credential -Message "Type the name and password for VPX login."
   $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName $vmName -Credential $cred
   $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher -Offer $offer -Skus $sku -Version $version
   ```


4. Add NIC.

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
Primary
```

> **Note**
>
> In a multi-NIC VPX deployment, one NIC must be primary. So, "-Primary" must be appended while adding that NIC to the VPX instance.

5. Specify OS disk and create VM.

```
$osDiskName=$vmName + "-"+ $osDiskSuffix1
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString()+ "vhds/"+
 $osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
$osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
$locName
```

10. Check NIC Configurations

After the VPX instance starts, you can check the IP addresses allocated to `IPConfigs` of the VPX NIC by using the following command.

```
$nic.IPConfig
```

11. Check VPX-side Configurations

When the Citrix ADC VPX instance starts, a private IP address associated with primary `IPconfig` of the primary NIC is added as the NSIP address. The remaining private IP addresses must be added as VIP or SNIP addresses, as determined by your requirements. Use the following command.

```
add nsip <Private IPAddress><netmask> -type VIP/SNIP
```

You've now configured multiple IP addresses for a Citrix ADC VPX instance in standalone mode.

Additional PowerShell scripts for Azure deployment

September 14, 2021

This section provides the PowerShell cmdlets with which you can perform the following configurations in Azure PowerShell:

- Provision a Citrix ADC VPX standalone instance
- Provision a Citrix ADC VPX pair in a high availability setup with an Azure external load balancer
- Provision a Citrix ADC VPX pair in a high availability setup with Azure internal load balancer

Also see the following topics for configurations that you can perform by using PowerShell commands:

- Configure a high-availability setup with multiple IP addresses and NICs by using PowerShell commands
- Configure GSLB on Citrix ADC VPX instances
- Configure GSLB on a NetScaler active-standby high-availability setup
- Configure multiple IP addresses for a Citrix ADC VPX instance in standalone mode by using PowerShell commands
- Configure multiple Azure VIPs for a standalone VPX instance

Provision a Citrix ADC VPX standalone instance

1. **Create a resource group**

   The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. The location specified here is the default location for resources in that resource group. Make sure all commands to create a load balancer use the same resource group.

   ```
   $rgName="<resource group name>"
   $locName="<location name, such as West US>
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   For example:

   ```
   $rgName = "ARM-VPX"
   $locName = "West US"
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. **Create a storage account**

   Choose a unique name for your storage account that contains only lowercase letters and numbers.

   `$saName="<storage account name>"`
   `$saType="<storage account type>"`, specify one: `Standard_LRS`, `Standard_GRS`, `Standard_RAGRS`, or `Premium_LRS`


```
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -
Type $saType -Location $locName
```

For example:

```
$saName="vpxstorage"
$saType="Standard\_LRS"
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. **Create an availability set**

   Availability set helps to keep your virtual machines available during downtime, such as during maintenance. A load balancer configured with an availability set ensures that your application is always available.

   ```
   $avName="<availability set name>"
   ```

   ```
   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -
   Location $locName
   ```

4. **Create a virtual network**

   Add a new virtual network with at least one subnet, if the subnet was not created previously.

   ```
   $FrontendAddressPrefix="10.0.1.0/24"
   $BackendAddressPrefix="10.0.2.0/24"
   $vnetAddressPrefix="10.0.0.0/16"
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet
    -AddressPrefix $FrontendAddressPrefix
   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet
    -AddressPrefix $BackendAddressPrefix
   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -
   Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet
   ,$backendSubnet
   ```

   For example:

   ```
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       backendSubnet -AddressPrefix $BackendAddressPrefix

   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName
        -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet
       $frontendSubnet,$backendSubnet
   ```



```
<!--NeedCopy-->
```

5. **Create a NIC**

   Create a NIC and associate the NIC with the Citrix ADC VPX instance. The front end Subnet created in the above procedure is indexed at 0 and the back end Subnet is indexed at 1. Now create NIC in one of the three following ways:

   a) *NIC with Public IP address*

   ```
   $nicName="<name of the NIC of the VM>"

   $pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
   $rgName -Location $locName -AllocationMethod Dynamic

   $nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
   $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
   PublicIpAddressId $pip.Id
   ```

   b) *NIC with Public IP and DNS label*

   ```
   $nicName="<name of the NIC of the VM>"

   $domName="<domain name label>"

   $pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
   $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod
   Dynamic
   ```

   Before assigning $domName, check it is available or not by using command:

   ```
   Test-AzureRmDnsAvailability -DomainQualifiedName $domName -Location
   $locName

   $nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
   $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
   PublicIpAddressId $pip.Id
   ```

   For example:

   ```
   $nicName="frontendNIC"

   $domName="vpxazure"

   $pip = New-AzureRmPublicIpAddress -Name $nicName -
       ResourceGroupName $rgName -DomainNameLabel $domName -Location
       $locName -AllocationMethod Dynamic

   ```



```
$nic = New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -SubnetId $vnet.
    Subnets\[0\].Id -PublicIpAddressId $pip.Id
<!--NeedCopy-->
```

c) *NIC with Dynamic Public Address and Static Private IP address*

Make sure that the private (static) IP address you add to the VM must be the same range as that of the subnet specified.

```
$nicName="<name of the NIC of the VM>"

$staticIP="<available static IP address on the subnet>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
$rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
$rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP
```

6. **Create a virtual object**

```
$vmName="<VM name>"

$vmSize="<VM size string>"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
$rgName

$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
 $avset.Id
```

7. **Get the Citrix ADC VPX image**

```
$pubName="<Image publisher name>"

$offerName="<Image offer name>"

$skuName="<Image SKU name>"

$cred=Get-Credential -Message "Type the name and password of the local
administrator account."
```

Provide your credentials that is used to log in into VPX

```
$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -
Credential $cred -Verbose

$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer
$offerName -Skus $skuName -Version "latest"
```



```
$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

For example:

```
$pubName="citrix"
```

The following command is used for displaying all offers from Citrix:

```
1 Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName |
     Select Offer
2
3 $offerName="netscalervpx110-6531"
4 <!--NeedCopy-->
```

The following command is used to know SKU offered by publisher for specific offer name:

```
Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -Offer
$offerName | Select Skus
```

8. **Create a virtual machine**

```
$diskName="<name identifier for the disk in Azure storage, such as
OSDisk>"
```

For example:

```
1 $diskName="dynamic"
2
3 $pubName="citrix"
4
5 $offerName="netscalervpx110-6531"
6
7 $skuName="netscalerbyol"
8
9 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
     Name $saName
10
11 $osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds/"
     + $diskName  + ".vhd"
12
13 $vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri
      -CreateOption fromImage
14 <!--NeedCopy-->
```

When you create VM from Images present in marketplace, use the following command to specify the VM plan:

```
Set-AzureRmVMPlan -VM $vm -Publisher $pubName -Product $offerName -Name
 $skuName
```



```
New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm
```

Provision a Citrix ADC VPX pair in a high availability setup with an Azure external load balancer

Log on to AzureRmAccount using your Azure user credentials.

1. **Create a resource group**

   The location specified here is the default location for resources in that resource group. Make sure that all commands used to create a load balancer use the same resource group.

   ```
   $rgName="<resource group name>"
   ```

   ```
   $locName="<location name, such as West US>"
   ```

   ```
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   For example:

   ```
   $rgName = "ARM-LB-NS"

   $locName = "West US"

   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. **Create a storage account**

   Choose a unique name for your storage account that contains only lowercase letters and numbers.

   ```
   $saName="<storage account name>"
   ```

   ```
   $saType="<storage account type>"
   ```

   , specify one: `Standard_LRS`, `Standard_GRS`, `Standard_RAGRS`, or `Premium_LRS`

   ```
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -
   Type $saType -Location $locName
   ```

   For example:

   ```
   $saName="vpxstorage"

   $saType="Standard_LRS"

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
       -Type $saType -Location $locName
   <!--NeedCopy-->
   ```


3. **Create an availability set**

   A load balancer configured with an availability set ensures that your application is always available.

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -
   Location $locName
   ```

4. **Create a virtual network**

   Add a new virtual network with at least one subnet, if the subnet was not created previously.

   ```
   1  $vnetName = "LBVnet"
   2
   3  $FrontendAddressPrefix="10.0.1.0/24"
   4
   5  $BackendAddressPrefix="10.0.2.0/24"
   6
   7  $vnetAddressPrefix="10.0.0.0/16"
   8
   9  $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
          frontendSubnet -AddressPrefix $FrontendAddressPrefix
   10
   11 $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
          backendSubnet -AddressPrefix $BackendAddressPrefix
   12
   13 $vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
           $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
          Subnet $frontendSubnet,$backendSubnet
   14 <!--NeedCopy-->
   ```

   Note: Choose the AddressPrefix parameter value as per your requirement.

   Assign front end and back end subnet to the virtual network that you created earlier in this step.

   If the front end subnet is the first element of array VNet, subnetId must be $vnet.Subnets[0].Id.

   If the front end subnet is the second element in the array, the subnetId must be $vnet.Subnets[1].Id, and so on.

5. **Configure front end IP address and create back end address pool**

   Configure a front end IP address for the incoming load balancer network traffic and create a back end address pool to receive the load balanced traffic.

   ```
   1  $pubName="PublicIp1"
   2
   ```



```
$publicIP1 = New-AzureRmPublicIpAddress -Name $pubName -
    ResourceGroupName $rgName -Location $locName -AllocationMethod
    Static -DomainNameLabel nsvpx
<!--NeedCopy-->
```

Note: Check for the availability of the value for DomainNameLabel.

```
$FIPName = "ELBFIP"

$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name
    $FIPName -PublicIpAddress $publicIP1

$BEPool = "LB-backend-Pool"

$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -
    Name $BEPool
<!--NeedCopy-->
```

6. **Create a health probe**

   Create a TCP health probe with port 9000 and interval 5 seconds.

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name
    HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -
    ProbeCount 2
<!--NeedCopy-->
```

7. **Create a load balancing rule**

   Create an LB rule for each service that you are load balancing.

   For example:

   You can use the following example to load balance HTTP service.

```
$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP-LB" -
    FrontendIpConfiguration $frontendIP1 -BackendAddressPool
    $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
     80 -BackendPort 80
<!--NeedCopy-->
```

8. **Create inbound NAT rules**

   Create NAT rules for services that you are not load balancing.

   For example, when creating an SSH access to a Citrix ADC VPX instance.

   Note: Protocol-FrontEndPort-BackendPort triplet must not be the same for two NAT rules.


```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH1     -FrontendIpConfiguration $frontendIP1 -Protocol
    TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -
    FrontendPort 10022 -BackendPort 22
<!--NeedCopy-->
```

9. **Create a load balancer entity**

   Create the load balancer adding all objects (NAT rules, load balancer rules, probe configurations) together.

```
$lbName="ELB"

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
    $lbName -Location $locName -InboundNatRule $inboundNATRule1,
    $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -
    LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1
    -Probe $healthProbe
<!--NeedCopy-->
```

10. **Create a NIC**

    Create two NICs and associate each NIC with each VPX instance

    a) NIC1 with VPX1

    For example:

```
$nicName="NIC1"

$lbName="ELB"

$bePoolIndex=0

\* Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0

\* Frontend subnet index

```



```
15 $lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
      $rgName
16
17 $nic1=New-AzureRmNetworkInterface -Name $nicName -
      ResourceGroupName $rgName -Location $locName -Subnet $vnet.
      Subnets\[$subnetIndex\] -LoadBalancerBackendAddressPool $lb.
      BackendAddressPools\[$bePoolIndex\] -LoadBalancerInboundNatRule
       $lb.InboundNatRules\[$natRuleIndex\]
18 <!--NeedCopy-->
```

b) NIC2 with VPX2

For example:

```
1  $nicName="NIC2"
2
3  $lbName="ELB"
4
5  $bePoolIndex=0
6
7  $natRuleIndex=1
8
9  \* Second Inbound NAT (SSH) rule we need to use
10
11 `$subnetIndex=0
12
13 \* Frontend subnet index
14
15 $lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
      $rgName
16
17 $nic2=New-AzureRmNetworkInterface -Name $nicName -
      ResourceGroupName $rgName -Location $locName -Subnet $vnet.
      Subnets\[$subnetIndex\] -LoadBalancerBackendAddressPool $lb.
      BackendAddressPools\[$bePoolIndex\] -LoadBalancerInboundNatRule
        $lb.InboundNatRules\[$natRuleIndex\]
18 <!--NeedCopy-->
```

11. **Create Citrix ADC VPX instances**

    Create two Citrix ADC VPX instances as part of the same resource group and availability set, and attach it to the external load balancer.

    a) Citrix ADC VPX instance 1

    For example:


```
$vmName="VPX1"

$vmSize="Standard\_A3"

$pubName="citrix"

$offerName="netscalervpx110-6531"

$skuName="netscalerbyol"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
    $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
   AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used
    to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
   $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
   Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
   Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
   " + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
   $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
    -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
   $vm1
<!--NeedCopy-->
```


b) Citrix ADC VPX instance 2

For example:

```
$vmName="VPX2"

$vmSize="Standard\_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
    $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
   AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be
   used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
   $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
   Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
   Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
   " + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
   $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
    -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
   $vm2
<!--NeedCopy-->
```

12. **Configure the virtual machines**

    When both the Citrix ADC VPX instances start, then connect to both Citrix ADC VPX instances

using the SSH protocol to configure the virtual machines.

a) Active-Active: Run the same set of configuration commands on the command line of both the Citrix ADC VPX instances.

b) Active-Passive: Run this command on the command line of both the Citrix ADC VPX instances.

```
add ha node ##nodeID <nsip of other Citrix ADC VPX>
```

In Active-Passive mode, run configuration commands on the primary node only.

Provision a Citrix ADC VPX pair in a high availability setup with Azure internal load balancer

Log on to AzureRmAccount using your Azure user credentials.

1. **Create a resource group**

   The location specified here is the default location for resources in that resource group. Make sure all commands to create a load balancer use the same resource group.

   ```
   $rgName="\<resource group name\>"
   ```

   ```
   $locName="\<location name, such as West US\>"
   ```

   ```
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   For example:

   ```
   $rgName = "ARM-LB-NS"

   $locName = "West US"

   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. **Create a storage account**

   Choose a unique name for your storage account that contains only lowercase letters and numbers.

   ```
   $saName="<storage account name>"
   ```

   ```
   $saType="<storage account type>"
   ```

   , specify one: `Standard_LRS`, `Standard_GRS`, `Standard_RAGRS`, or `Premium_LRS`

   ```
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -
   Type $saType -Location $locName
   ```

   For example:


```
$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. **Create an availability set**

   A load balancer configured with an availability set ensures that your application is always available.

   ```
   $avName="<availability set name>"
   ```

   ```
   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -
   Location $locName
   ```

4. **Create a virtual network**

   Add a new virtual network with at least one subnet, if the subnet was not created previously.

   ```
   $vnetName = "LBVnet"

   $vnetAddressPrefix="10.0.0.0/16"

   $FrontendAddressPrefix="10.0.1.0/24"

   $BackendAddressPrefix="10.0.2.0/24"

   $vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
       $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
      Subnet $frontendSubnet,$backendSubnet\`

   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
      frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
      backendSubnet -AddressPrefix $BackendAddressPrefix
   <!--NeedCopy-->
   ```

   Note: Choose the AddressPrefix parameter value as per your requirement.

   Assign front end and back end subnet to the virtual network that you created earlier in this step.

   If the front end subnet is the first element of array VNet, subnetId must be $vnet.Subnets[0].Id.

If the front end subnet is the second element in the array, the subnetId must be $vnet.Subnets[1].Id, and so on.

5. **Create a backend address pool**

```
$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name "
LB-backend"
```

6. **Create NAT rules**

Create NAT rules for services that you are not load balancing.

```
1 $inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
     Name "Inboundnatrule1" -FrontendIpConfiguration $frontendIP -
     Protocol TCP -FrontendPort 3441 -BackendPort 3389
2
3 $inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
     Name "RDP2" -FrontendIpConfiguration $frontendIP -Protocol TCP
     -FrontendPort 3442 -BackendPort 3389
4 <!--NeedCopy-->
```

Use front end and back end ports as per your requirement.

7. **Create a health probe**

Create a TCP health probe with port 9000 and interval 5 seconds.

```
1 $healthProbe = New-AzureRmLoadBalancerProbeConfig -Name "
     HealthProbe"  " -Protocol tcp -Port 9000 -IntervalInSeconds 5 -
     ProbeCount 2
2 <!--NeedCopy-->
```

8. **Create a load balancing rule**

Create an LB rule for each service that you are load balancing.

For example:

You can use the following example to load balance HTTP service.

```
1 $lbrule = New-AzureRmLoadBalancerRuleConfig -Name "lbrule1" -
     FrontendIpConfiguration $frontendIP -BackendAddressPool
     $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort
     80 -BackendPort 80
2 <!--NeedCopy-->
```

Use front end and back end ports as per your requirement.

9. **Create a load balancer entity**

Create the load balancer adding all objects (NAT rules, load balancer rules, probe configurations) together.

```
$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name
    "InternalLB" -Location $locName -FrontendIpConfiguration
    $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -
    LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -
    Probe $healthProbe
<!--NeedCopy-->
```

10. **Create a NIC**

    Create two NICs and associate each NIC with each Citrix ADC VPX instance

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress
    10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools\[0\] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules\[0\]
<!--NeedCopy-->
```

    This NIC is for Citrix ADC VPX 1. The Private IP must be in same subnet as that of subnet added.

```
$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic2-be -Location $locName -PrivateIpAddress
    10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools\[0\] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules\[1\].
<!--NeedCopy-->
```

    This NIC is for Citrix ADC VPX 2.The parameter `Private IPAddress` can have any private IP as per your requirement.

11. **Create Citrix ADC VPX instances**

    Create two VPX instances part of the same resource group and availability set, and attach it to the internal load balancer.

    a) Citrix ADC VPX instance 1

    For example:

```
$vmName="VPX1"

$vmSize="Standard\_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName
```



```
6
7  $vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avset.Id
8
9  $cred=Get-Credential -Message "Type Credentials which will be used
       to login to VPX instance"
10
11 $vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
       $vmName -Credential $cred -Verbose
12
13 $vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
       Offer $offerName -Skus $skuName -Version "latest"
14
15 $vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id
16
17 $diskName="dynamic"
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
       Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
       " + $diskName  + ".vhd"
22
23 $vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
       $osDiskUri1 -CreateOption fromImage
24
25 Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
       -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
       $vm1
28 <!--NeedCopy-->
```

b) Citrix ADC VPX instance 2

For example:

```
1  $vmName="VPX2"
2
3  $vmSize="Standard\_A3"
4
5  $avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
       $rgName
6
7  $vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
```



```
    AvailabilitySetId $avset.Id
8
9  $cred=Get-Credential -Message " Type Credentials which will be
    used to login to VPX instance "
10
11 $vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
    $vmName -Credential $cred -Verbose
12
13 $vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"
14
15 $vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id
16
17 $diskName="dynamic"
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
    " + $diskName  + ".vhd"
22
23 $vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage
24
25 Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
     -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm2
28 <!--NeedCopy-->
```

12. **Configure the virtual machines**

    When both the Citrix ADC VPX instances start, then connect to both Citrix ADC VPX instances using the SSH protocol to configure the virtual machines.

    a) Active-Active: Run the same set of configuration commands on the command line of both the Citrix ADC VPX instances.

    b) Active-Passive: Run this command on the command line of both the Citrix ADC VPX instances.

    ```
    add ha node ##nodeID <nsip of other Citrix ADC VPX>
    ```

    In Active-Passive mode, run configuration commands on the primary node only.

Azure FAQs

September 14, 2021

- **Is the upgrade procedure of Citrix ADC VPX instance installed from Azure Marketplace different from the on-premises upgrade procedure?**

  No. You can upgrade your Citrix ADC VPX instance in the Microsoft Azure cloud to Citrix ADC VPX release 11.1 or later, using standard Citrix ADC VPX upgrade procedures. You can upgrade either using GUI or CLI procedures. For any new installations, use the Citrix ADC VPX image for Microsoft Azure cloud.

  To download the Citrix ADC VPX upgrade builds, go to **Citrix Downloads** > **Citrix ADC Firmware**.

- **How to correct MAC moves and interface mutes observed on Citrix ADC VPX instances hosted on Azure?**

  In Azure Multi-NIC environment, by default, all data interfaces might show MAC moves and interface mutes. To avoid MAC moves and interface mutes on Azure environments, Citrix recommends you to create a VLAN per data interface (without tag) of the ADC VPX instance and bind the primary IP of the NIC in Azure.

  For more information, see CTX224626 article.

Deploy a Citrix ADC VPX instance on the Google Cloud Platform

September 14, 2021

You can deploy a Citrix ADC VPX instance on the Google Cloud Platform (GCP). A VPX instance in GCP enables you to take advantage of GCP cloud computing capabilities and use Citrix load balancing and traffic management features for your business needs. You can deploy VPX instances in GCP as standalone instances. Both single NIC and multi NIC configurations are supported.

Supported features

A VPX instance running in GCP supports the following features:

- Load Balancing
- ICA Proxy
- Content Switching
- Authentication, authorization, and auditing
- Rewrite

- Responder
- RDP Proxy
- nFactor
- LDAP
- VPN (CVPN/Full)
- GSLB

Limitation

- IPv6 isn't supported.

Hardware requirements

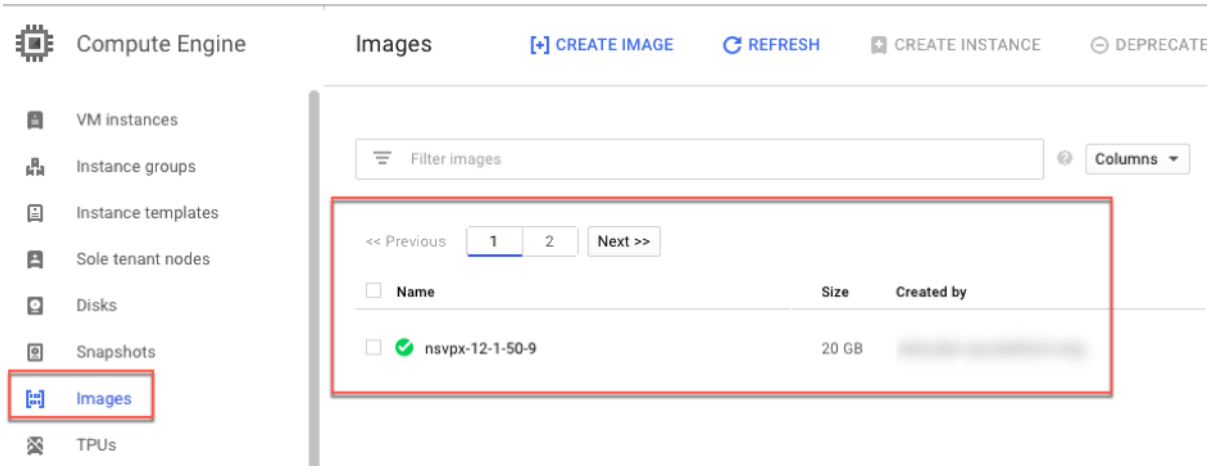
VPX instance in GCP must have minimum of 2 vCPUs and 4 GB RAM.

Prerequisites

1. Install the "gcloud" utility on your device. You can find the utility at this link: https://cloud.google.com/sdk/install
2. Download the NSVPX-GCP image from the Citrix download site.
3. Upload the file(for example, NSVPX-GCP-12.1-50.9_nc_64.tar.gz) to a storage bucket on Google by following the steps given at https://cloud.google.com/storage/docs/uploading-objects.
4. Run the following command on the gcloud utility to create an image.

```
gcloud compute images create <IMAGE_NAME> --source-uri=gs://<
    STORAGE_BUCKET_NAME>/<FILE_NAME>.tar.gz --guest-os-features=
    MULTI_IP_SUBNET
<!--NeedCopy-->
```

It might take a moment for the image to be created. After the image is created, it appears under **Compute** > **Compute Engine** in the GCP console.



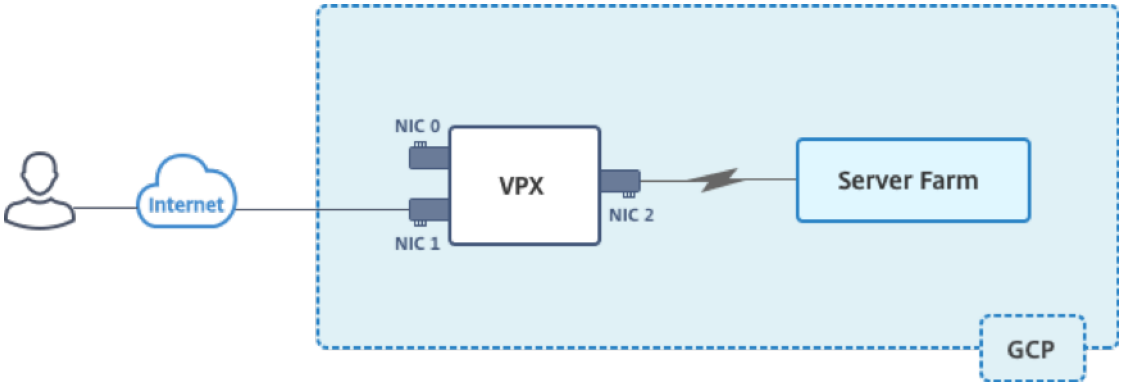
Points to note

Consider the following GCP-specific points before you begin your deployment.

- After creating the instance, you cannot add or remove any network interfaces.
- For a multi-NIC deployment, create separate VPC networks for each NIC. One NIC can be associated with only one network.
- For a single-NIC instance, the GCP console creates a network by default.
- Minimum 4 vCPUs are required for an instance with more than two network interfaces.
- If IP forwarding is required, you must enable IP forwarding while creating the instance and configuring the NIC.

Scenario: Deploy a multi-NIC, multi-IP standalone VPX instance

This scenario illustrates how to deploy a Citrix VPX standalone instance in GCP. In this scenario, you create a standalone VPX instance with multiple NICs. The instance communicates with back-end servers (the server farm).



Create three NICs to serve the following purposes.

NIC	Purpose	Associated with VPC network
NIC 0	Serves management traffic (Citrix ADC IP)	Management network
NIC 1	Serves client-side traffic (VIP)	Client network
NIC 2	Communicates with back-end servers (SNIP)	Back-end server network

Also, set up the required communication routes between the instance and the back-end servers, and between the instance and the external hosts on the public internet.

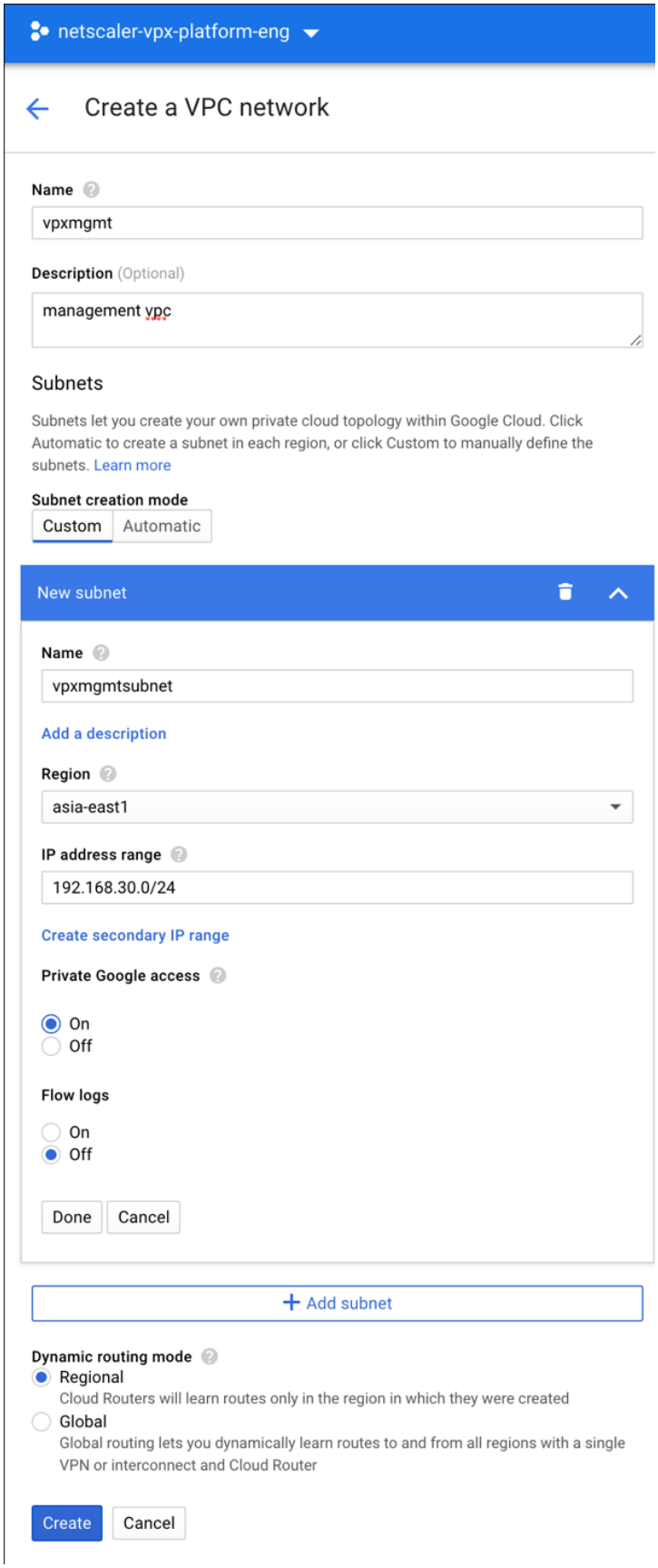
Summary of deployment steps

1. Create three VPC networks for three different NICs.
2. Create firewall rules for ports 22, 80, and 443
3. Create an instance with three NICs

Note: Create an instance in the same region where you've created the VPC networks.

Step 1. Create VPC networks.

Create three VPC networks that is associated with management NIC, client NIC, and server NIC. To create a VPC network, log on to **Google console > Networking > VPC network > Create VPC Network**. Complete the required fields, as shown in the screen capture, and click **Create**.

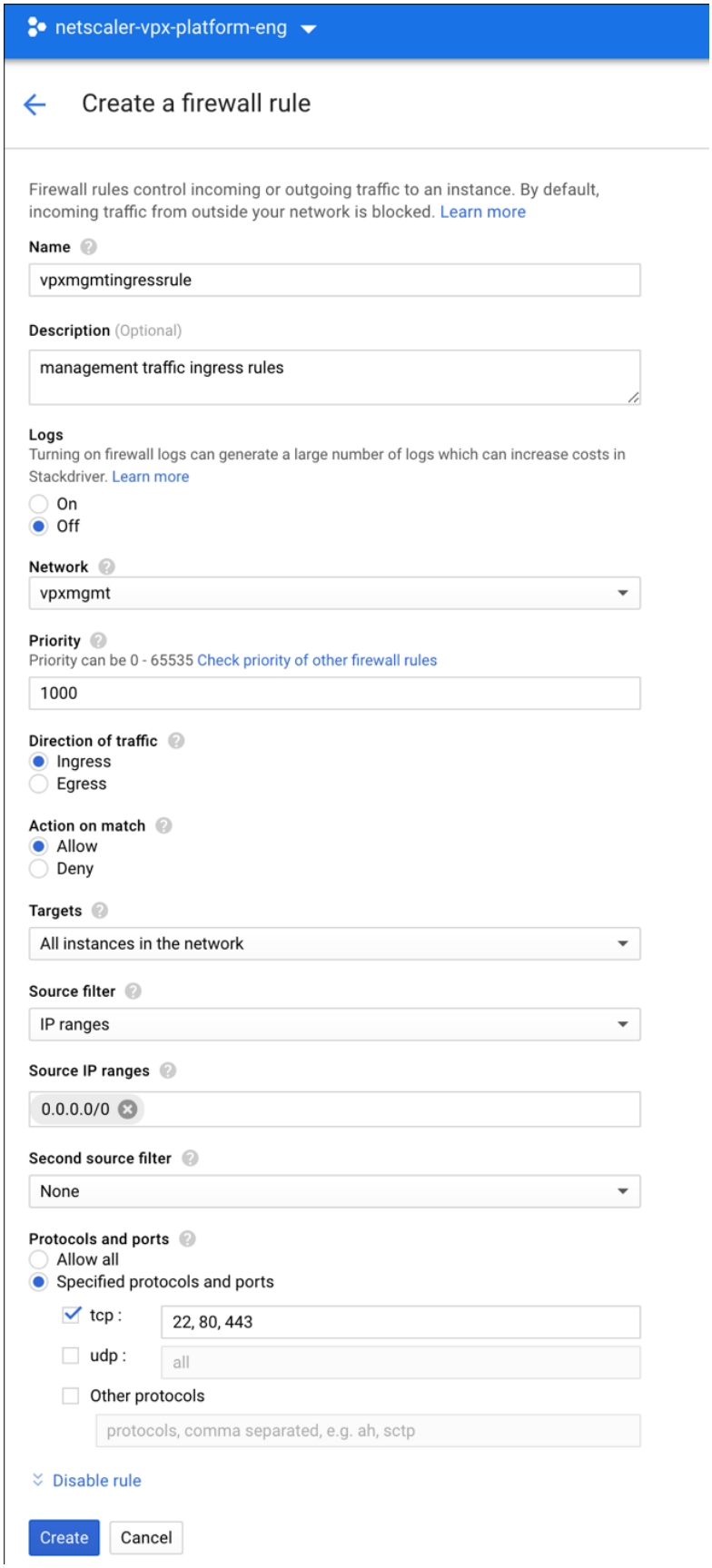


Similarly, create VPC networks for client and server-side NICs.

Note: All three VPC networks must be in the same region, which is asia-east1 in this scenario.

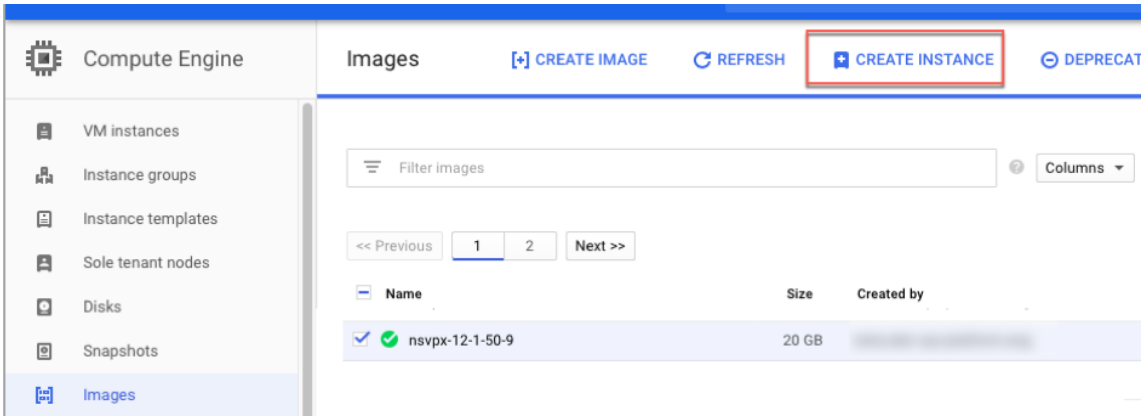
Step 2. Create firewall rules for ports 22, 80, and 443.

Create rules for SSH (port 22), HTTP (port 80), and HTTPS (port 443) for each VPC networks. For more information about firewall rules, see Firewall Rules Overview.



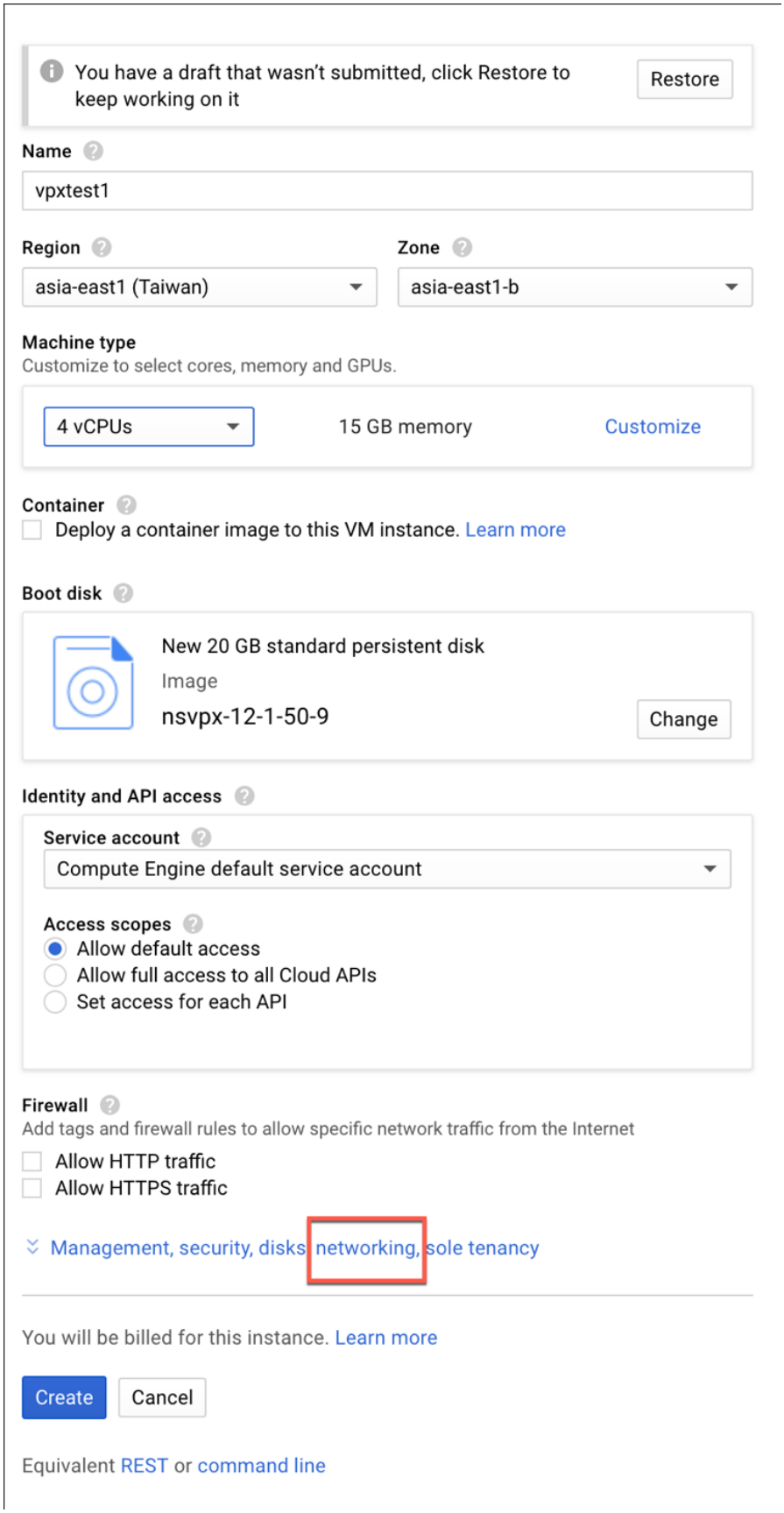
Step 3. Create the VPX instance.

1. Log on to the GCP console.
2. Under **Compute**, hover over Compute Engine, and select **Images**.
3. Select the image, and click **Create Instance**.

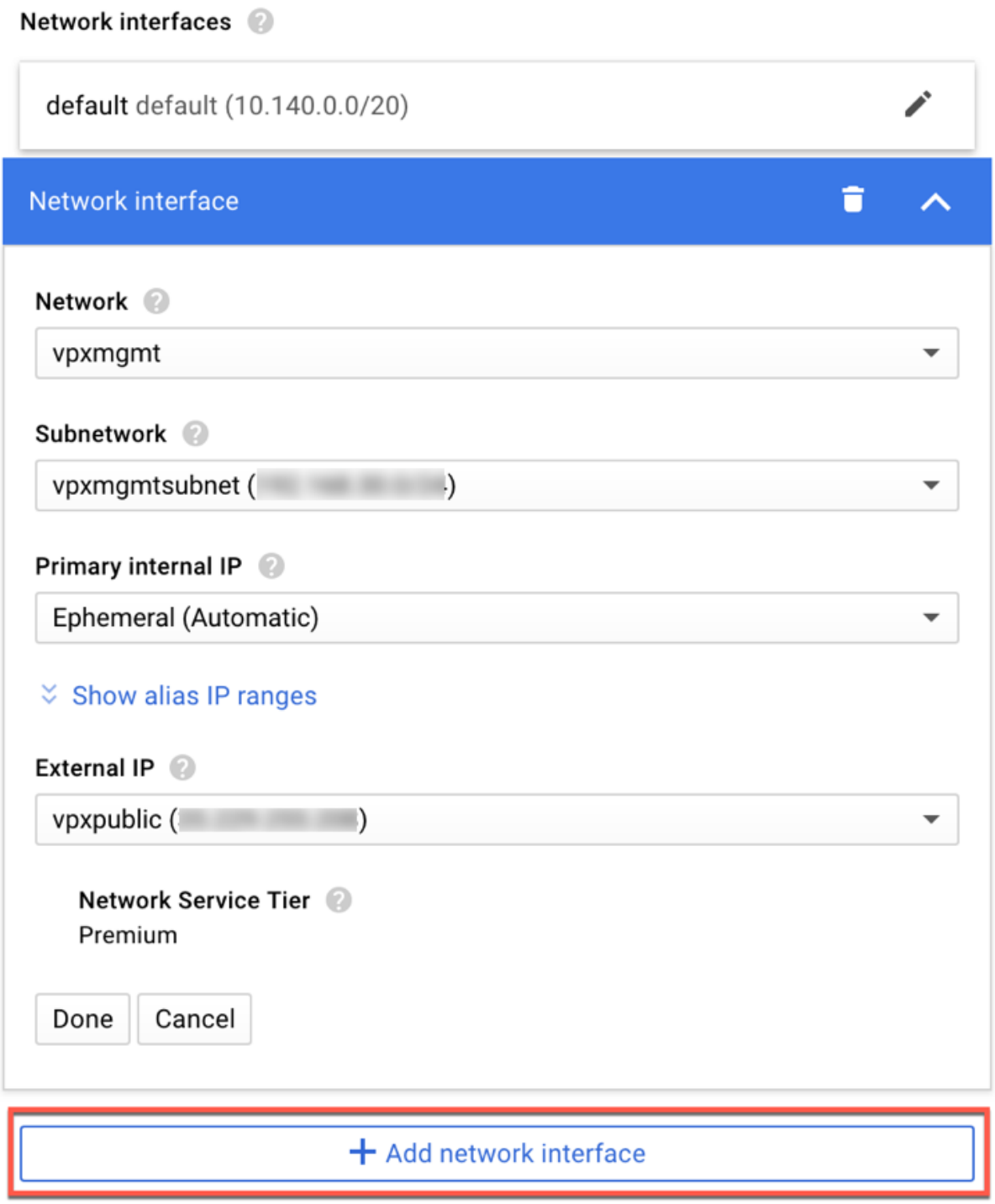


4. Select an instance with 4 vCPUs, to support multiple NICs.
5. Click the networking option from Management, security, disks, networking, sole tenancy to add the additional NICs.

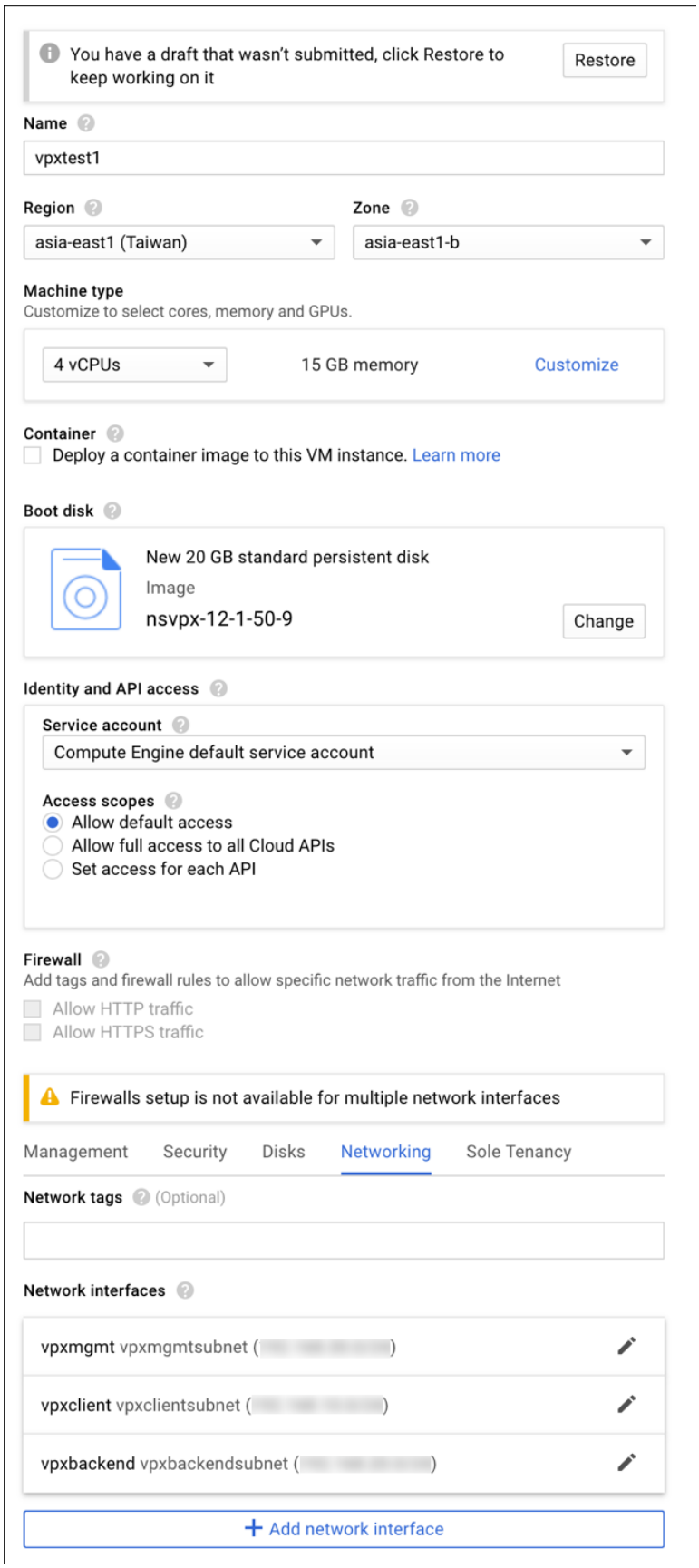
   Note: Container image is not supported on VPX instances on GCP.



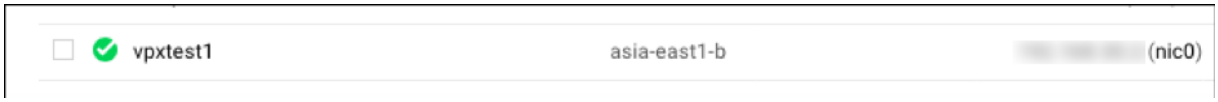
6. Under **Networking interfaces**, click the edit icon to edit the default NIC. This NIC is the management NIC.
7. In the **Network interfaces** window, under **Network**, select the VPC network you created for the management NIC.
8. For the management NIC, create a static external IP address. Under the External IP list, click **Create IP address**.
9. In the **Reserve a new static IP address** window, add a name and description and click **Reserve**.
10. Click **Add network interface** to create NICs for a client and server-side traffic.



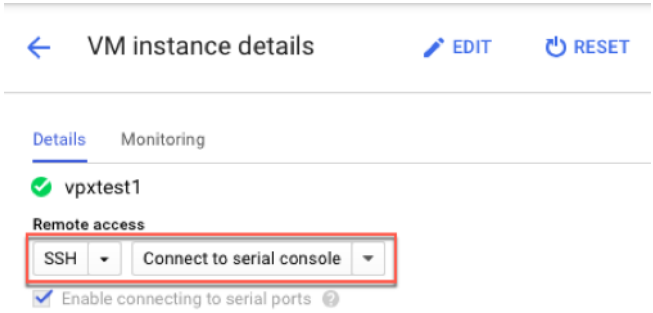
After you've created all the NICs, click **Create** to create the VPX instance.



The instance appears under **VM instances**.

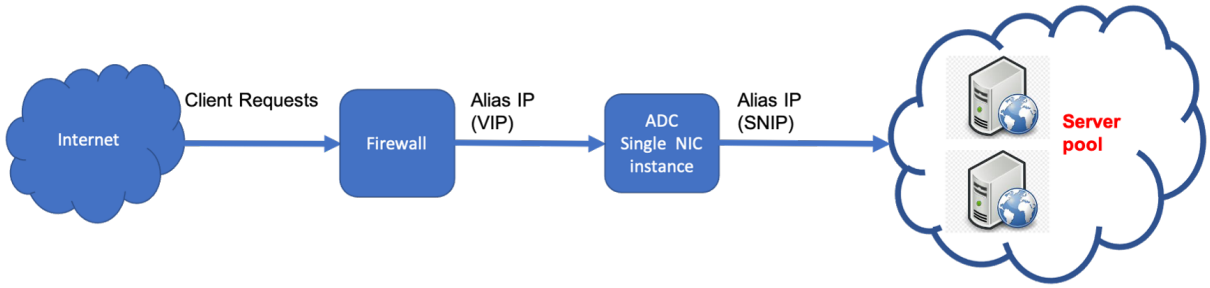


Use the GCP SSH or the serial console to configure and manage the VPX instance.



Scenario: Deploy a single-NIC, standalone VPX instance

This scenario illustrates how to deploy a Citrix VPX standalone instance with a single NIC in GCP. The alias IP addresses are used to achieve this deployment.



Create a single NIC (NIC0) to serve the following purposes:

- Handle management traffic (Citrix ADC IP) in the management network.
- Handle client-side traffic (VIP) in the client network.
- Communicate with back-end servers (SNIP) in the back-end server network.

Set up the required communication routes between the following:

- Instance and the back-end servers.
- Instance and the external hosts on the public internet.

Summary of deployment steps

1. Create a VPC network for NIC0.
2. Create firewall rules for ports 22, 80, and 443.

3. Create an instance with a single NIC.
4. Add Alias IP addresses to VPX.
5. Add VIP and SNIP on VPX.
6. Add a load balancing virtual server.
7. Add a service or service group on the instance.
8. Bind the service or service group to the load balancing virtual server on the instance.

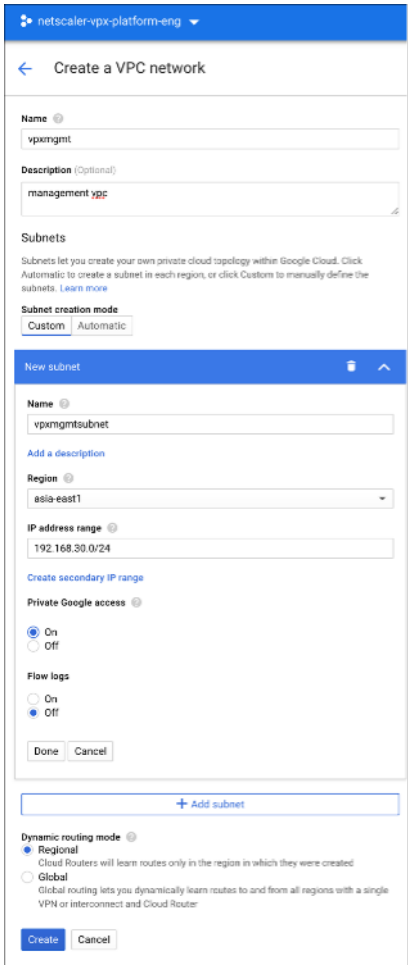
> **Note**:
>
> Create an instance in the same region where you've created the VPC networks.

Step 1. Create one VPC network.

Create one VPC network to associate with NIC0.

To create a VPC network, do these steps:

1. Log on to **GCP console > Networking > VPC network > Create VPC Network**
2. Complete the required fields, and click **Create**.



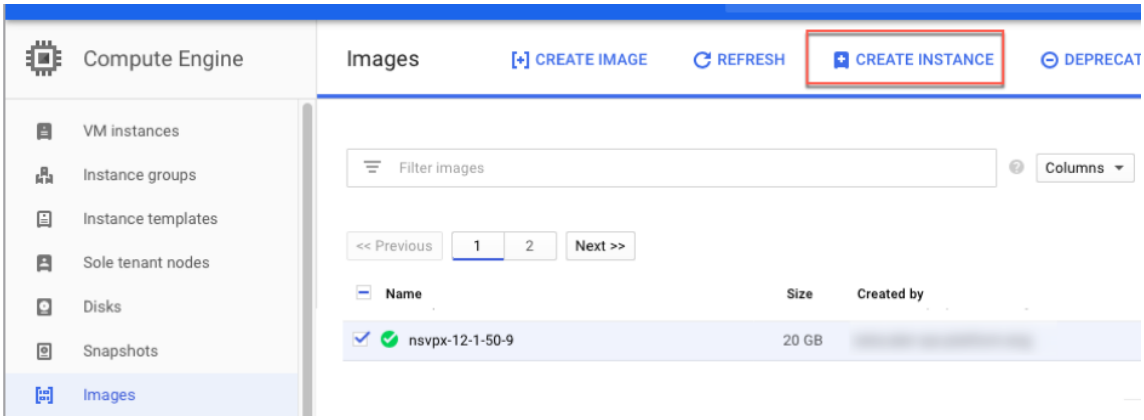
Step 2. Create firewall rules for ports 22, 80, and 443.

Create rules for SSH (port 22), HTTP (port 80), and HTTPS (port 443) for the VPC network. For more information about firewall rules, see Firewall Rules Overview.

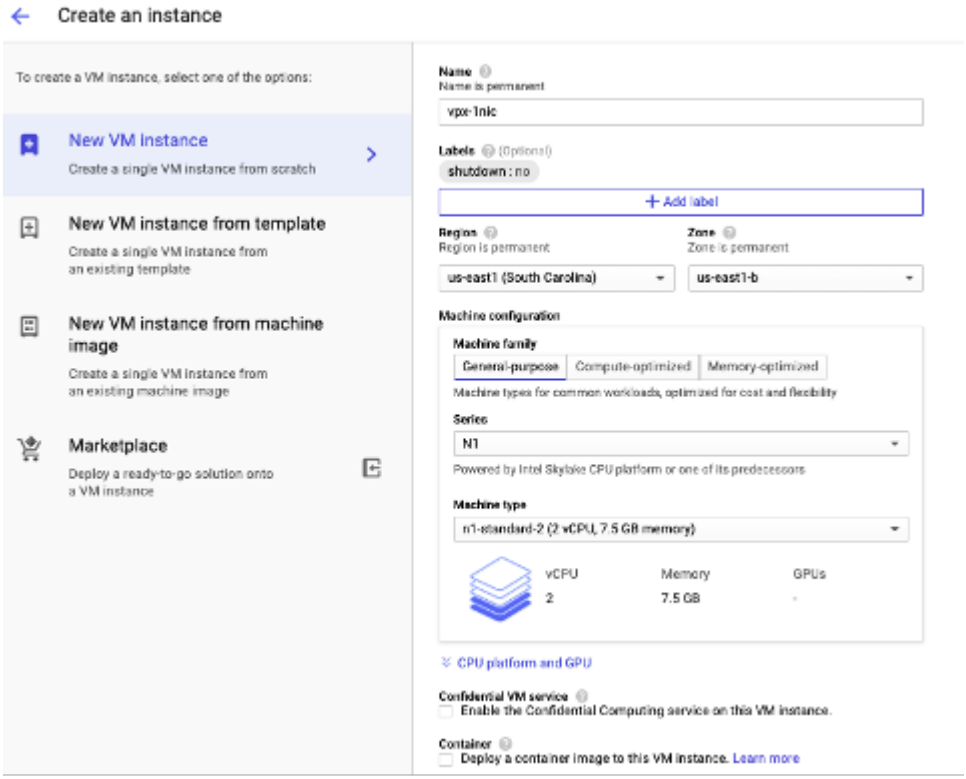
Step 3. Create an instance with single NIC.

To create an instance with single NIC, do these steps:

1. Log on to the **GCP console**.
2. Under **Compute**, hover over **Compute Engine**, and select **Images**.
3. Select the image, and click **Create Instance**.



4. Select an instance type with two vCPUs (minimum requirement for ADC).



5. Click the **Networking** tab from the **Management, security, disks, networking** window.
6. Under **Network interfaces**, click the **Edit** icon to edit the default NIC.
7. In the **Network interfaces** window, under **Network**, select the VPC network that you created.
8. You can create a static external IP address. Under the **External IP addresses**, click **Create IP address**.
9. In the **Reserve a static address** window, add a name and description and click **Reserve**.
10. Click **Create** to create the VPX instance.
    The new instance appears under VM instances.

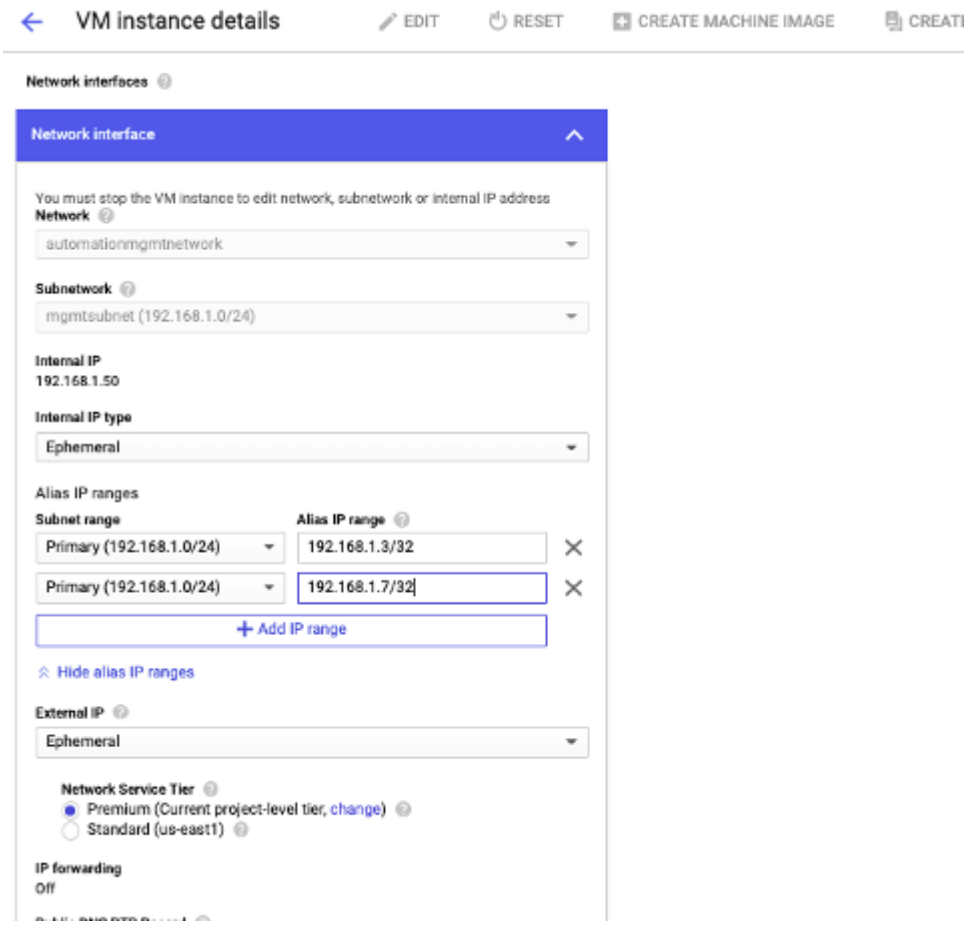
Step 4. Add alias IP addresses to the VPX instance.

Assign two alias IP addresses to the VPX instance to use as VIP and SNIP addresses.

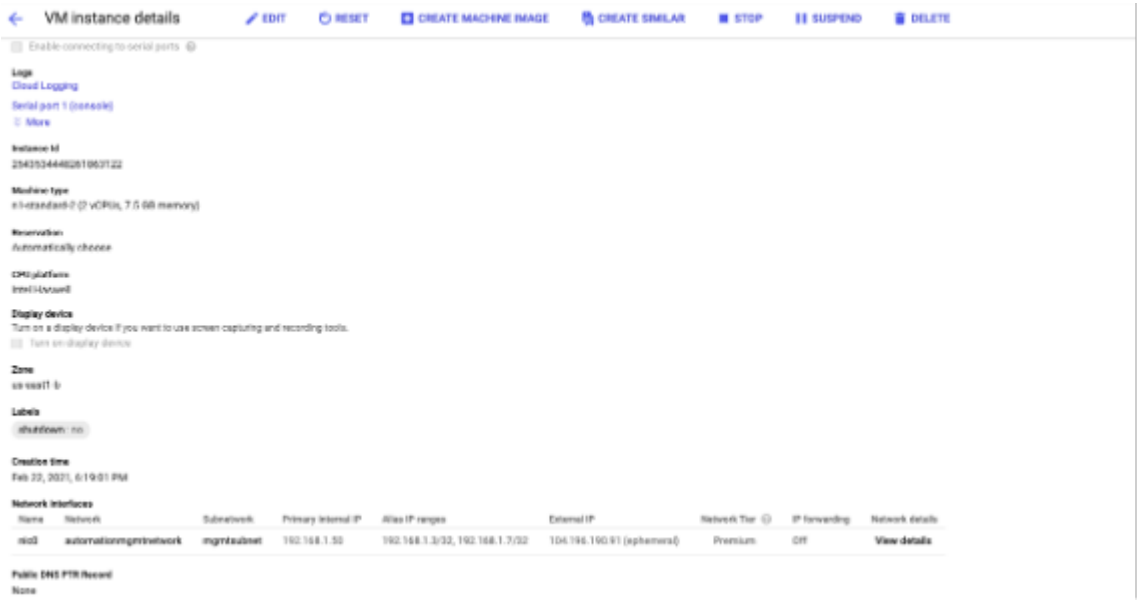
> **Note:**
>
> Do not use the primary internal IP address of the VPX instance to configure the VIP or SNIP.

To create an alias IP address, perform these steps:

1. Navigate to the VM instance and click **Edit**.
2. In the **Network interface** window, edit the NIC0 interface.
3. In the **Alias IP range** field, enter the alias IP addresses.



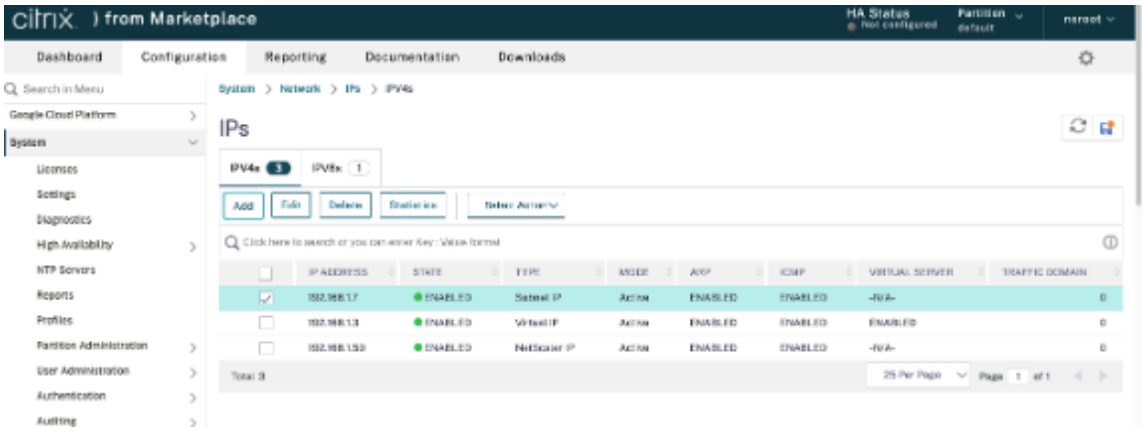
4. Click **Done**, and then **Save**.
5. Verify the alias IP addresses in the **VM instance details** page.



Step 5. Add VIP and SNIP on the VPX instance.

On the VPX instance, add client alias IP address and server alias IP address.

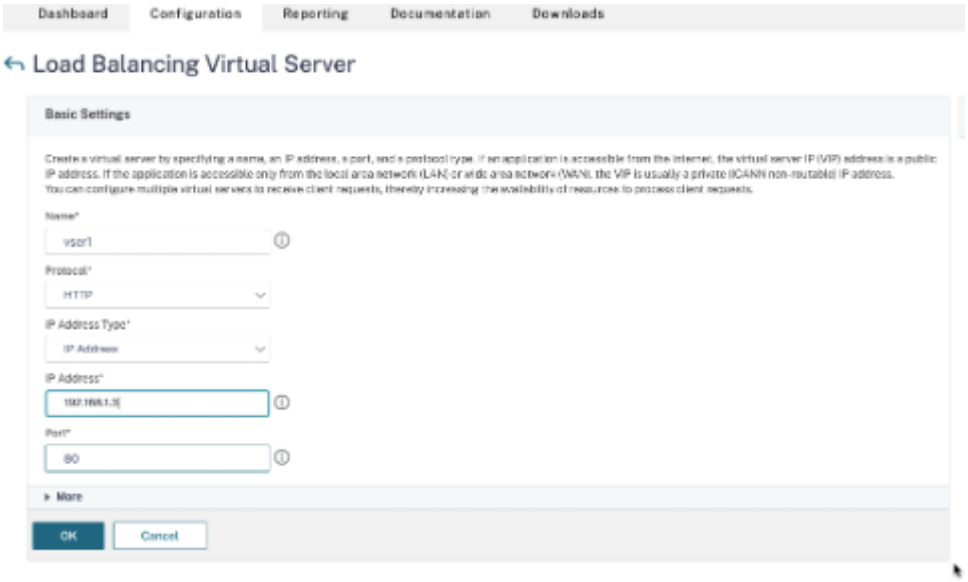
1. On the Citrix ADC GUI, navigate to **System > Network > IPs > IPv4s**, and click **Add**.



2. To create a client alias IP (VIP) address:
   - Enter the client-alias IP address and netmask configured for the VPC subnet in the VM instance.
   - In the **IP Type** field, select **Virtual IP** from the drop-down menu.
   - Click **Create**.
3. To create a server alias IP (SNIP) address:
   - Enter the server-alias IP address and netmask configured for the VPC subnet in the VM instance.
   - In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   - Click **Create**.

Step 6. Add load balancing virtual server.

1. On the Citrix ADC GUI, navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**, and click **Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (client alias IP), and Port.
3. Click **OK** to create the load balancing virtual server.



Step 7. Add a service or service group on the VPX instance.

1. From the Citrix ADC GUI, navigate to **Configuration > Traffic Management > Load Balancing > Services**, and click **Add**.
2. Add the required values for Service Name, IP Address, Protocol, and Port, and click **OK**.

Step 8. Bind the service/service group to the Load Balancing Virtual Server on the instance.

1. From the GUI, navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 6**, and click **Edit**.
3. In the **Service and Service Groups** window, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in **Step 7**, and click **Bind**.

Points to note after you've deployed the VPX instance on GCP

- Log on to the VPX with user name `nsroot` and instance ID as password. At the prompt, change the password and save the configuration.
- For collecting a technical support bundle, run the command `shell /netscaler/showtech_cloud.pl` instead of the customary `show techsupport`.

- After deleting a Citrix ADC VM from GCP console, delete the associated Citrix ADC internal target instance also. To do so, go to gcloud CLI and type the following command:

```
gcloud compute -q  target-instances delete <instance-name>-
   adcinternal --zone <zone>
<!--NeedCopy-->
```

> Note: `<instance-name>-adcinternal` is the name of the target instance that must be deleted.

Citrix ADC VPX licensing

A Citrix ADC VPX instance on GCP requires a license. The following licensing options are available for Citrix ADC VPX instances running on GCP.

- **Subscription-based licensing**: Citrix ADC VPX appliances are available as paid instances on the GCP marketplace. Subscription-based licensing is a pay-as-you-go option. Users are charged hourly. The following VPX models and license editions are available on the GCP marketplace.

VPX model	License editions
VPX10	Standard, Advanced, Premium

- **Bring your own license (BYOL)**: If you bring your own license (BYOL), see the VPX Licensing Guide at http://support.citrix.com/article/CTX122426. You have to:
  - Use the licensing portal within the Citrix website to generate a valid license.
  - Upload the license to the instance.
- **Citrix ADC VPX Check-In/Check-Out licensing**: For more information, see Citrix ADC VPX Check-In/Check-Out Licensing.

VPX Express for on-premises and cloud deployments does not require a license file. For more information on Citrix ADC VPX Express see the "Citrix ADC VPX Express license" section in Citrix ADC licensing overview.

GDM templates to deploy a Citrix ADC VPX instance

You can use a Citrix ADC VPX Google Deployment Manager (GDM) template to deploy a VPX instance on GCP. For details, see Citrix ADC GDM Templates.

Citrix ADC marketplace images

You can use the images in GDM templates to bring up the Citrix ADC appliance.

The following table lists the images that are available on GCP marketplace.

Release	Image name	Image location
13.0	citrix-adc-vpx-1000-advanced-13-0-61-48	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-0-61-48
13.0	citrix-adc-vpx-1000-advanced-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-0-latest
13.0	citrix-adc-vpx-1000-premium-13-0-61-48	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-0-61-48
13.0	citrix-adc-vpx-1000-premium-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-0-latest
13.0	citrix-adc-vpx-1000-standard-13-0-61-48	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-0-61-48
13.0	citrix-adc-vpx-1000-standard-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-0-latest
13.0	citrix-adc-vpx-5000-enterprise-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-0-58-32
13.0	citrix-adc-vpx-5000-enterprise-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-0-latest

Release	Image name	Image location
13.0	citrix-adc-vpx-5000-platinum-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-0-58-32
13.0	citrix-adc-vpx-5000-platinum-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-0-latest
13.0	citrix-adc-vpx-5000-standard-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-0–58-32
13.0	citrix-adc-vpx-5000-standard-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-0-latest
13.0	citrix-adc-vpx-3000-enterprise-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-0-58-32
13.0	citrix-adc-vpx-3000-enterprise-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-0-latest
13.0	citrix-adc-vpx-3000-platinum-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-0-58-32
13.0	citrix-adc-vpx-3000-platinum-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-0-latest
13.0	citrix-adc-vpx-3000-standard-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-0–58-32

Release	Image name	Image location
13.0	citrix-adc-vpx-3000-standard-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-0-latest
13.0	citrix-adc-vpx-200-enterprise-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-0-58-32
13.0	citrix-adc-vpx-200-enterprise-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-0-latest
13.0	citrix-adc-vpx-200-platinum-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-0-58-32
13.0	citrix-adc-vpx-200-platinum-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-0-latest
13.0	citrix-adc-vpx-200-standard-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-0–58-32
13.0	citrix-adc-vpx-200-standard-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-0-latest
13.0	citrix-adc-vpx-10-enterprise-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-0-58-32
13.0	citrix-adc-vpx-10-enterprise-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-0-latest

Release	Image name	Image location
13.0	citrix-adc-vpx-10-platinum-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-0-58-32
13.0	citrix-adc-vpx-10-platinum-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-0-latest
13.0	citrix-adc-vpx-10-standard-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-10-standard-13-0-58-32
13.0	citrix-adc-vpx-10-standard-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-10-standard-13-0-latest
13.0	citrix-adc-vpx-express-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-0-58-32
13.0	citrix-adc-vpx-express-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-0-latest
13.0	citrix-adc-vpx-byol-13-0-58-32	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-0-58-32
13.0	citrix-adc-vpx-byol-13-0-latest	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-0-latest

Resources

- Creating Instances with Multiple Network Interfaces
- Creating and Starting a VM Instance

Related information

- Deploy a VPX high-availability pair on Google Cloud Platform

Deploy a VPX high-availability pair on Google Cloud Platform

September 14, 2021

You can configure two Citrix ADC VPX instances on Google Cloud Platform (GCP) as a high availability (HA) active-passive pair. When you configure one instance as the primary node and the other as the secondary node, the primary node accepts connections and manages servers. The secondary node monitors the primary. If for any reason, if the primary node is unable to accept connections, the secondary node takes over.

For more information on HA, see High Availability.

The nodes must be in the same region; however, they can be either in the same zone or different zones. For more information, see Regions and Zones.

Each VPX instance requires at least three IP subnets (Google VPC networks):

- A management subnet
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP, MIP, and so on)

Citrix recommends three network interfaces for a standard VPX instance.

You can deploy a VPX high-availability pair in the following methods:

- Using external static IP address
- Using private IP address

GDM templates to deploy a VPX high-availability pair on GCP

You can use a Citrix ADC Google Deployment Manager (GDM) template to deploy a VPX high-availability pair on GCP. For details, see Citrix ADC GDM Templates.

Forwarding rules support for VPX high-availability pair on GCP

You can deploy a VPX high-availability pair on the GCP using forwarding rules.

For more information on forwarding rules, see Forwarding rules overview.

Prerequisites

- Forwarding rules must be in the same region as the VPX instances.
- Target instances must be in the same zone as the VPX instance.
- Number of target instances for both primary and secondary nodes must match.

Example:

You have a high-availability pair in the `us-east1` region with primary VPX in `us-east1-b` zone and secondary VPX in `us-east1-c` zone. A forwarding rule is configured for the primary VPX with the target instance in `us-east1-b` zone. Configure a target instance for secondary VPX in `us-east1-c` zone to update the forwarding rule on failover.

Limitations

Only forwarding rules that are configured with target instances at the back end are supported in VPX high-availability deployment.

Deploy a VPX high-availability pair with external static IP address on the Google Cloud Platform

September 14, 2021

You can deploy a VPX high-availability pair on GCP using an external static IP address. The client IP address of the primary node must be bound to an external static IP address. Upon failover, the external static IP address is moved to the secondary node for traffic to resume.

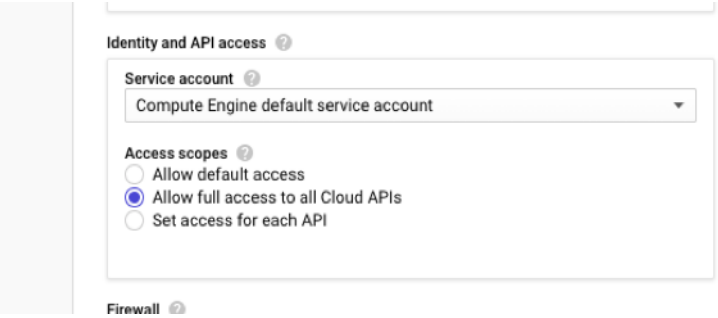
A static external IP address is an external IP address that is reserved for your project until you decide to release it. If you use an IP address to access a service, you can reserve that IP address so that only your project can use it. For more information, see Reserving a Static External IP Address.

For more information on HA, see High Availability.

Before you start

- Read the Limitation, Hardware requirements, Points to note mentioned in Deploy a Citrix ADC VPX instance on Google Cloud Platform. This information applies to HA deployments also.
- Enable **Cloud Resource Manager API** for your GCP project.

- Allow full access to all Cloud APIs while creating the instances.



- Ensure that the IAM role associated with your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [

"compute.addresses.use",
 “compute.forwardingRules.list” ,
 “compute.forwardingRules.setTarget” ,
 “compute.instances.setMetadata”
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.instances.get",
"Compute.instances.list",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
 “compute.targetInstances.list” ,
 “compute.targetInstances.use” ,
"compute.zones.list",
]
<!--NeedCopy-->
```

- If you have configured alias IP addresses on an interface other than the management interface, ensure that your GCP service account has the following extra IAM permissions:

```
"compute.instances.updateNetworkInterface"
<!--NeedCopy-->
```

- If you have configured GCP forwarding rules on the primary node, read the limitations and requirements mentioned in Forwarding rules support for VPX high-availability pair on GCP to update them to new primary on failover.

How to deploy a VPX HA pair on Google Cloud Platform

Here’s a summary of the HA deployment steps:

1. Create VPC networks in the same region. For example, Asia-east.
2. Create two VPX instances (primary and secondary nodes) on the same region. They can be in the same zone or different zones. For example Asia east-1a and Asia east-Ib.
3. Configure HA settings on both instances by using the Citrix ADC GUI or ADC CLI commands.

Step 1. Create VPC networks

Create VPC networks based on your requirements. Citrix recommends you to create three VPC networks for associating with management NIC, client NIC, and server NIC.

To create a VPC network, perform these steps:

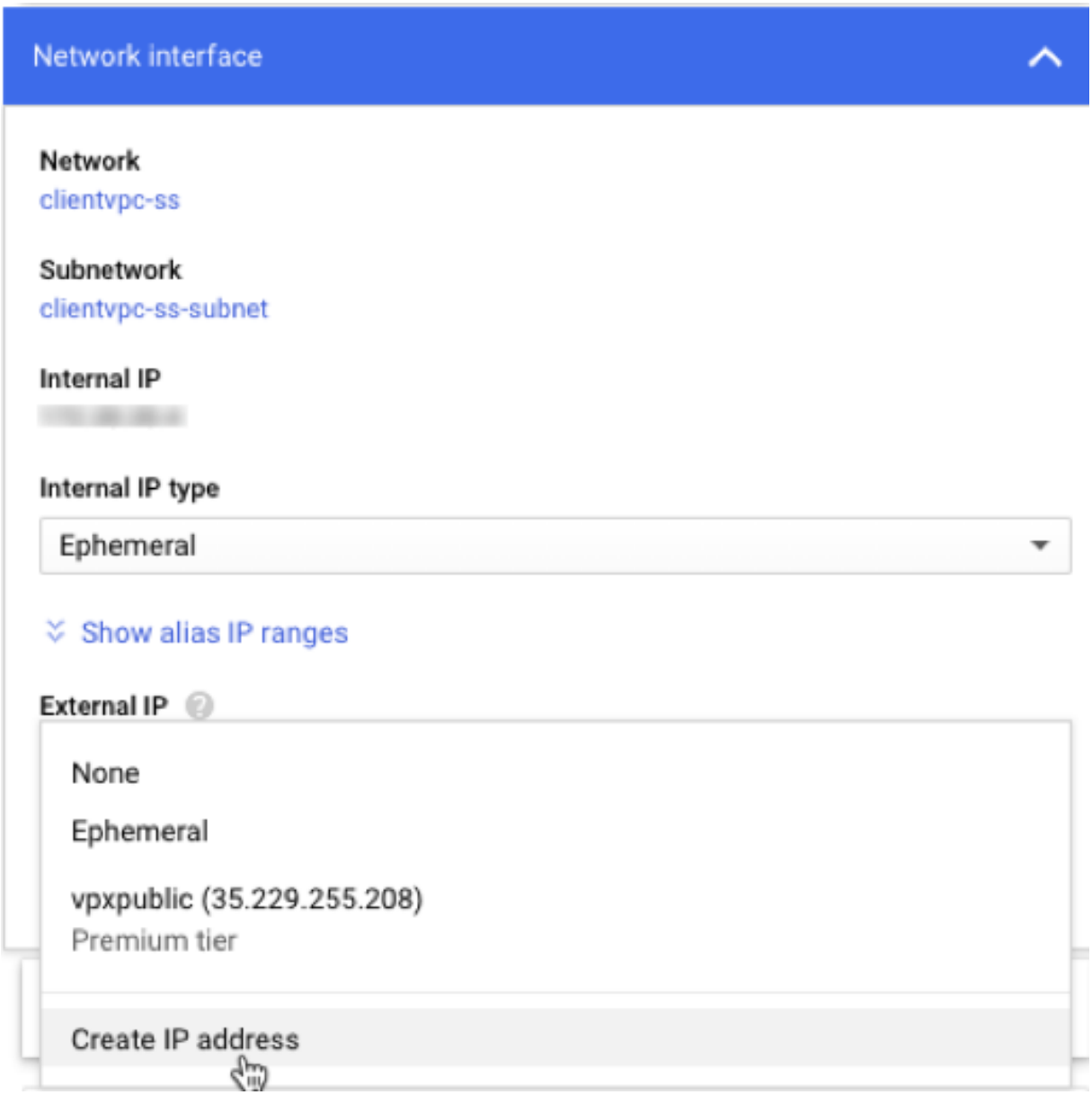
1. Log on the **Google console > Networking > VPC network > Create VPC Network**.
2. Complete the required fields, and click **Create**.

For more information, see the **Create VPC Networks** section in Deploy a Citrix ADC VPX instance on Google Cloud Platform.

Step 2. Create two VPX instances

Create two VPX instances by following the steps given in Scenario: deploy a multi-NIC, multi-IP standalone VPX instance.

> **Important**
>
> Assign a static external IP address to client IP address (VIP) of the primary node. You can use an existing reserved IP address or create a new one. To create a static external IP address, navigate to **Network interface > External IP**, click **Create IP address**.



After the failover, when the old primary becomes the new secondary, the static external IP address moves from the old primary and is attached to the new primary. For more information, see the Google cloud document Reserving a Static External IP Address.

After you've configured the VPX instances, you can configure the VIP and SNIP addresses. For more information, see Configuring Citrix ADC-owned IP addresses.

Step 3. Configure high availability

After you've created the instances on Google Cloud Platform, you can configure HA by using the Citrix ADC GUI for CLI.

Configure HA by using the GUI

Step 1. Set up high availability in INC mode on both the instances.

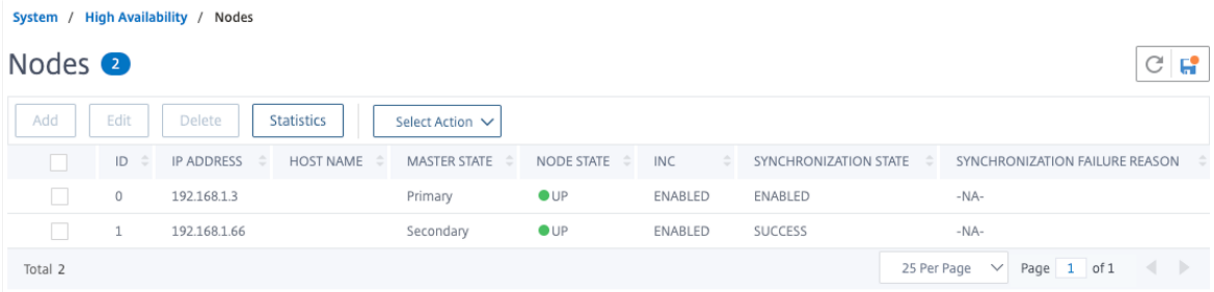
On the **primary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the secondary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

On the **secondary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the primary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

Before you proceed further, ensure that the Synchronization state of the secondary node is shown as **SUCCESS** in the **Nodes** page.



> **Note**
>
> Now, the secondary node has the same log-on credentials as the primary node.

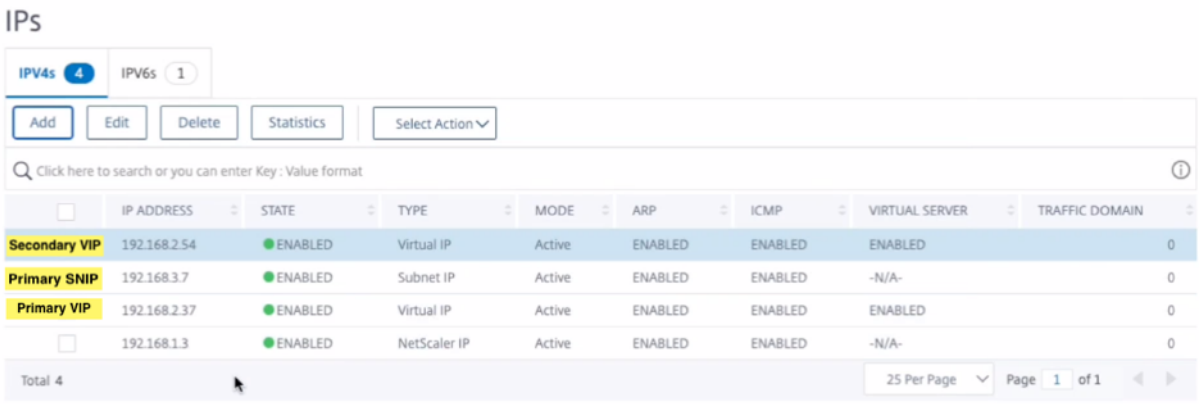
Step 2. Add Virtual IP address and Subnet IP address on both the nodes.

On the **primary node**, perform the following steps:

1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. Add a primary VIP address by following these steps:
   a) Enter the internal IP address of the client-facing interface of the primary instance and netmask configured for the client subnet in the VM instance.

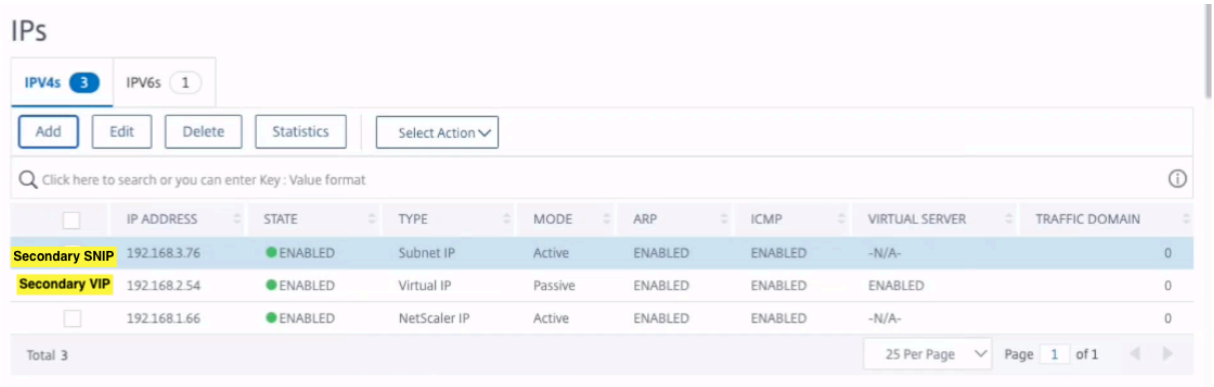
b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
c) Click **Create**.

3. Add a primary SNIP address by following these steps:
   a) Enter the internal IP address of the server-facing interface of the primary instance and netmask configured for the server subnet in the primary instance.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.
4. Add a secondary VIP address by following these steps:
   a) Enter the internal IP address of the client-facing interface of the secondary instance and netmask configured for the client subnet in the VM instance.
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
   c) Click **Create**.



On the **secondary node**, perform the following steps:

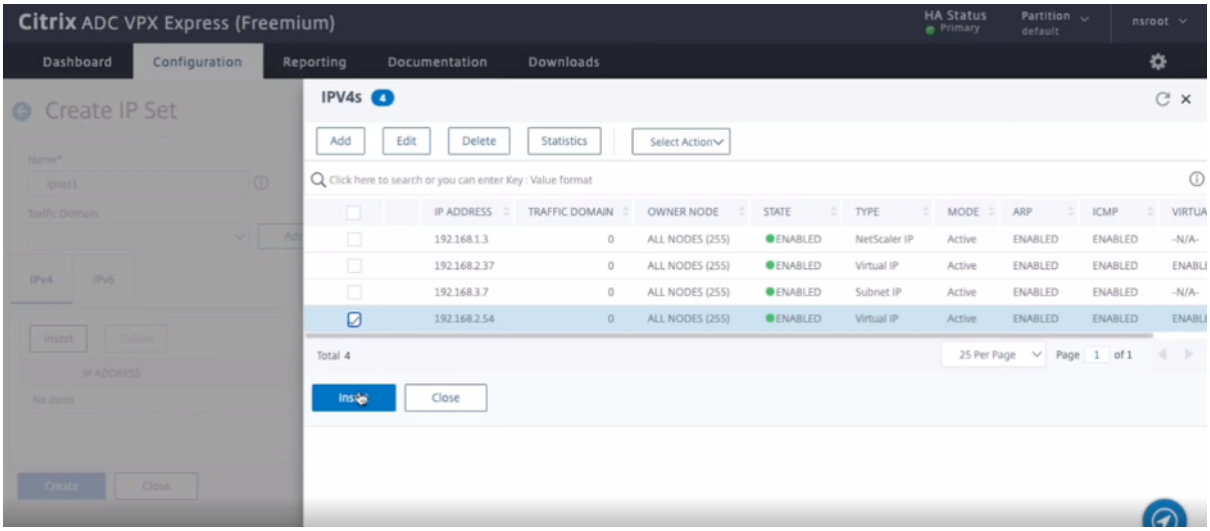
1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. Add a secondary VIP address by following these steps:
   a) Enter the internal IP address of the client-facing interface of the secondary instance and netmask configured for the client subnet in the VM instance.
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
3. Add a secondary SNIP address by following these steps:
   a) Enter the internal IP address of the server-facing interface of the secondary instance and netmask configured for the server subnet in the secondary instance.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.



Step 3. Add IP set and bind IP set to the secondary VIP on both the instances.

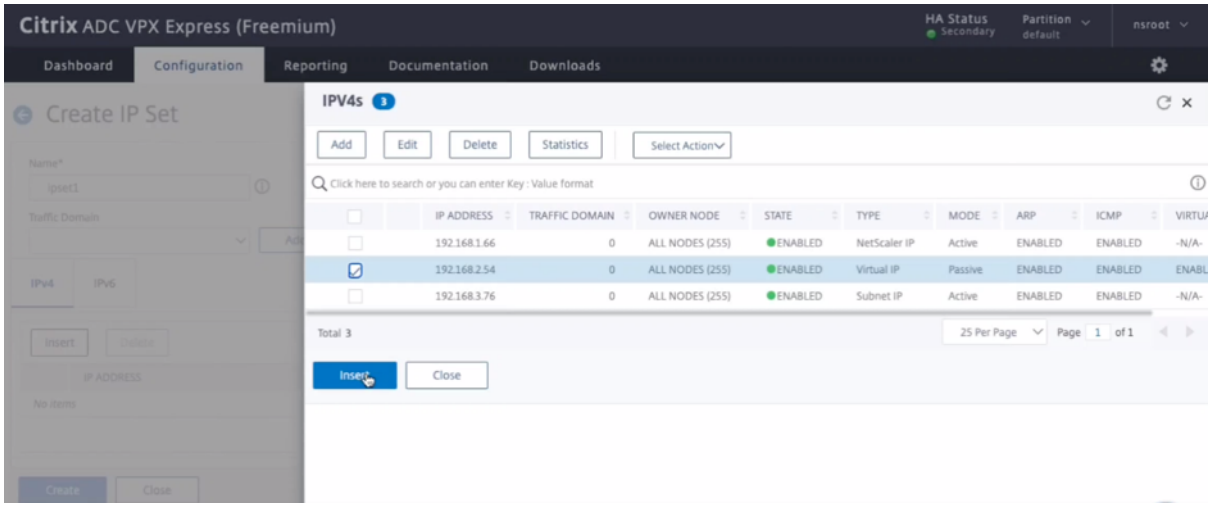
On the **primary node**, perform the following steps:

1. Navigate to **System > Network > IP Sets > Add**.
2. Add an IP set name and click **Insert**.
3. From the **IPV4s** page, select the virtual IP (secondary VIP) and click **Insert**.
4. Click **Create** to create the IP set.



On the **secondary node**, perform the following steps:

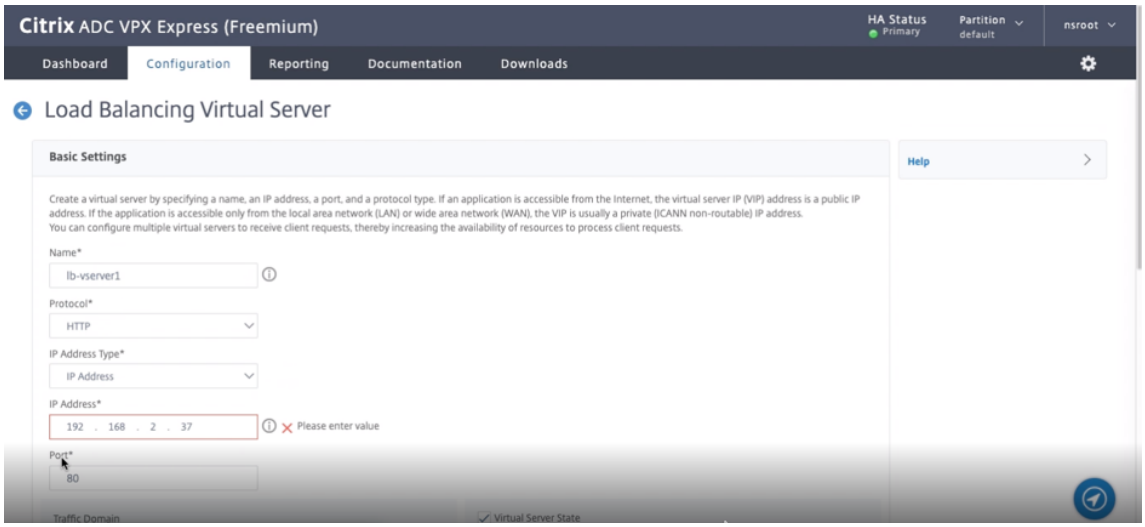
1. Navigate to **System > Network > IP Sets > Add**.
2. Add an IP set name and click **Insert**.
3. From the **IPV4s** page, select the virtual IP (secondary VIP) and click **Insert**.
4. Click **Create** to create the IP set.



> **Note**
>
> IP set name must be same on both the instances.

Step 4. Add a load balancing virtual server on the primary instance.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.
2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP address (primary VIP), and Port.



3. Click **More**. Navigate to **IP Range IP Set Settings**, select **IPset** from the drop-down menu, and provide the IPset created in **Step 3**.
4. Click **OK** to create the load balancing virtual server.

Step 5. Add a service or service group on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

Step 6. Bind the service or service group to the load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 4**, and click **Edit**.
3. In the **Service and Service Groups** tab, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in the **Step 5**, and click **Bind**.

Save the configuration. After a forced failover, the secondary becomes the new primary. The external static IP of the old primary VIP moves to the new secondary VIP.

Configure high availability using CLI

Step 1. Set up high availability in INC mode in both the instances.

On the primary node, type the following command.

```
1 add ha node 1 <sec_ip> -inc ENABLED
2 <!--NeedCopy-->
```

On the secondary node, type the following command.

```
1 add ha node 1 <prim_ip> -inc ENABLED
2 <!--NeedCopy-->
```

`sec_ip` refers to the internal IP address of the management NIC of the secondary node.

`prim_ip` refers to the internal IP address of the management NIC of the primary node.

Step 2. Add Virtual and Subnet IPs on both the nodes.

On the primary node, type the following command.

```
1 add ns ip <primary_vip> <subnet> -type VIP
2
3 add ns ip <secondary_vip> <subnet> -type VIP
4
5 add ns ip <primary_snip> <subnet> -type SNIP
6 <!--NeedCopy-->
```

`primary_vip` refers to the internal IP address of the client-facing interface of the primary instance.

`secondary_vip` refers to the internal IP address of the client-facing interface of the secondary instance.

`primary_snip` refers to the internal IP address of the server-facing interface of the primary instance.

On the secondary node, type the following command.


```
1 add ns ip <secondary_vip> <subnet> -type VIP
2
3 add ns ip <secondary_snip> <subnet> -type SNIP
4 <!--NeedCopy-->
```

`secondary_vip` refers to the internal IP address of the client-facing interface of the secondary instance.

`secondary_snip` refers to the internal IP address of the server-facing interface of the secondary instance.

Step 3. Add IP set and bind IP set to secondary VIP on both the instances.

On the primary node, type the following command:

```
1 add ipset <ipsetname>
2 bind ipset <ipsetname> <secondary VIP>
3 <!--NeedCopy-->
```

On the secondary node, type the following command:

```
1 add ipset <ipsetname>
2 bind ipset <ipsetname> <secondary VIP>
3 <!--NeedCopy-->
```

> **Note**
>
> IP set name must be same on both the instances.

Step 4. Add a virtual server on the primary instance.

Type the following command:

```
1 add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
     > -ipset <ipset_name>
2 <!--NeedCopy-->
```

Step 5. Add a service or service group on the primary instance.

Type the following command:

```
1 add service <service_name> <service_ip_address> <protocol> <port>
2 <!--NeedCopy-->
```

Step 6. Bind the service/service group to the load balancing virtual server on the primary instance.

Type the following command:


```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> *Note*:
>
> To save your configuration, type the command `save config`. Otherwise, the configurations are lost after you restart the instances.

Step 7. Verify the configuration.

Ensure that the external IP address attached to the primary client NIC moves to the secondary on a failover.

1. Make a cURL request to the external IP address and make sure that it is reachable.
2. On the primary instance, perform failover:

   From GUI, navigate to **Configuration > System > High Availability > Action > Force Failover**.

   From CLI, type the following command:

   ```
   force ha failover -f
   <!--NeedCopy-->
   ```

   On the GCP console, goto the Secondary instance. The external IP address must have moved to the client NIC of secondary after failover.
3. Issue a cURL request to the external IP and ensure it is reachable again.

Deploy a VPX high-availability pair with private IP address on Google Cloud Platform

September 14, 2021

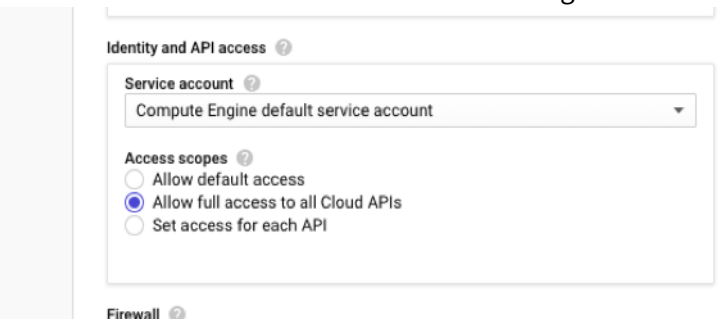
You can deploy a VPX high-availability pair on GCP using private IP address. The client IP (VIP) must be configured as alias IP address on the primary node. Upon failover, the Client IP address is moved to the secondary node, for the traffic to resume.

For more information on high availability, see High Availability.

Before you start

- Read the Limitation, Hardware requirements, Points to note mentioned in Deploy a Citrix ADC VPX instance on Google Cloud Platform. This information applies to high availability deployments also.

- Enable **Cloud Resource Manager API** for your GCP project.
- Allow full access to all Cloud APIs while creating the instances.



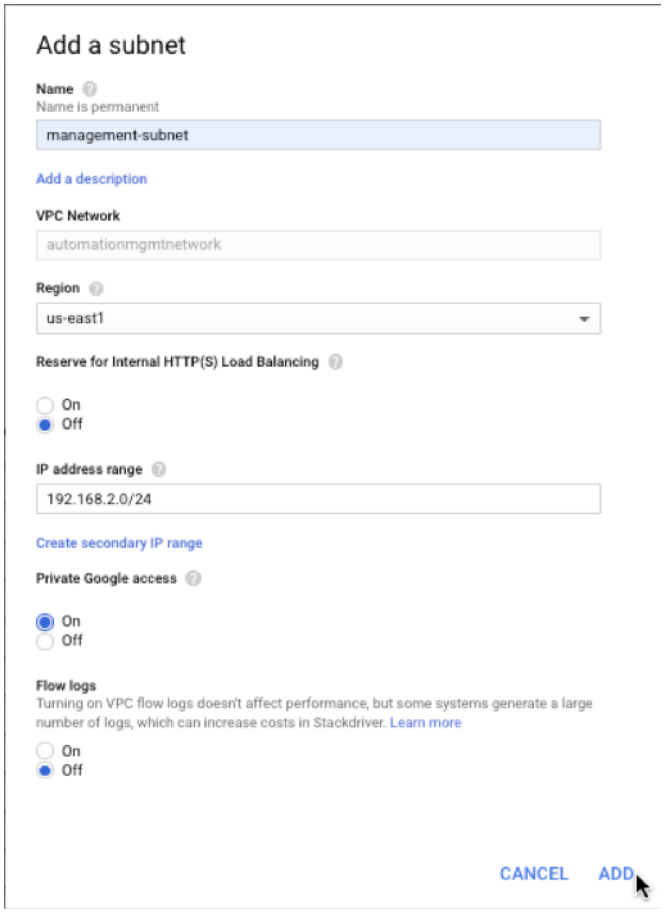
- Ensure that your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [
 “compute.forwardingRules.list” ,
 “compute.forwardingRules.setTarget” ,
 “compute.instances.setMetadata” ,
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
 “compute.targetInstances.list” ,
 “compute.targetInstances.use” ,
"compute.zones.list",
]
<!--NeedCopy-->
```

- If you have configured external IP addresses on an interface other than the management interface, ensure that your GCP service account has the following additional IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.addresses.use"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
]
<!--NeedCopy-->
```

- If your VMs do not have internet access, you must enable **Private Google Access** on the management subnet.



- If you have configured GCP forwarding rules on the primary node, read the limitations and requirements mentioned in Forwarding rules support for VPX high-availability pair on GCP to update them to new primary on failover.

How to deploy a VPX high availability pair on Google Cloud Platform

Here is a summary of the high availability deployment steps:

1. Create VPC networks in the same region. For example, Asia-east.
2. Create two VPX instances (primary and secondary nodes) on the same region. They can be in the same zone or different zones. For example Asia east-1a and Asia east-Ib.
3. Configure high availability settings on both instances by using the Citrix ADC GUI or ADC CLI commands.

Step 1. Create VPC networks

Create VPC networks based on your requirements. Citrix recommends you to create three VPC networks for associating with management NIC, client NIC, and server NIC.

To create a VPC network, perform these steps:

1. Log on the **Google console > Networking > VPC network > Create VPC Network**.
2. Complete the required fields, and click **Create**.

For more information, see the **Create VPC Networks** section in Deploy a Citrix ADC VPX instance on Google Cloud Platform.

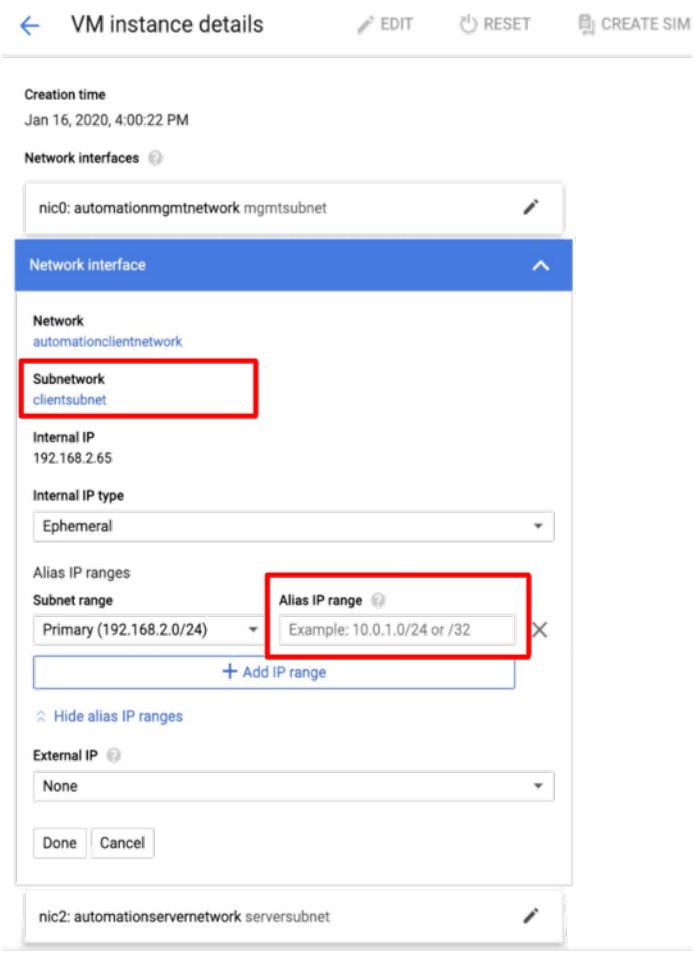
Step 2. Create two VPX instances

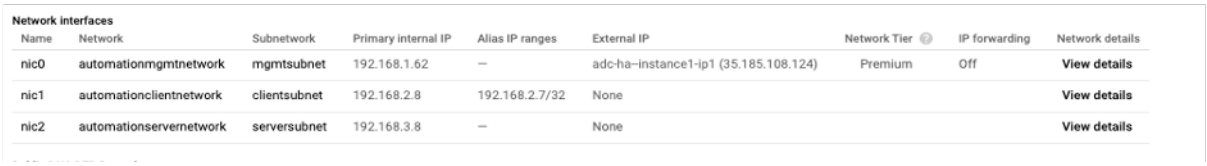
Create two VPX instances by following the steps given in Scenario: deploy a multi-NIC, multi-IP standalone VPX instance.

> **Important:**
>
> Assign a client alias IP address to the primary node. Do not use the internal IP address of the VPX instance to configure the VIP.

To create a client alias IP address, perform these steps:

1. Navigate to the VM instance and click **Edit**.
2. In the **Network Interface** window, edit the client interface.
3. In the **Alias IP range** field, enter the client alias IP address.





After the failover, when the old primary becomes the new secondary, the alias IP addresses move from the old primary and is attached to the new primary.

After you have configured the VPX instances, you can configure the Virtual (VIP) and Subnet IP (SNIP) addresses. For more information, see Configuring Citrix ADC-owned IP addresses.

Step 3. Configure high availability

After you've created the instances on Google Cloud Platform, you can configure high availability by using the Citrix ADC GUI or CLI.

Configure high availability by using the GUI

Step 1. Set up high availability in INC Enabled mode on both the nodes.

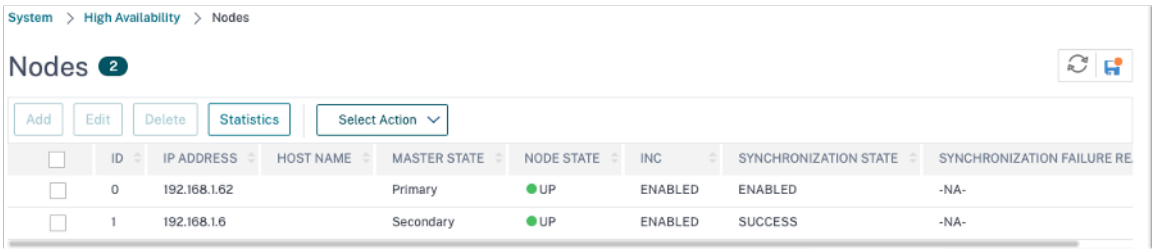
On the **primary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the secondary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

On the **secondary node**, perform the following steps:

1. Log on to the instance with user name `nsroot` and instance ID of the node from GCP console as the password.
2. Navigate to **Configuration > System > High Availability > Nodes**, and click **Add**.
3. In the **Remote Node IP address** field, enter the private IP address of the management NIC of the primary node.
4. Select the **Turn on INC (Independent Network Configuration) mode on self node** check box.
5. Click **Create**.

Before you proceed further, ensure that the Synchronization state of the secondary node is shown as **SUCCESS** in the **Nodes** page.



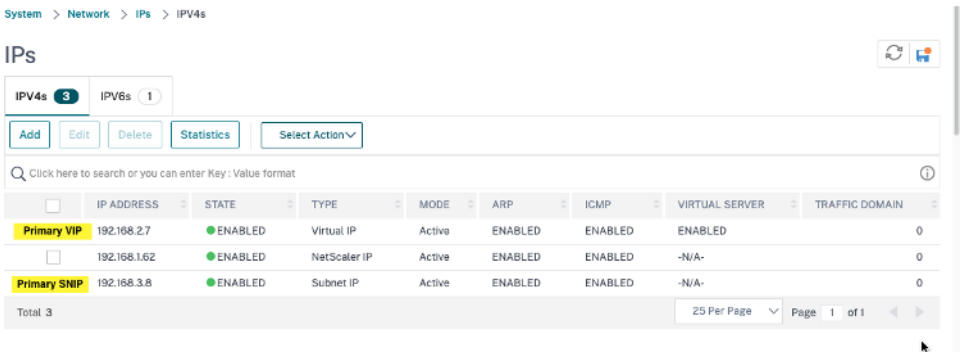
> **Note**
>
> Now, the secondary node has the same log-on credentials as the primary node.

Step 2. Add Virtual IP address and Subnet IP address on both the nodes.

On the primary node, perform the following steps:

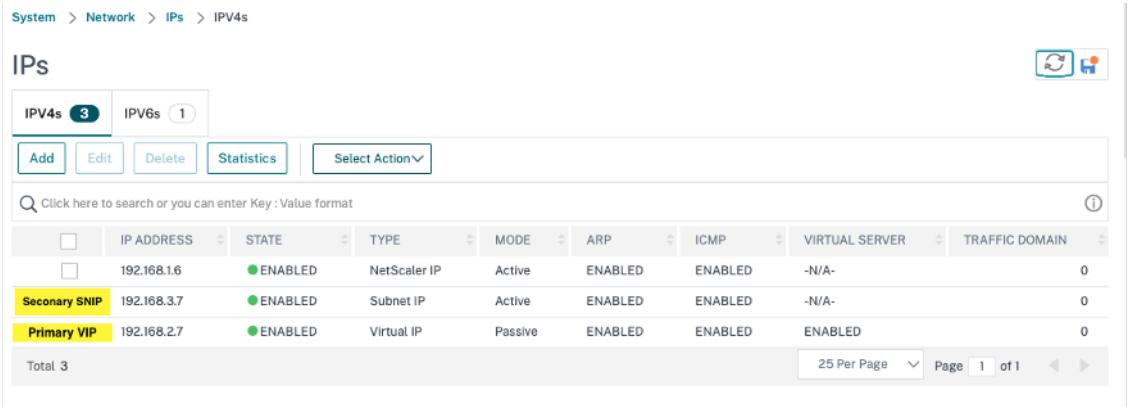
1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. To create a client alias IP (VIP) address:
   a) Enter the Alias IP address and netmask configured for the client subnet in the VM instance.
   b) In the **IP Type** field, select **Virtual IP** from the drop-down menu.
   c) Click **Create**.
3. To create a server IP (SNIP) address:

a) Enter the internal IP address of the server-facing interface of the primary instance and netmask configured for the server subnet.
b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
c) Click **Create**.



On the secondary node, perform the following steps:

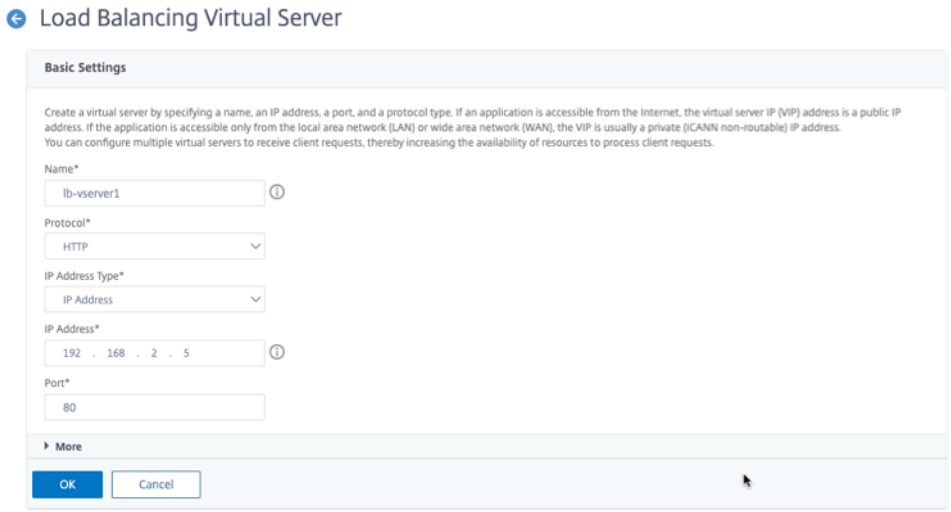
1. Navigate to **System > Network > IPs > IPv4s**, and click **Add**.
2. To create a client alias IP (VIP) address:
   a) Enter the Alias IP address and netmask configured for the client subnet on the primary VM instance.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.
3. To create a server IP (SNIP) address:
   a) Enter the internal IP address of the server-facing interface of the secondary instance and netmask configured for the server subnet.
   b) In the **IP Type** field, select **Subnet IP** from the drop-down menu.
   c) Click **Create**.



Step 3. Add a load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.

2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (primary client alias IP address) and Port, and click **OK**.



Step 4. Add a service or service group on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

Step 5. Bind the service or service group to the load balancing virtual server on the primary node.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 3**, and click **Edit**.
3. In the **Service and Service Groups** tab, click **No Load Balancing Virtual Server Service Binding**.
4. Select the service configured in the **Step 4**, and click **Bind**.

Step 5. Save the configuration.

After a forced failover, the secondary becomes the new primary. The client alias IP (VIP) and the server alias IP (SNIP) from the old primary moves to the new primary.

Configure high availability by using the CLI

Step 1. Set up high availability in **INC Disabled** mode in both the instances by using the Citrix ADC CLI.

On the primary node, type the following command.

```
1 add ha node 1 <sec_ip> -inc ENABLED
2 <!--NeedCopy-->
```

On the secondary node, type the following command.


```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

The `sec_ip` refers to the internal IP address of the management NIC of the secondary node.

The `prim_ip` refers to the internal IP address of the management NIC of the primary node.

Step 2. Add VIP and SNIP on both nodes.

Type the following commands on the primary node:

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP

<!--NeedCopy-->
```

> **Note:**
>
> Enter the Alias IP address and netmask configured for the client subnet in the VM instance.

```
add ns ip <primary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

The `primary_snip` refers to the internal IP address of the server-facing interface of the primary instance.

Type the following commands on the secondary node:

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
<!--NeedCopy-->
```

> Note
>
> Enter the Alias IP address and netmask configured for the client subnet on the primary VM instance.

```
add ns ip <secondary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

The `secondary_snip` refers to the internal IP address of the server-facing interface of the secondary instance.

> **Note:**
>
> Enter the IP address and netmask configured for the server subnet in the VM instance.

Step 3. Add a virtual server on the primary node.

Type the following command:


```
add <server_type> vserver <vserver_name> <protocol> <
    primary_client_alias_ip> <port>
<!--NeedCopy-->
```

Step 4. Add a service or service group on the primary node.

Type the following command:

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

Step 5. Bind the service or service group to the load balancing virtual server on the primary node.

Type the following command:

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> **Note:**
>
> To save your configuration, type the command `save config`. Otherwise, the configurations are lost after you restart the instances.

Add back-end GCP Autoscaling service

September 14, 2021

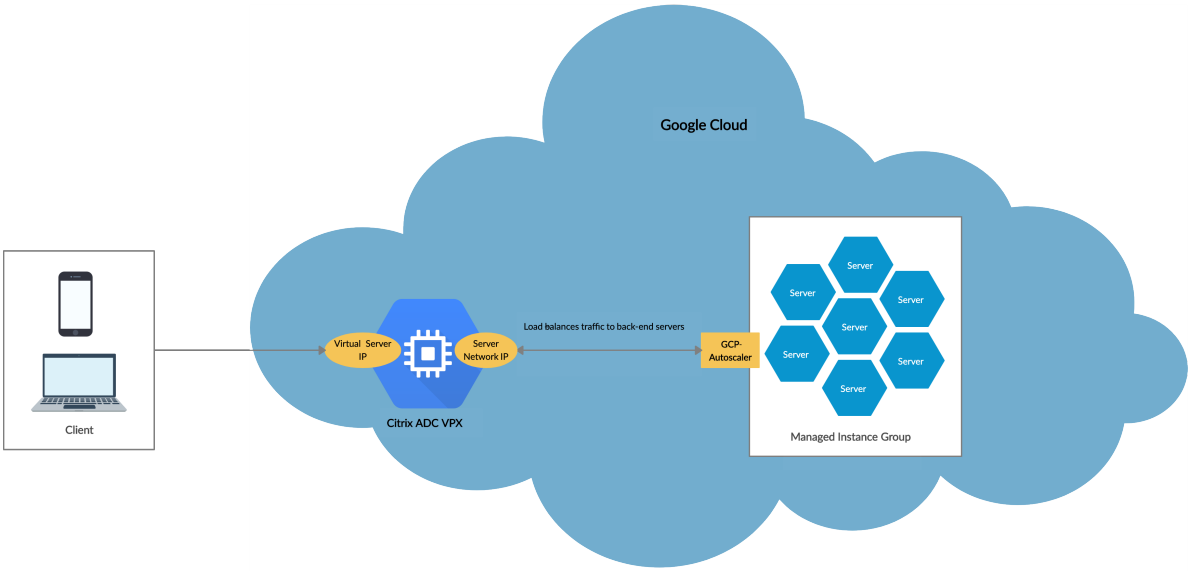
Efficient hosting of applications in a cloud requires easy and cost-effective management of resources, depending on the application demand. To meet the increasing demand, you have to scale network resources upward. When demand subsides, you need to scale down to avoid the unnecessary cost of underutilized resources. To minimize the cost of running the application, you have to constantly monitor traffic, memory and CPU use, and so on. However, monitoring traffic manually is cumbersome. For the application environment to scale up or down dynamically, you must automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

Integrated with the GCP Autoscaling service, the Citrix ADC VPX instance provides the following advantages:

- **Load balance and management**: Auto configures servers to scale up and scale down, depending on demand. The VPX instance auto detects managed instance groups in the back-end subnet and allows you to select the managed instance groups to balance the load. The virtual and subnet IP addresses are auto configured on the VPX instance.

- **High availability**: Detects managed instance groups that span multiple zones and load-balance servers.
- **Better network availability**: The VPX instance supports:
  - Back-end servers on same placement groups
  - Back-end servers on different zones

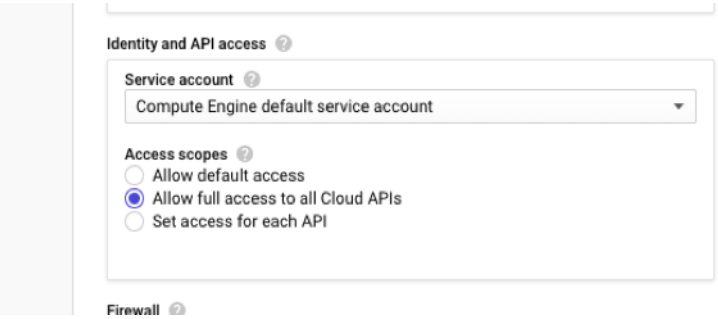
This diagram illustrates how the GCP Autoscaling service works in a Citrix ADC VPX instance acting as the load balancing virtual server.



Before you begin

Before you start using Autoscaling with your Citrix ADC VPX instance, you must complete the following tasks.

- Create a Citrix ADC VPX instance on GCP according to your requirement.
  - For more information about how to create a Citrix ADC VPX instance, see Deploy a Citrix ADC VPX instance on the Google Cloud Platform.
  - For more information about how to deploy VPX instances in HA mode, see Deploy a VPX high-availability pair on the Google Cloud Platform.
- Enable **Cloud Resource Manager API** for your GCP project.
- Allow full access to all Cloud APIs while creating the instances.



- Ensure your GCP service account has the following IAM permissions:

```
REQUIRED_INSTANCE_IAM_PERMS = [

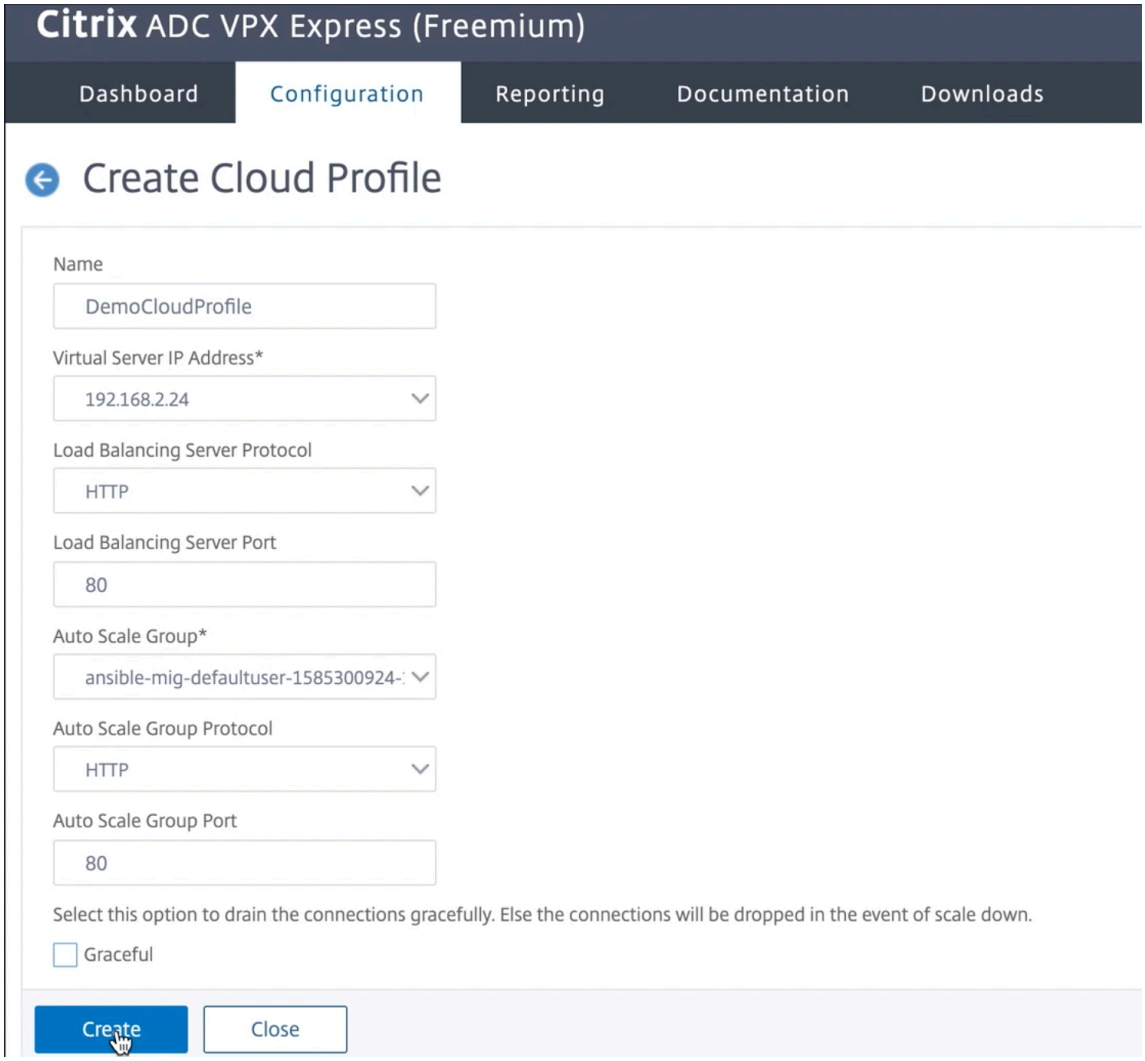
"compute.instances.get",
"compute.zones.list",
"compute.instanceGroupManagers.list",
"compute.instanceGroupManagers.get"
]
<!--NeedCopy-->
```

- To set up Autoscaling, ensure the following are configured:
  - Instance template
  - Managed Instance group
  - Autoscaling policy

Add the GCP Autoscaling service to a Citrix ADC VPX instance

You can add the Autoscaling service to a VPX instance with a single click by using the GUI. Complete these steps to add the Autoscaling service to the VPX instance:

1. Log on to the VPX instance by using your credentials for `nsroot`.
2. When you log on to the Citrix ADC VPX instance for the first time, you see the default Cloud Profile page. Select the GCP managed instance group from the drop-down menu and click **Create** to create a cloud profile.

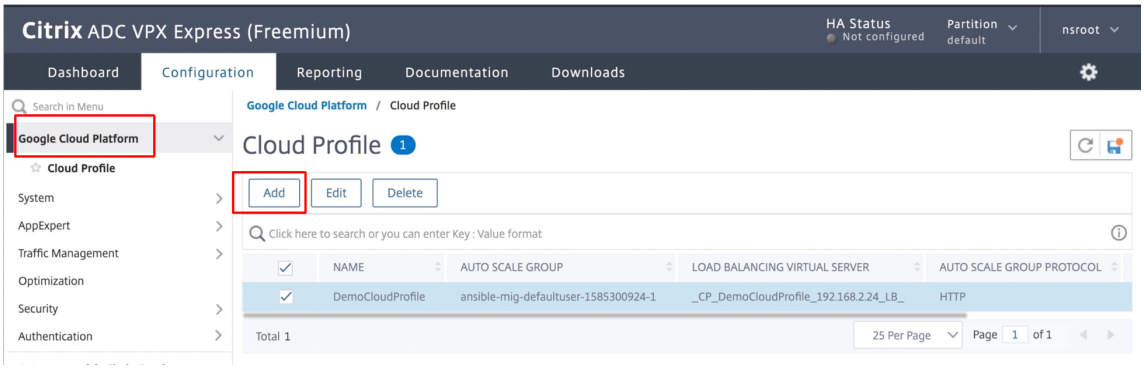


- The **Virtual Server IP Address** field is auto-populated from all the IP addresses associated with the instances.
- The **Autoscale Group** is prepopulated from the managed instance group configured on your GCP account.
- When selecting the **Autoscale Group Protocol** and **Autoscale Group Port**, ensure that your servers listen on the configured protocol and ports. Bind the correct monitor in the service group. By default, the TCP monitor is used.
- Clear the **Graceful** check box because it is not supported.

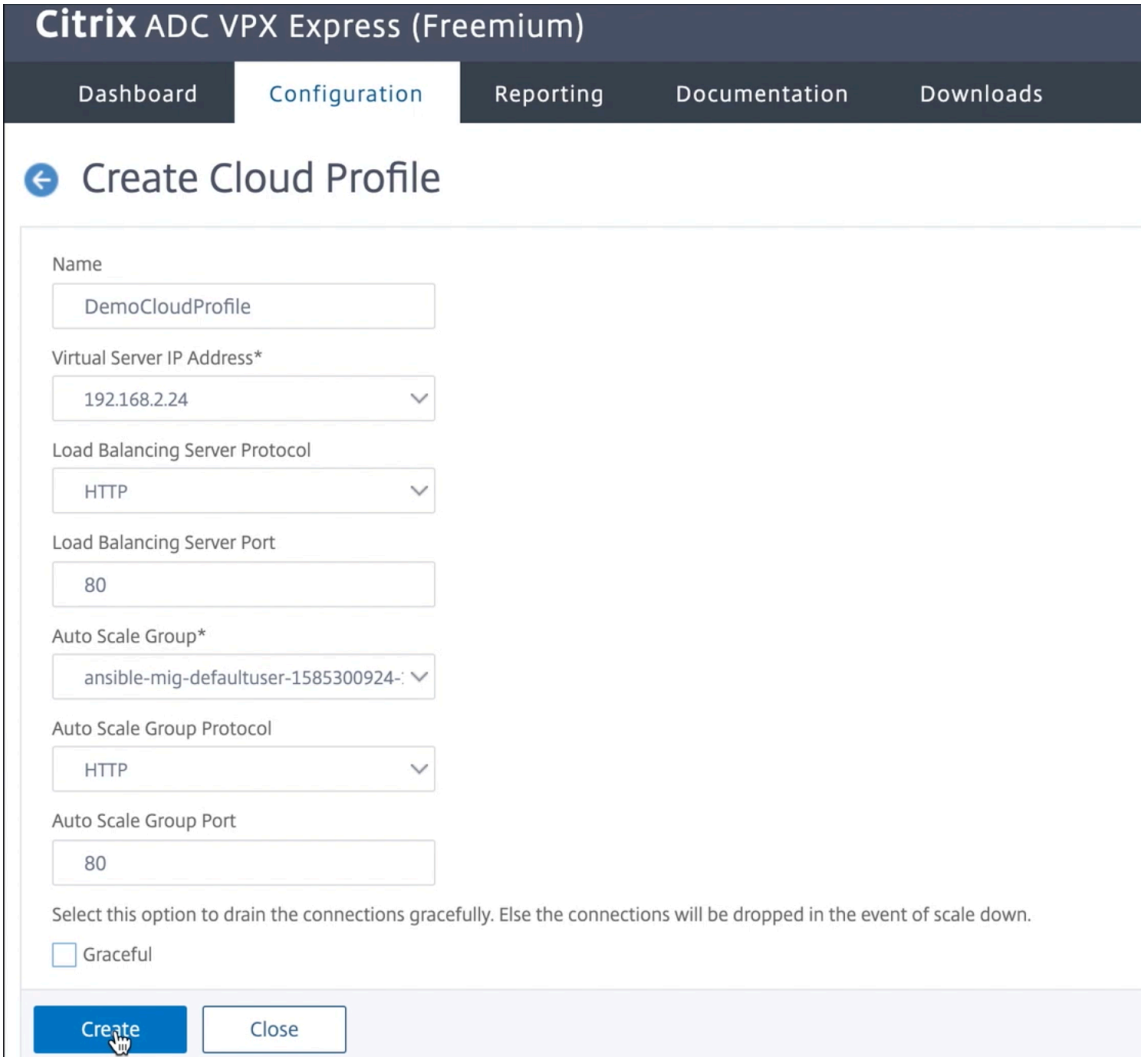
> Note:
>
> For SSL Protocol type Autoscaling, after you create the Cloud Profile, the load balance virtual server or service group is down because of a missing certificate. You can bind the certificate to the virtual server or service group manually.

3. After the first time logon if you want to create Cloud Profile, on the GUI go to **System > Google**

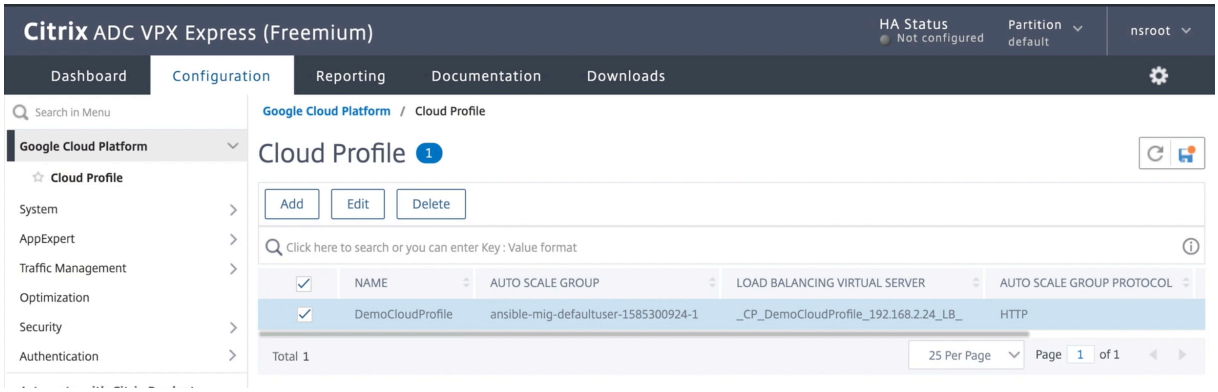
Cloud Platform > Cloud Profile and click **Add**.



The **Create Cloud Profile** configuration page appears.



Cloud Profile creates a Citrix ADC load-balancing virtual server and a service group with members as the servers of the managed instance group. Your back-end servers must be reachable through the SNIP configured on the VPX instance.



VIP scaling support for Citrix ADC VPX instance on GCP

September 14, 2021

A Citrix ADC appliance resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. The number of public virtual IP (VIP) addresses needed for a deployment varies on a case-by-case basis.

The GCP architecture restricts each interface on the instance to be connected to a different VPC. A VPC on GCP is a collection of subnets, and each subnet can span across zones of a region. In addition, GCP imposes the following limitation:

- There is a 1:1 mapping of number of public IP addresses to number of NICs. Only one public IP address can be assigned to a NIC.
- A maximum of only 8 NICs can be attached on a higher capacity instance type.

For example, an n1-standard-2 instance can have only 2 NICs, and the Public VIPs that can be added is limited to 2. For more information, see VPC resource quotas.

To achieve higher scales of public virtual IP addresses on a Citrix ADC VPX instance, you can configure the VIP addresses as part of the metadata of the instance. The ADC VPX instance internally uses forwarding rules provided by the GCP to achieve VIP scaling. The ADC VPX instance also provides high availability to the VIPs configured.
After you configure VIP addresses as part of the metadata, you can configure an LB virtual server using the same IP that is used to create the forwarding rules. Thus, we can use forwarding rules to mitigate the limitations we have w.r.t scale in using public VIP addresses on an ADC VPX instance on GCP.

For more information on forwarding rules, see Forwarding rules overview.

For more information on HA, see High Availability.

Points to note

- Google charges some additional cost for each virtual IP forwarding rule. The actual cost depends on the number of entries created. The associated cost can be found from the Google pricing documents.
- Forwarding rules are applicable only for public VIPs. You can use alias IP addresses when the deployment needs private IP addresses as VIPs.
- You can create forwarding rules only for the protocols, which need the LB virtual server. VIPs can be created, updated, or deleted on the fly. You can also add a new load balancing virtual server with the same VIP address but with a different protocol.

Before you start

- Citrix ADC VPX instance must be deployed on GCP.
- External IP address must be reserved. For more information, see Reserving a static external IP address.
- Ensure that your GCP service account has the following IAM permissions:

```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
"compute.subnetworks.use",
"compute.targetInstances.create"
"compute.targetInstances.get"
"compute.targetInstances.use",
]

<!--NeedCopy-->
```

Configure external IP addresses for VIP scaling on Citrix ADC VPX instance

1. In the Google Cloud Console, navigate to the **VM Instances** page.
2. Create a new VM instance or use an existing instance.
3. Click the instance name. On the **VM instance details** page, click **Edit**.

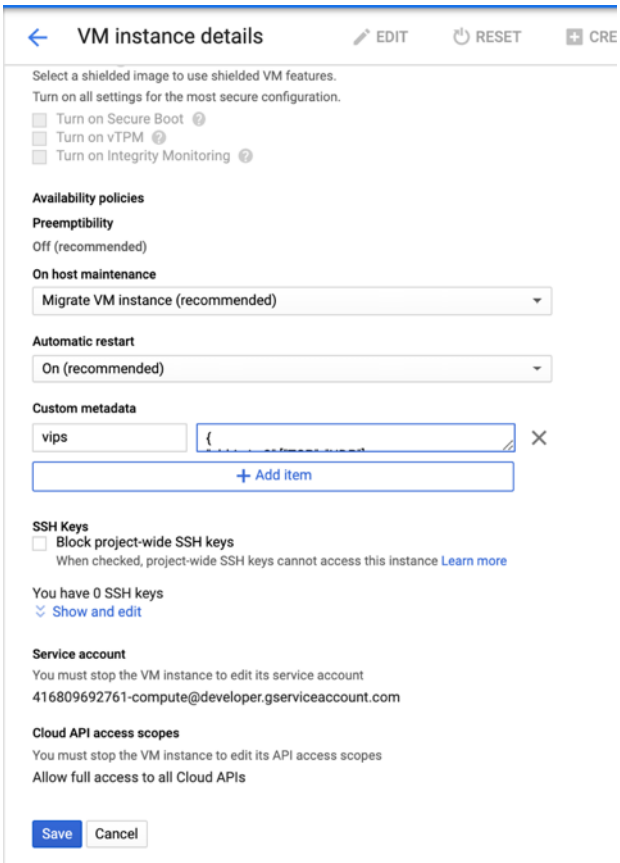
4. Update the **Custom metadata** by entering the following:

   - Key = vips
   - Value = Provide a value in the following JSON format:

     {
     “Name of external reserved IP”: [list of protocols],
     }

   GCP supports the following protocols:

   - AH
   - ESP
   - ICMP
   - SCT
   - TCP
   - UDP



For more information, see Custom metadata.

Example for Custom metadata:

{
“external-ip1-name”:[“TCP”, “UDP”],

“external-ip2-name”:[“ICMP”, “AH”]
}

In this example, the ADC VPX instance internally creates one forwarding rule for each IP, protocol pair. The metadata entries are mapped to the forwarding rules. This example helps you understand how many forwarding rules are created for a metadata entry.

Four forwarding rules are created as follows:

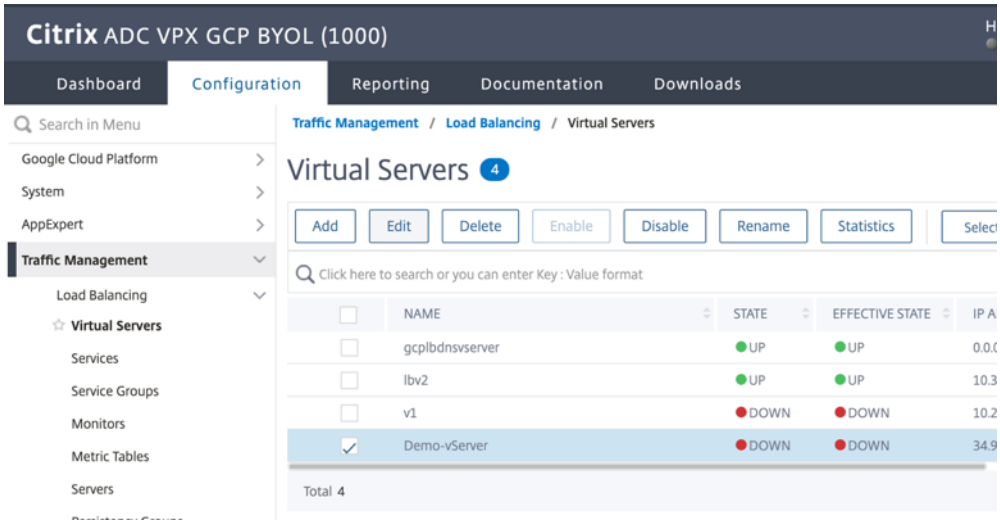
a) external-ip1-name and TCP
b) external-ip1-name and UDP
c) external-ip2-name and ICMP
d) external-ip2-name and AH

5. Click **Save**.

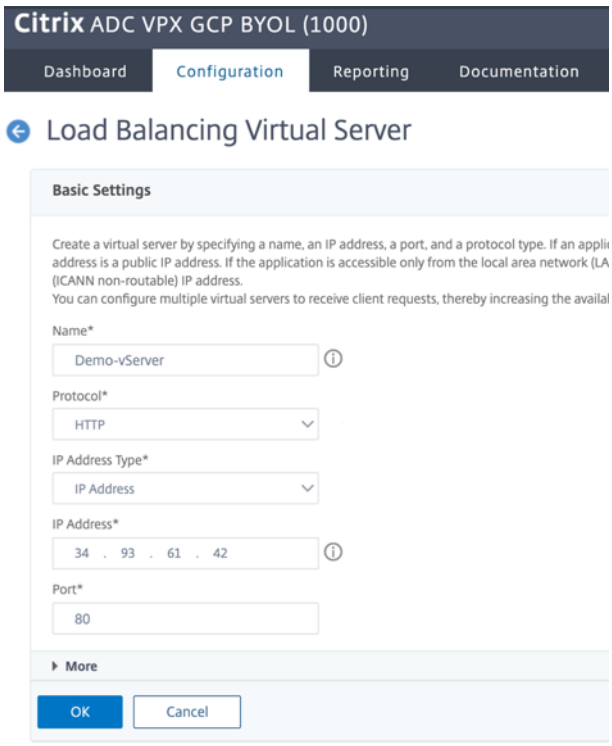
Setting up a load balancing virtual server with external IP address on a Citrix ADC VPX instance

Step 1. Add a load balancing virtual server.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers > Add**.

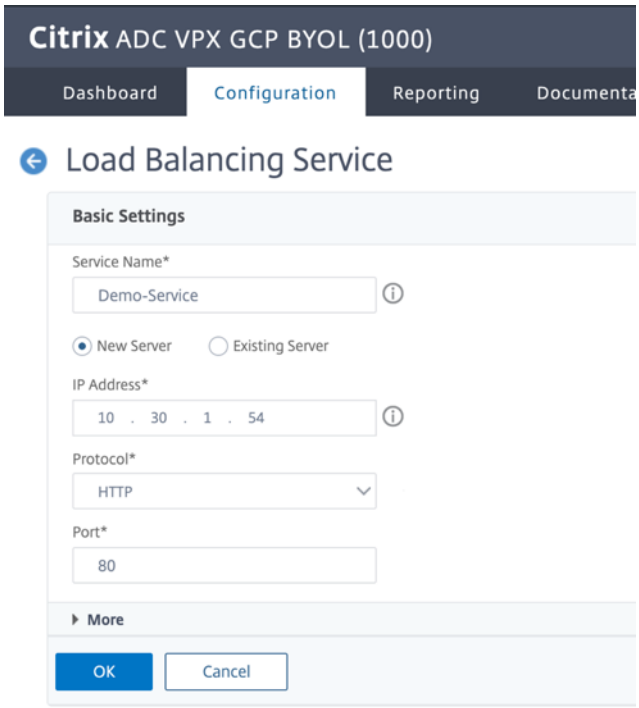


2. Add the required values for Name, Protocol, IP Address Type (IP Address), IP Address (External IP address of the forwarding rule that is added as VIP on ADC) and Port, and click **OK**.



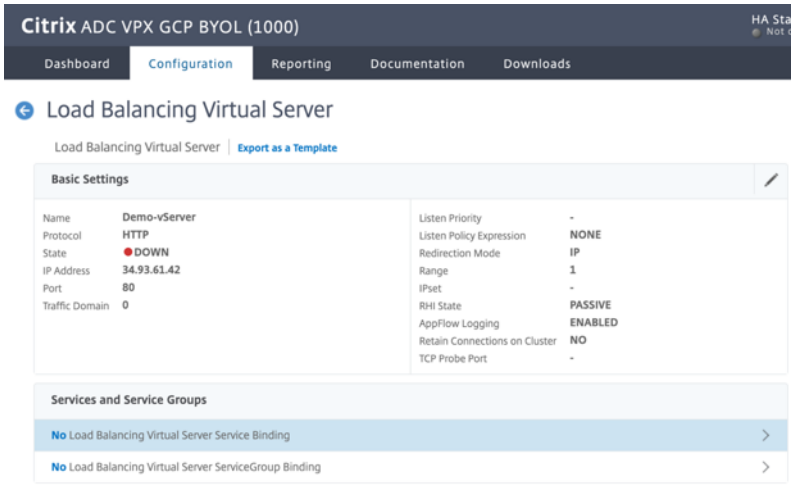
Step 2. Add a service or service group.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Services > Add**.
2. Add the required values for Service Name, IP Address, Protocol and Port, and click **OK**.

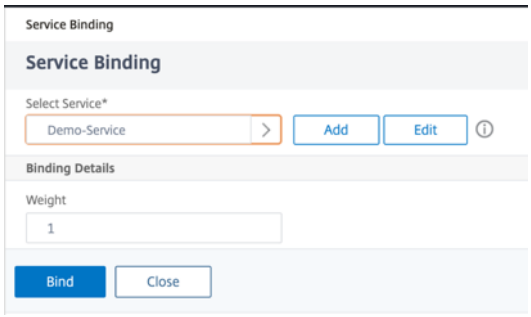


Step 3. Bind the service or service group to the load balancing virtual server.

1. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
2. Select the load balancing virtual server configured in **Step 1**, and click **Edit**.
3. In the **Service and Service Groups** page, click **No Load Balancing Virtual Server Service Binding**.



4. Select the service configured in the **Step 3**, and click **Bind**.



5. Save the configuration.

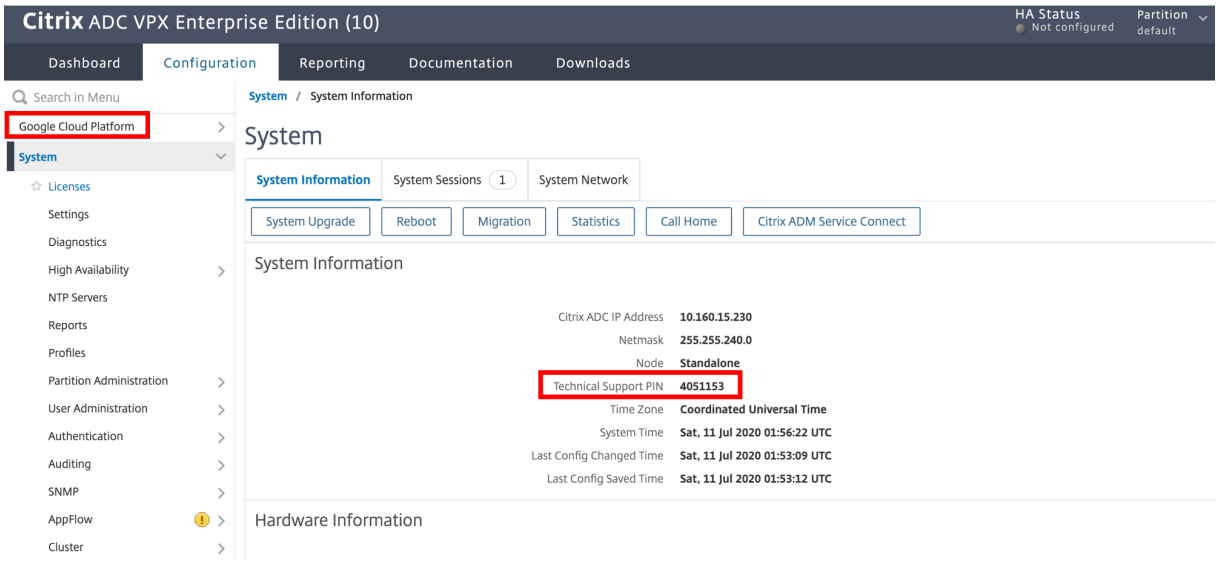
Troubleshoot a VPX instance on GCP

September 14, 2021

Google Cloud Platform (GCP) provides console access to a Citrix ADC VPX instance. You can debug only if the network is connected. To view an instance's System Log, access the console and check **System Log files**.

Citrix supports fee based Citrix ADC VPX instances (utility license with hourly fee) on GCP. To file a support case, find your GCP account number and support PIN code, and call Citrix support. You are asked to provide your name and email address. To find the support PIN, log on to the VPX GUI and navigate to the **System** page.

Here is an example of a system page showing the support PIN.



Jumbo frames on Citrix ADC VPX instances

September 14, 2021

Citrix ADC VPX appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A Citrix ADC appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

For more information, seeConfiguring Jumbo Frames Support on a Citrix ADC Appliance.

Jumbo frames support is available on Citrix ADC VPX appliances running on the following virtualization platforms:

- VMware ESX
- Linux-KVM Platform
- Citrix XenServer
- Amazon Web Services (AWS)

Jumbo frames on VPX appliances work similar to jumbo frames on MPX appliances. For more information on Jumbo Frames and its use cases, see Configuring Jumbo Frames on MPX appliances. The use cases of jumbo frames on MPX appliances also apply to VPX appliances.

Configure jumbo frames for a VPX instance running on VMware ESX

Perform the following tasks to configure jumbo frames on a Citrix ADC VPX appliance running on the VMware ESX server:

1. Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501–9000. Use the CLI or GUI to set the MTU size. The Citrix ADC VPX appliances running on VMware ESX support receiving and transmitting jumbo frames containing up to only 9000 bytes of IP data.
2. Set the same MTU size on the corresponding physical interfaces of the VMware ESX server by using its management applications. For more information about setting the MTU size on the physical interfaces of VMware ESX, see http://vmware.com/.

Configure jumbo frames for a VPX instance running on Linux-KVM server

Perform the following tasks to configure jumbo frames on a Citrix ADC VPX appliance running on a Linux-KVM Server:

1. Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501–9216. Use the Citrix ADC VPX CLI or GUI to set the MTU size.
2. Set the same MTU size on the corresponding physical interfaces of a Linux-KVM Server by using its management applications. For more information about setting the MTU size on the physical interfaces of Linux-KVM, see http://www.linux-kvm.org/.

Configure jumbo frames for a VPX instance running on Citrix XenServer

Perform the following tasks to configure jumbo frames on a Citrix ADC VPX appliance running on Citrix XenServer:

1. Connect to the XenServer using XenCenter.
2. Shut down all the VPX instances that use the Networks for which the MTU must be changed.
3. On the **Networking** tab, select the network - network 0/1/2.
4. Select **Properties** and edit MTU.

After configuring the jumbo frames on the XenServer, you can configure the jumbo frames on the ADC appliance. For more information, see Configuring Jumbo Frames Support on a Citrix ADC Appliance.

Configure jumbo frames for a VPX instance running on AWS

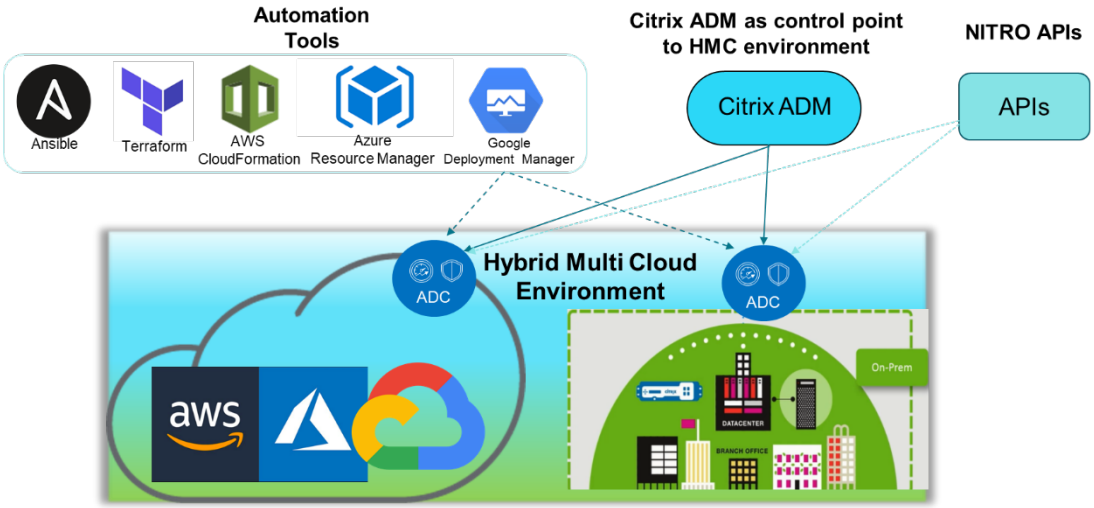
Host-level configuration is not required for VPX on Azure. To configure Jumbo Frames on VPX, follow the steps given in Configuring Jumbo Frames Support on a Citrix ADC Appliance.

Automate deployment and configurations of Citrix ADC

September 14, 2021

Citrix ADC provides multiple tools to automate your ADC deployments and configurations. This document provides a brief summary of various automation tools and references to various automation resources that you can use to manage ADC configurations.

The following illustration provides an overview of Citrix ADC automation in a hybrid multi cloud (HMC) environment.



Automate Citrix ADC using Citrix ADM

Citrix ADM acts as automation control point to your distributed ADC infrastructure. The Citrix ADM provides comprehensive set of automation capabilities from provisioning ADC appliances to upgrading it. The following are the key automation features of ADM:

- Provisioning Citrix ADC VPX instances on AWS
- Provisioning Citrix ADC VPX instances on Azure
- StyleBooks
- Configuration jobs
- Configuration audit
- ADC upgrades

- SSL certificate management
- Integrations - GitHub, ServiceNow, Event notifications integrations

Citrix ADM blogs and videos on automation

- Application migrations using StyleBooks
- Integrate ADC configurations with CI/CD using ADM StyleBooks
- Simplifying public cloud Citrix ADC deployments through ADM
- 10 ways Citrix ADM service supports easier Citrix ADC upgrades

Citrix ADM also provides APIs for its various capabilities that integrate Citrix ADM and Citrix ADC as part of the overall IT automation. For more information, see Citrix ADM Service APIs.

Automate Citrix ADC using Terraform

Terraform is a tool that takes infrastructure as code approach to provision and manage cloud, infrastructure, or service. Citrix ADC terraform resources are available in GitHub for use. Refer GitHub for detailed documentation and usage.

- Citrix ADC Terraform modules to configure ADC for various use cases such as Load Balancing and GSLB
- Terraform cloud scripts to deploy ADC in AWS
- Terraform cloud scripts to deploy ADC in Azure

Videos on Terraform for ADC automation

- Automate your Citrix ADC deployments with Terraform
- Provision and configure ADC in HA setup in AWS using Terraform

Automate Citrix ADC using Ansible

Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. Citrix ADC Ansible modules and sample playbooks can be found in GitHub for use. Refer GitHub for detailed documentation and usage.

- Ansible modules to configure ADC
- Automate ADC with Ansible-whitepaper
- Ansible modules for ADM

Citrix is a certified Ansible Automation Partner. Users having Red Hat Ansible Automation Platform subscription can access Citrix ADC Collections from Red Hat Automation Hub.

Terraform and Ansible automation blogs

- Terraform and Ansible Automation for app delivery and security

Public cloud templates for ADC deployments

Public cloud templates simplify provisioning of your deployments in public clouds. Different Citrix ADC templates are available for various environments. For usage details, refer to respective GitHub repositories.

AWS CFTs:

- CFTs to provision Citrix ADC VPX on AWS

Azure Resource Manager (ARM) Templates:

- ARM templates to provision Citrix ADC VPX on Azure

Google Cloud Deployment Manager (GDM) Templates:

- GDM templates to provision Citrix ADC VPX on Google

Videos on Templates

- Deploy Citrix ADC HA in AWS using CloudFormation Template
- Deploy Citrix ADC HA across Availability Zones using AWS QuickStart
- Citrix ADC HA deployment in GCP using GDM templates

AWS Quick Starts

- Citrix WAF Quick Start
- AWS Quick Start for Citrix ADC VPX for Web Applications on AWS

NITRO APIs

The Citrix ADC NITRO protocol allows you to programmatically configure and monitor the Citrix ADC appliance by using Representational State Transfer (REST) interfaces. Therefore, NITRO applications can be developed in any programming language. For applications that must be developed in Java or .NET or Python, NITRO APIs are exposed through relevant libraries that are packaged as separate Software Development Kits (SDKs).

- NITRO API documentation
- Citrix ADC API reference
- Sample ADC use case configuration using NITRO API

FAQs

September 14, 2021

The following section helps you to categorize the FAQs based on Citrix Application Delivery Controller (ADC) VPX.

- Feature and functionality
- Encryption
- Pricing and packaging
- Citrix ADC VPX Express
- Hypervisor
- Capacity planning or sizing
- System requirements
- Other technical FAQs

Feature and functionality

What is Citrix ADC VPX?

Citrix ADC VPX is a virtual ADC appliance that can be hosted on a Hypervisor installed on industry standard servers.

Does Citrix ADC VPX include all the web application optimization functionality as ADC appliances?

Yes. Citrix ADC VPX includes all load balancing, traffic management, application acceleration, application security (including Citrix ADC Gateway and Citrix Application Firewall), and offload functionality. For a complete overview of the Citrix ADC feature and functionality, see Application delivery your way.

Are there any limitations with Citrix Application Firewall when using it on Citrix ADC VPX?

Citrix Application Firewall on Citrix ADC VPX provides the same security protections as it does on Citrix ADC appliances. Performance or throughput of Citrix Application Firewall varies by platform.

Are there any differences between Citrix ADC Gateway on Citrix ADC VPX and Citrix ADC Gateway on Citrix ADC appliances?

Functionally, they are identical. Citrix ADC Gateway on Citrix ADC VPX supports all the Citrix ADC Gateway features available in Citrix ADC software release 9.1. However, because Citrix ADC appliances provide dedicated SSL acceleration hardware, it offers greater SSL VPN scalability than a Citrix ADC VPX instance.

Other than the obvious difference of being able to run on a Hypervisor, how does Citrix ADC VPX differ from Citrix ADC physical appliances?

There are two main areas where customers see differences in behavior. The first is Citrix ADC VPX cannot offer the same performance as many Citrix ADC appliances. The second is that while Citrix ADC appliances incorporate its own L2 networking functionality, Citrix ADC VPX relies upon the Hypervisor for its L2 networking services. Generally, it does not limit how the Citrix ADC VPX can be deployed. There can be certain L2 functionality that is configured on a physical Citrix ADC appliance must be configured on the underlying Hypervisor.

How does Citrix ADC VPX play a role in the Application Delivery market?

Citrix ADC VPX changes the game in the application delivery market in the following ways:

- By making a Citrix ADC appliance even more affordable, Citrix ADC VPX enables any IT organization to deploy a Citrix ADC appliance. It is not just for their most mission-critical web applications, but for all of their Web applications.
- Citrix ADC VPX allows customers to further converge networking and virtualization within their data centers. Citrix ADC VPX cannot only be used to optimize web applications hosted on virtualized servers. It also enables web application delivery itself to become a virtualized service that can be easily and rapidly deployed anywhere. IT organizations use the standard data center processes for tasks such as provisioning, automation, and charge-back for the web application delivery infrastructure.
- Citrix ADC VPX opens up new deployment architectures that are not practical if only physical appliances are used. Citrix ADC VPX and Citrix ADC MPX appliances can be used basis, tailored to the individual needs of each respective application to handle processor-intensive actions such as compression and application firewall inspection. At the data center edge, Citrix ADC MPX appliances handle high-volume network-wide tasks such as initial traffic distribution, SSL encryption or decryption, denial of service (DoS) attack prevention, and global load balancing. Pairing high-performance Citrix ADC MPX appliances with easy-to-deploy Citrix ADC VPX virtual appliance brings unparalleled flexibility and customization capabilities to modern, large-scale, data center environments while also reducing overall data center costs.

How does Citrix ADC VPX fit into our Citrix delivery center strategy?

With the availability of Citrix ADC VPX, the entire Citrix delivery center offering is available as a virtualized offering. The entire Citrix delivery center benefits from the powerful management, provisioning, monitoring, and reporting capabilities available in Citrix XenCenter. This can be deployed rapidly into almost any environment, and managed centrally from anywhere. With one integrated, virtualized application delivery infrastructure, organizations can deliver desktops, client-server applications, and Web applications.

Encryption

Does Citrix ADC VPX support SSL offload?

Yes. However, Citrix ADC VPX does all SSL processing in software, so Citrix ADC VPX does not offer the same SSL performance as Citrix ADC appliances. Citrix ADC VPX can support up to 750 new SSL transactions per second.

Does third-party SSL cards installed on the server hosting Citrix ADC VPX accelerate SSL encryption or decryption?

No. Supporting third-party SSL cards cannot associate the Citrix ADC VPX to specific hardware implementations. It greatly diminishes an organizations ability to flexibly host Citrix ADC VPX anywhere within the data center. Citrix ADC MPX appliances must be used when more SSL throughput than Citrix ADC VPX provides is required.

Does Citrix ADC VPX support the same encryption ciphers as physical Citrix ADC appliances?

VPX supports all encryption ciphers as physical Citrix ADC appliances, except the ECDSA.

What is the SSL transactions throughput of Citrix ADC VPX?

See Citrix ADC VPX data sheet for information on SSL transactions throughput.

Pricing and packaging

How is Citrix ADC VPX packaged?

Citrix ADC VPX selection is similar to the selection of Citrix ADC appliances. First, the customer selects the Citrix ADC edition based on its functionality requirements. Then, the customer selects the specific Citrix ADC VPX bandwidth tier based on their throughput requirements. Citrix ADC VPX is available in Standard, Advanced, and Premium Editions. Citrix ADC VPX offers from 10 Mbps (VPX 10) to 100 Gbps (VPX 100G). More details can be found in the Citrix ADC VPX data sheet.

Is Citrix ADC VPX priced the same for all Hypervisors?

Yes.

Are the same Citrix ADC SKUs used for VPX on all Hypervisors?

Yes.

Can a Citrix ADC VPX license be moved from one Hypervisor to another (For example from VMware to Hyper-V)?

Yes. Citrix ADC VPX licenses are independent of the underlying Hypervisor. If you decide to move the Citrix ADC VPX virtual machine from one Hypervisor to another, you do not have to get a new license. However, you might need to rehost the existing Citrix ADC VPX license.

Can Citrix ADC VPX instances be upgraded?

Yes. Both the throughput limits and Citrix ADC family edition can be upgraded. Upgrade SKUs for both types of upgrade are available.

If I want to deploy Citrix ADC VPX in a high availability pair, how many licenses do I need?

As with Citrix ADC physical appliances, a Citrix ADC high availability configuration requires two active instances. Therefore, the customer must purchase two licenses.

Citrix ADC VPX Express and 90-day fee trial

Does Citrix ADC VPX Express include all Citrix ADC standard functionality? Does it include Citrix ADC Gateway and load balancing for Citrix Virtual Apps (formerly XenApp) Web Interface and XML broker?

Yes. Citrix ADC VPX Express includes full Citrix ADC Standard functionality. Starting from Citrix ADC release 12.0–56.20, Citrix modified the VPX express behavior.

Does Citrix ADC VPX Express include all Citrix ADC standard functionality? Does it include Citrix ADC Gateway and load balancing for Citrix Virtual Apps Web Interface and XML broker?

Starting from Citrix ADC release 12.0–56.20, VPX Express offers the Citrix ADC Standard Edition feature set, except Gateway functionality. Earlier to the 12.0–56.20 release, VPX expresses includes all features in the standard edition.

Does Citrix ADC VPX Express require a license?

With the new Citrix ADC VPX Express release (12.0–56.20 and onwards), VPX Express is free and requires no license files to install and comes with no commitment. If you have a VPX Express license already, then the prior VPX Express behavior is preserved. If the VPX Express *license file* is removed and the 12.0–56.20 and onwards release is used, the new VPX express behavior takes effect.

Does the Citrix ADC VPX Express license expire?

With the new VPX express, no. There is no license and no expiry date. If you have a VPX express license already, the license expires one year after download.

Does Citrix ADC VPX Express include the five free Citrix ADC Gateway concurrent licenses?

Yes, if you own a VPX express license.

Is there a limit to how many Citrix ADC VPX Expresses a customer can download?

Five.

Does Citrix ADC VPX Express support the same encryption ciphers as Citrix ADC MPX appliances?

For general availability, all the same strong encryption ciphers supported on Citrix ADC appliances are available on Citrix ADC VPX and Citrix ADC VPX Express. It is subjected to the same import or export regulations.

Can I file technical support cases for Citrix ADC VPX Express?

No. A retail Citrix ADC VPX license such as, VPX-10, VPX-200, VPX-1000, VPX- 3000 is required to file technical support cases. However, Citrix ADC VPX Express users are free to use both the Citrix ADC VPX Knowledge Center, and request help from the community using the Z discussion forums.

Can Citrix ADC VPX Express be upgraded to a retail version?

Yes. Simply purchase the retail Citrix ADC VPX license you need, and then apply the corresponding license to the Citrix ADC VPX Express instance.

Hypervisor

What VMware versions does Citrix ADC VPX support?

Citrix ADC VPX supports both VMware ESX and ESXi for versions 3.5 or later. For more information, see Support matrix and usage guidelines

For VMware, how many virtual network interfaces can you allocate to a VPX?

You can allocate up to 10 virtual network interfaces to a Citrix ADC VPX.

From vSphere, how can we access the Citrix ADC VPX command line?

The VMware vSphere client provides built-in access to the Citrix ADC VPX command line through a console tab. Also, you can use any SSH or Telnet client to access the command line. You can use the NSIP address of the Citrix ADC VPX in the SSH or Telnet client.

How can you access the Citrix ADC VPX GUI?

To access the Citrix ADC VPX GUI, type the NSIP of the Citrix ADC VPX, for example, `http://NSIP address` in the address field of any browser.

Can two Citrix ADC VPX instances installed on the same VMware ESX be configured in a high availability setup?

Yes, but it is not recommended. A hardware failure would affect both Citrix ADC VPX instances.

Can two Citrix ADC VPX instances running on two different VMware ESX systems be configured in a high availability setup?

Yes. It is recommended in a high availability setup.

For the VMware, are interface related events supported on Citrix ADC VPX?

No. Interface related events are not supported.

For the VMware, are tagged VLANs supported on Citrix ADC VPX?

Yes. Citrix ADC tagged VLANs are supported on Citrix ADC VPX from release 11.0 and higher. For more information, see Citrix documentation.

For VMware, are link aggregation and LACP supported on Citrix ADC VPX?

No. Link Aggregation and LACP are not supported for Citrix ADC VPX. Link aggregation must be configured at the VMware level.

How do we access Citrix ADC VPX documentation?

The documentation is available from the Citrix ADC VPX GUI. After logging in, select the **Documentation** tab.

Capacity planning or sizing

What performance can I expect with Citrix ADC VPX?

Citrix ADC VPX offers good performance. See Citrix ADC VPX data sheet for a specific performance level achievable using Citrix ADC VPX.

Given that server CPU power varies, how can we estimate the maximum performance of a Citrix ADC instance?

Using a faster CPU can result in higher performance (up to the maximum allowed by the license), while using a slower CPU can certainly limit the performance.

Are Citrix ADC VPX bandwidth or throughput limits for inbound only traffic, or both inbound and outbound traffic?

Citrix ADC VPX bandwidth limits are enforced for traffic inbound to the Citrix ADC only, regardless of whether the request traffic or response traffic. It indicates that a Citrix ADC VPX-1000 (for example) can process both 1 Gbps of inbound traffic and 1 Gbps of outbound traffic simultaneously. Inbound and outbound traffic is not the same as request and response traffic. To the Citrix ADC, both traffic coming from endpoints (request traffic) and traffic coming from origin servers (response traffic) is “inbound” (that is, coming into the Citrix ADC).

Can multiple instances of Citrix ADC VPX be run on the same server?

Yes. However, ensure that the physical server has enough CPU and I/O capacity to support the total workload running on the host, or Citrix ADC VPX performance can be impacted.

If more than one instance of Citrix ADC VPX is running on a physical server, what is the minimum hardware requirement per Citrix ADC VPX instance?

Each Citrix ADC VPX instance must be allocated 2 GB of physical RAM, 20 GB of hard disk space, and 2 vCPUs.

Can I host Citrix ADC VPX and other applications on the same server?

Yes. For example, Citrix ADC VPX, Citrix Virtual Apps Web Interface and Citrix Virtual Apps XML Broker can all be virtualized and can run on the same server. For best performance, ensure that the physical host has enough CPU and I/O capacity to support all the running workloads.

Will adding CPU cores to a single Citrix ADC VPX instance increase the performance of that instance?

Depending on the license, a Citrix ADC VPX instance can use up to 4 vCPU today. Adding an extra CPU to a Citrix ADC VPX instance that can use more CPUs increases the performance.

Why Citrix ADC VPX looks like consuming more than 90% of the CPU even though it is idle?

It is normal behavior and Citrix ADC appliances exhibit the same behavior. To see the true extent of Citrix ADC VPX CPU utilization, use the stat CPU command in the Citrix ADC CLI, or view Citrix ADC VPX CPU utilization from the Citrix ADC GUI. The Citrix ADC packet processing engine is always "looking for work," even when there is no work to be done. Therefore, it does everything to take control of the CPU and not release it. On a server installed with Citrix ADC VPX and nothing else, results in looking like (from the Hypervisor perspective) that Citrix ADC VPX is consuming the entire CPU. Looking at the CPU utilization from "inside Citrix ADC" (by using the CLI or the GUI) provides a picture of Citrix ADC VPX CPU capacity being used.

System requirements

What is the minimum hardware requirement for Citrix ADC VPX?

See Citrix ADC VPX data sheet for its system requirements.

Citrix ADC VPX requires:

- Processor requirements: Dual core server with Intel VT-x.
- Memory available: 2 GB RAM(4 GB if there is VPX running on 13.0 firmware), 20 GB hard drive.
- Hypervisor: Citrix Hypervisor 5.6 or later; VMware ESX/ESXi 3.5 or later, Windows Server 2008 R2 with Hyper-V.
- Connectivity: 100 Mbps minimum. 1 Gbps recommended.

- A NIC compatible with the Hypervisor.

> **Note**
>
> AMD processors are not supported.

What is Intel VT-x?

These features, sometimes referred to as “hardware assist” or “virtualization assist,” trap sensitive or privileged CPU instructions run by the guest OS out to the Hypervisor. This simplifies hosting guest OSs (BSD for a Citrix ADC VPX) on the Hypervisor.

How common are VT-x?

Virtually, all servers shipped within the last two years might support VT-x. Many servers ship with virtualization assist disabled in the BIOS. Before assuming you cannot run Citrix ADC VPX, check if you need to change this setting on the server.

Is there a hardware compatibility list (HCL) for Citrix ADC VPX?

As long as the server supports Intel VT-x, Citrix ADC VPX must run on any server compatible with the underlying Hypervisor. See the Hypervisor HCL for a comprehensive list of supported platforms.

What version of Citrix ADC OS is Citrix ADC VPX based on?

Citrix ADC VPX is based on Citrix ADC 9.1 or later releases.

Since Citrix ADC VPX runs on BSD, can it be run natively on a server with BSD Unix installed?

No. Citrix ADC VPX requires the Hypervisor to run. Detailed Hypervisor supports can be found in Citrix ADC VPX data sheet.

Other technical FAQs

Does link aggregation on a physical server with multiple NIC’s work?

LACP is not supported. For the Citrix Hypervisor, Static link aggregation is supported and has limits of four channels and seven virtual interfaces. For VMware, static link aggregation is not supported within Citrix ADC VPX, but can be configured at the VMware level.

Is MAC based forwarding (MBF) supported on VPX? Is there any change from the Citrix ADC appliance implementation?

MBF is supported and it behaves the same way as with the Citrix ADC appliance. The Hypervisor basically switches all the packets received from Citrix ADC VPX to the outside and conversely.

How is the Citrix ADC VPX upgrade process carried out?

Upgrades are performed the same way as for Citrix ADC appliances: download a kernel file and use install ns or the upgrade utility in the GUI.

What is the size of the `/var` partition when using the default image for VPX? How to increase the disk space?

The size of the root disk is limited to 20 GB to keep the disk image small.

If you want to increase the `/var/core/` or the `/var/crash/` directory space, attach an extra disk. To increase the `/var` size, currently, you must attach an extra disk and create a symbolic link to `/var`, after copying the critical contents to the new disk.

What can we expect to regard the NetScaler VPX build numbering and interoperability with other builds?

Citrix ADC VPX has similar build numbering as the 9.1. Cl (classic) and 9.1. Nc (nCore) release, for instance 9.1_97.3.vpx, 9.1_97.3.nc, and 9.1_97.3.cl.

Can the Citrix ADC VPX be a part of a high availability setup with a Citrix ADC appliance?

Not a supported configuration.

Are all the interfaces visible in Citrix ADC VPX directly related to the number of interfaces on the Hypervisor?

No. You can add up to seven interfaces (10 for VMware) through the Citrix ADC VPX configuration utility with only one physical NIC on the Hypervisor.

Can Citrix Hypervisor XenMotion or VMware VMotion or Hyper-V live migration be used to move active instances of Citrix ADC VPX?

Citrix ADC VPX does not support XenMotion or Hyper-V live migration. VMotion is supported from Citrix ADC 12.1 release onwards. For more information, see Release Notes.

Licensing overview

September 14, 2021

Citrix offers a wide range of product editions and licensing models for MPX and VPX appliances, to meet your organization's need.

For proper operation of a Citrix ADC appliance, it must have one of the Citrix ADC family edition licenses. The ADC product line has three family editions:

- Standard Edition
- Advanced Edition
- Premium Edition

For more information, see the Citrix ADC data sheet.

After you've selected the Citrix ADC edition, you can select one of the MPX and VPX licensing offerings. Based on the criteria such as perpetual and subscription (yearly and hourly subscription), vCPU and bandwidth, on-premises and cloud, and so on.

Citrix ADC VPX licenses

The following are VPX-specific licenses.

Citrix ADC VPX Express license

Starting with Citrix ADC release 12.0 56.20, VPX Express for on-premises and cloud deployments does not require a license file and it comes with the following features:

- 20 Mbps bandwidth
- All ADC standard license features, except Citrix Gateway and L4 and L7 defenses
- Maximum 250 SSL sessions
- 20 Mbps SSL throughput

You can upgrade the VPX Express License to the following two options:

1. A standalone Citrix ADC VPX license
2. Citrix ADC Pooled Capacity license for VPX instances. For more information, see Citrix ADC Pooled Capacity.

> **Important**
>
> Clustering is available in Standard edition for VPX public cloud, and in VPX Express license.

Citrix ADC VPX pooled capacity license

You can use Citrix Application Delivery Management (ADM) to create a licensing framework that comprises a common bandwidth and instance pool. For complete information, see Citrix ADC pooled capacity.

Related resources

The Citrix Licensing System

How to Allocate Citrix ADC VPX Licenses

VPX licensing on cloud

VPX deployment is supported on public cloud providers such as Azure, AWS, and Google. For more information, see the following documents:

- VPX-Azure license
- VPX-AWS license
- VPX-GCP license

Allocate and apply a license

September 14, 2021

In the Citrix MPX and VPX ADC GUI, you can use your hardware serial number (HSN) or your license access code to allocate your licenses. Alternatively, if a license is already present on your local computer, you can upload it to the appliance.

For all other functionality, such as returning or reallocating your license, you must use the licensing portal. Optionally, you can still use the licensing portal for license allocation. For more information, see Use Manage Licenses in My Account on citrix.com.

Citrix Licensing guide

Citrix Licensing guide also covers information about installing licenses in a Citrix ADC appliance and installing licenses in other Citrix products. For more information, see Citrix Licensing Guide.

Prerequisites

> **Note:**
>
> Purchase separate licenses for each appliance in a high availability pair. Ensure that the same types of licenses are installed on both the appliances. For example, if you purchase a Premium license for one appliance, you must purchase another Premium license for the other appliance.

To use the hardware serial number or license access code to allocate your licenses:

- You must be able to access public domains through the appliance. For example, the appliance should be able to access www.citrix.com. The license allocation software internally accesses the Citrix licensing portal for your license. To access a public domain:
  - Use a proxy server or set up a DNS server.
  - Configure a Citrix ADC IP (NSIP) address or a subnet IP (SNIP) address on your Citrix ADC appliance.
- Your license must be linked to your hardware, or you must have a valid license access code. Citrix sends your license access code by email when you purchase a license.

Allocate a license by using the GUI

If your license is already linked to your hardware, the license allocation process can use the hardware serial number. Otherwise, you must type the license access code.

You can partially allocate licenses as required for your deployment. For example, if your license file contains 10 licenses, but your current requirement is for only six licenses, you can allocate six licenses now, and allocate more licenses later. You cannot allocate more than the total number of licenses present in your license file.

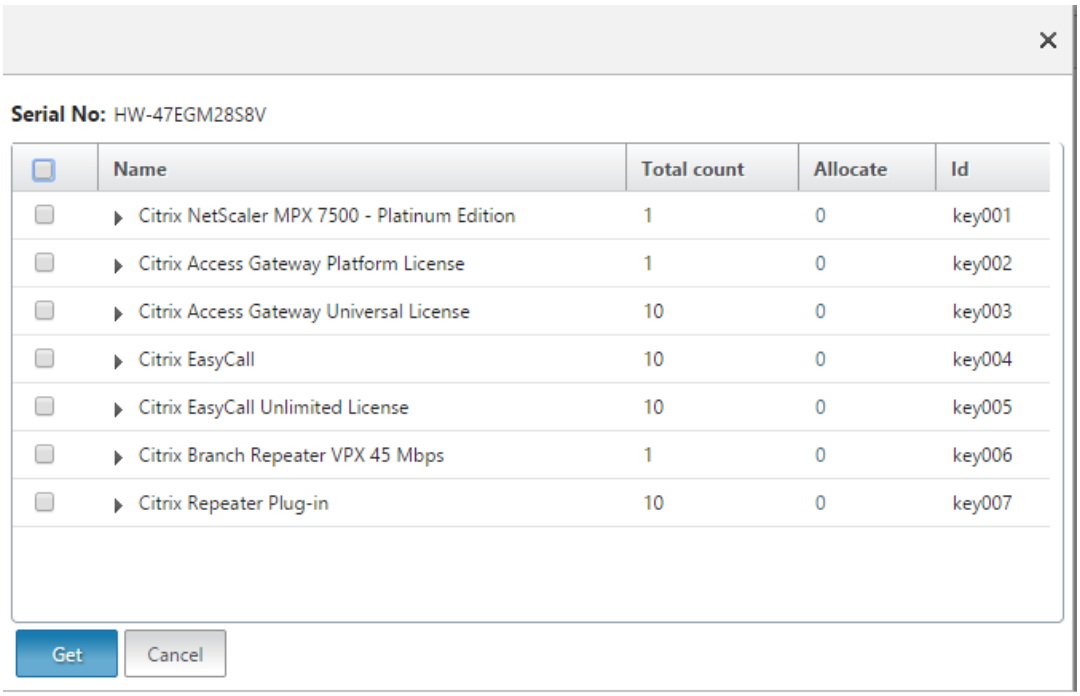
To allocate your license

1. In a web browser, type the IP address of the Citrix ADC appliance (for example, `http://192.168.100.1`).
2. In User Name and Password, type the administrator credentials.
3. On the **Configuration** tab, navigate to **System > Licenses**.
4. In the details pane, click **Manage Licenses**, click **Add New License**, and then select one of the following options:
   - **Use Serial Number**: The software internally fetches the serial number of your appliance and uses this number to display your licenses.
   - **Use license access code**: Citrix emails the license access code for the license that you purchased. Enter the license access code in the text box.

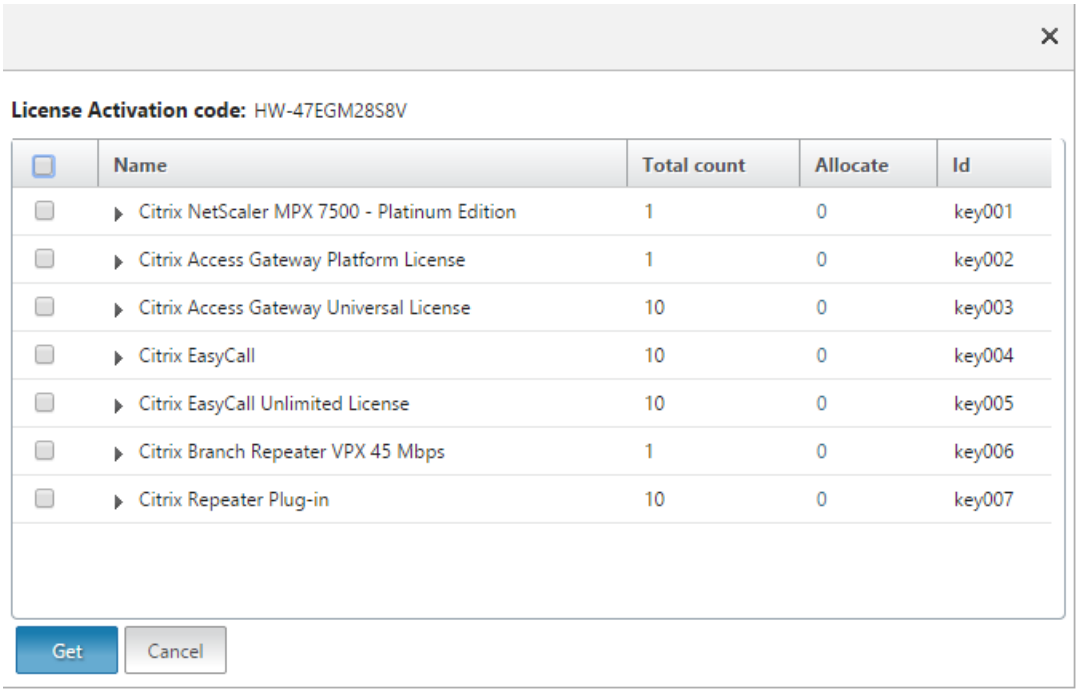
If you do not want to configure internet connectivity on the Citrix ADC appliance, you can use a proxy server. Select the **Connect through Proxy Server** check box and specify the IP address and port of your proxy server.

5. Click **Get Licenses**. Depending on the option that you selected, one of the following dialog boxes appears.

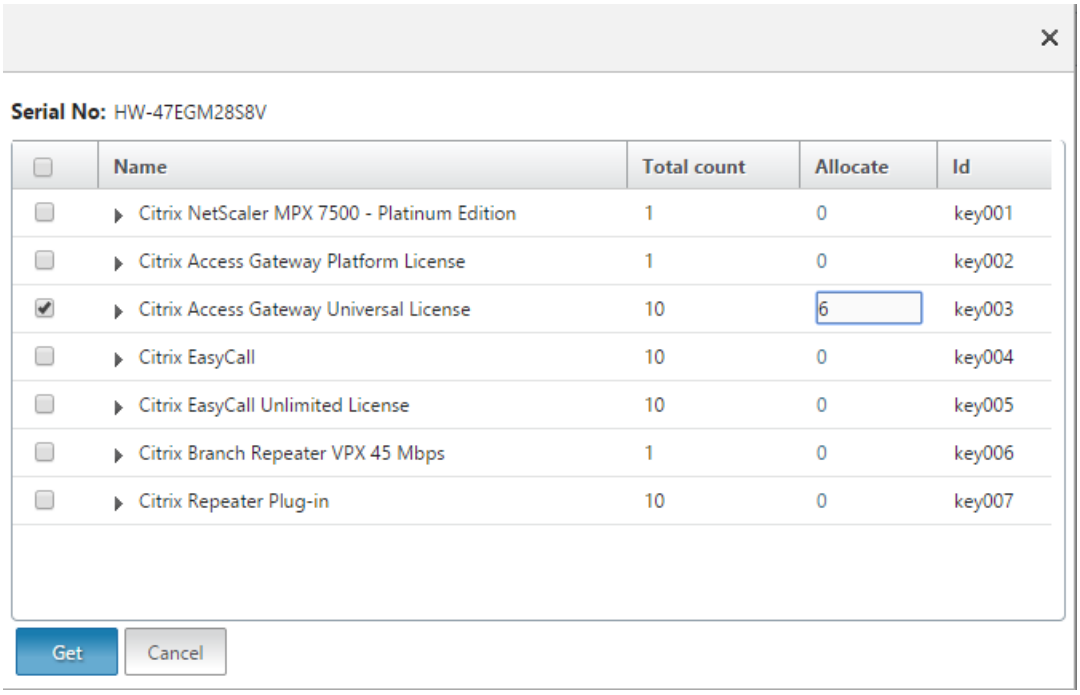
   - The following dialog box appears if you selected Hardware Serial Number.



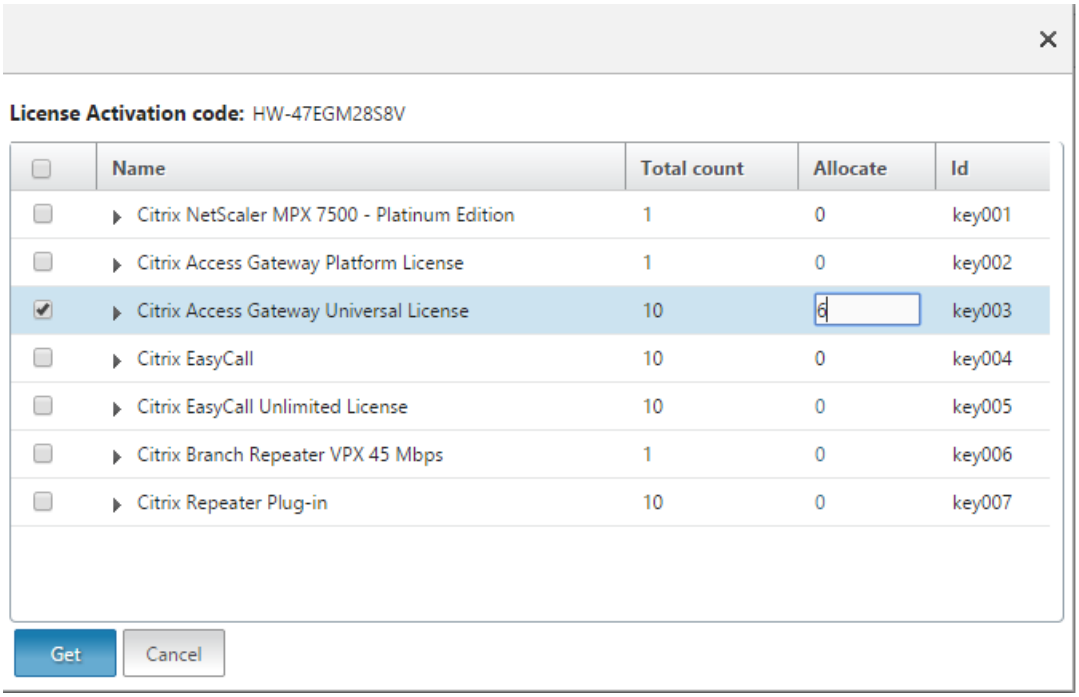
   - The following dialog box appears if you selected license access code.



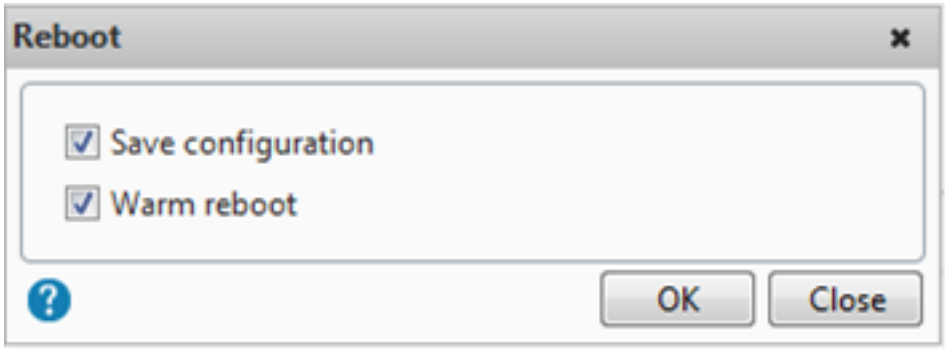
6. Select the license file that you want to use to allocate your licenses.
7. In the **Allocate** column, enter the number of licenses to be allocated. Then click **Get**.
   - If you selected **Hardware Serial Number**, enter the number of licenses, as shown in the following screenshot.



   - If you selected **license access code**, enter the number of licenses, as shown in the following screenshot.



8. Click restart for the license to take effect.
9. In the restart dialog box, click **OK** to proceed with the changes, or click **Close** to cancel the changes.



Install a license

If you downloaded your license file to your local computer by accessing the licensing portal, you must upload the license to the appliance.

To install a license file by using the GUI

1. In a web browser, type the IP address of the Citrix ADC appliance (for example, `http://192.168.100.1`).
2. In User Name and Password, type the administrator credentials.

3. On the **Configuration** tab, navigate to System Licenses.
4. In the details pane, click **Manage Licenses**.
5. Click **Add New License**, then select **Upload license files from a local computer**.
6. Click **Browse**. Navigate to the location of the license files, select the license file, and then click **Open**.
7. Click restart to apply the license.
8. In the restart dialog box, click **OK** to proceed with the changes, or click **Close** to cancel the changes.

To install the licenses by using the CLI

1. Open an **SSH connection** to the ADC appliance by using an SSH client, such as PuTTY.
2. Log on to the ADC appliance by using the administrator credentials.
3. Switch to the shell prompt, create a license subdirectory in the `nsconfig` directory, if it does not exist, and copy one or more new license files to this directory.

   Example

   ```
   login: nsroot
   Password: nsroot
   Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
   Done
   > shell
   Last login: Mon Aug  4 03:51:42 from 10.103.25.64
   root@ns# mkdir /nsconfig/license
   root@ns# cd /nsconfig/license
   <!--NeedCopy-->
   ```

   Copy one or more new license files to this directory.

   Note: The Citrix ADC appliance does not prompt for a reboot option when you use the command line interface to install the licenses. Run the reboot -w command to warm restart the system, or run the restart command to restart the system normally.

Verify licensed features

Before using a feature, ensure that your license supports the feature.

To verify the licensed features by using the CLI

1. Open an **SSH connection** to the ADC appliance by using an SSH client, such as PuTTY.
2. Log on to the ADC appliance by using the administrator credentials.

3. At the command prompt, enter the sh ns license command to display the features supported by the license.

   Example

```
sh ns license
        License status:
                              Web Logging: YES
                        Surge Protection: YES
                          .......

                          HTML Injection: YES
 Done
<!--NeedCopy-->
```

To verify the licensed features by using the GUI

1. In a web browser, type the IP address of the ADC appliance, such as `http://192.168.100.1`.
2. In User Name and Password, type the administrator credentials.
3. Provide the User name and Password and click **Login**.
4. In the navigation pane, expand **System**, and then click **Licenses**. You see a green check mark next to the licensed features.

Enable or disable a feature

When you use the Citrix ADC appliance for the first time, you must enable a feature before you can use its functionality. If you configure a feature before it is enabled, a warning message appears. The configuration is saved but it applies only after the feature is enabled.

To enable a feature by using the CLI

At the command prompt, type the following commands to enable a feature and verify the configuration:

- enable feature <FeatureName>
- show feature

  Example

```
 enable feature lb cs
 done
  >show feature

```



```
5           Feature                         Acronym
               Status
6           -------                         -------
               ------
7    1)     Web Logging                     WL                    OFF
8    2)     Surge Protection                SP                    ON
9    3)     Load Balancing                  LB                    ON
10   4)     Content Switching               CS                    ON
11   5)     Cache Redirection               CR                    ON
12   .
13   .
14   .
15   24)    NetScaler Push                  push                  OFF
16   Done
17  <!--NeedCopy-->
```

The example shows how to enable load balancing (lb) and content switching (cs).

If the license key is not available for a particular feature, the following error message appears for that feature:

ERROR: feature(s) not licensed

Note: To enable an optional feature, you must have a feature-specific license. For example, you have purchased and installed the Citrix NetScaler Advanced Edition license. However, to enable the Integrated Caching feature, you must purchase and install the AppCache license.

To disable a feature by using the CLI

At the command prompt, type the following commands to disable a feature and verify the configuration:

- disable feature <FeatureName>
- show feature

Example

The following example shows how to disable load balancing (LB).

```
1   > disable feature lb
2    Done
3   > show feature
4
5           Feature                         Acronym
               Status
```



```
6           -------                         -------
                ------
7    1)     Web Logging                     WL                  OFF
8    2)     Surge Protection                SP                  ON
9    3)     Load Balancing                  LB                  OFF
10   4)     Content Switching               CS                  ON
11   .
12   .
13   .
14   24)    NetScaler Push                  push                OFF
15   Done
16  >
17  <!--NeedCopy-->
```

Check license expiry information

You can check Citrix ADC license expiry information through GUI or CLI.

To check Citrix ADC license expiry information through GUI:

Go to **Configuration > System > Licenses**.

A GUI alert appears when the ADC license expiry date is less than or equal to 30 days.

To check the license expiry information through CLI:

Type the command “show ns license”.


```
> sh license
    License status:

    Web Logging: YES
    Surge Protection: YES

    Web Logging: YES
    Surge Protection: YES

    …

Days to expiry: 204

Done
>
<!--NeedCopy-->
```

After the license expires, the Citrix ADC appliance generates an SNMP alarm “NS_LICENSE_EXPIRY,” and an expiry event is logged to console.

Upon license expiry, the Citrix ADC appliance automatically restarts to revoke the license. If Citrix ADC appliance uses Citrix service provider (CSP) licenses, the appliance does not restart automatically to revoke the license. However, if the user restarts the appliance, it restarts as unlicensed.

Upgrade a license

You can upgrade a Citrix ADC appliance from one family edition to another and from one capacity range to another by purchasing a higher capacity license.

Upgrades are of two types:

- Edition upgrades: Standard to Advanced, Standard to Premium, and Advanced to Premium. Edition upgrades must be within the same bandwidth.
- Capacity upgrades: You can upgrade from lower to higher capacity, for both vCPU and bandwidth. Capacity upgrades can only be performed on the same Edition (Standard, Advanced, or Premium).

If you want to upgrade both capacity and edition, first upgrade capacity, restart the appliance, and then upgrade the edition.

Example: To upgrade a VPX 10 Mbps Standard Edition license to the VPX 200 Mbps Premium Edition, the upgrade must be done in two steps.

- VPX upgrade from 10 Mbps Standard Edition to 200 Mbps Standard Edition.
- VPX upgrade from 200 Mbps Standard Edition to 200 Mbps Premium Edition.

> Note
>
> You can use Citrix Application Delivery Management (ADM) to create a licensing framework that comprises a common bandwidth and instance pool. For complete information, see Citrix ADC pooled capacity.

Related resources

- The Citrix Licensing System
- How to Allocate Citrix ADC VPX Licenses

Data governance

September 14, 2021

What is Citrix ADM service connect?

Citrix Application Delivery Management (ADM) service connect is a feature to enable seamless on-boarding of Citrix ADC MPX, SDX, and VPX instances, and Citrix Gateway appliances onto Citrix ADM service. This feature lets the Citrix ADC instance or Citrix Gateway appliance automatically, securely connect with Citrix ADM service, and send system, usage, and telemetry data to it. Based on this data, you get insights and recommendations for your Citrix ADC infrastructure on Citrix ADM service.

By using the Citrix ADM service connect feature and onboarding your Citrix ADC instances or Citrix Gateway appliances to Citrix ADM service. You can also manage all your Citrix ADC and Citrix Gateway assets whether on-premises or in the cloud. Also, you benefit from access to a rich set of visibility features that help in quick identification of performance issues, high resource usage, critical errors, and so on. Citrix ADM service provides a wide range of capabilities for your Citrix ADC instances and applications. For more information on Citrix ADM service, see Citrix Application Delivery Management Service

> **Important**
>
> - Citrix Gateway appliance also supports the Citrix ADM service connect feature. For better ease, the Citrix Gateway appliance isn't called explicitly in the consecutive sections.

What is Citrix ADM service?

Citrix ADM service is a cloud-based solution that helps you manage, monitor, orchestrate, automate, and troubleshoot your Citrix ADC instances. It also provides you analytical insights and curated ma-

chine learning based recommendations about Citrix ADC instances and about application health, performance, and security. For more information, see Citrix ADM service Overview

How the Citrix ADM service connect is enabled?

Citrix ADM service connect is enabled by default, after you install or upgrade Citrix ADC or Gateway to release 13.0 build 61.xx and above.

What data is captured using Citrix ADM service connect?

The following details are captured using Citrix ADM service connect:

- **Citrix ADC details**
  - Serial ID
  - Encoded Serial ID
  - Host ID
  - UUID
  - Management IP address
  - Host name
  - Version
  - Build type
  - Build
  - License type
  - Hypervisor
  - Deployment type(standalone/HA)
  - Platform type
  - Platform description
  - System ID
  - Modes enabled on ADC
  - Features enabled on ADC
- **License Information**
  - Features licensed on Citrix ADC
  - License number
- **Key usage metrics**
  - System date time
  - CPU usage percentage
  - Management CPU percentage
  - Throughput
  - SSL new sessions

  - SSL encryption throughput
  - SSL decryption throughput
  - System Uptime

- **Configuration**

  - ns.conf file

    > **Note**
    >
    > Before the Citrix ADM service connect sends the `ns.conf` file from Citrix ADC appliance to the Citrix ADM service, it anonymizes the encrypted or hashed passwords. The Citrix ADM service connect checks for "-encrypted" or "-passcrypt" parameters and replaces the associated encrypted or hashed value with 'XXXX'. The Citrix ADM service connect then encodes and compresses the `ns.conf` file, and sends it to the Citrix ADM service endpoint.

- **Critical error details**

  - Hard disk failures
  - SSL card failures
  - Power Supply Unit (PSU) failures
  - Flash drive failure
  - Warm reboot
  - Sustained memory usage above 90% or a memory leak
  - Sustained rate limit drops

How the data is used?

By collecting the data, Citrix can provide you with timely and in-depth insights about your Citrix ADC installations, which include the following:

- **Key metrics**. Details of key metrics about CPU, memory, throughput, SSL throughput, and highlight anomalous behavior on Citrix ADC instances.
- **Critical errors**. Any critical errors that might have occurred on your Citrix ADC instances.
- **Deployment advisory**. Identify Citrix ADC instances that are deployed in standalone mode but have high throughput and are vulnerable to a single point of failure.

How long the collected data is kept?

Any data collected is kept for no longer than 13 months.

If you decide to terminate the use of the service by disabling the Citrix ADM service connect feature from the Citrix ADC, any previously collected data is deleted after a period of 30 days.

Where the data is stored and how secure is it?

All data collected by Citrix ADM service connect is stored in one of the three regions–United States, European Union, and Australia and New Zealand (ANZ). For more information, see Geographical Considerations.

The data is stored securely with strict tenant isolation at the database layer.

How to disable Citrix ADM service connect?

If you want to disable data collection through Citrix ADM service connect, see How to enable and disable Citrix ADM service connect.

Introduction to Citrix ADM service connect for Citrix ADC appliances

September 14, 2021

Citrix ADM service is a cloud-based solution that helps you manage, monitor, orchestrate, automate, and troubleshoot your Citrix ADC instances. It also provides analytical insights and curated machine learning based recommendations for your applications health, performance, and security. For more information, see Citrix Application Delivery Management Service.

Citrix Application Delivery Management (ADM) service connect is a feature to enable seamless onboarding of Citrix ADC instances onto Citrix ADM service. This feature helps Citrix ADC instances and Citrix ADM service to function as a holistic solution, which offers customers multi fold benefits.

Citrix ADM service connect feature lets the Citrix ADC instance automatically connect with the Citrix ADM service and send system, usage, and telemetry data to it. Based on this data, the Citrix ADM service gives you some insights and recommendations on your Citrix ADC and Gateway infrastructure like the following:

- Security advisory insight highlighting your vulnerable ADC appliances.
- Upgrade advisory insight highlighting ADC appliances that have reached or about to reach end of maintenance and end of life.
- Quick identification of performance issues, high resource usage, and critical errors.

To harness the power of Citrix ADM service, you can choose to onboard your Citrix ADC instances to Citrix ADM service. The onboarding process uses ADM service connect, and makes the experience smooth and faster for you.

> **Points to note**
>
> - Citrix ADM service connect is now available on Citrix ADC MPX, SDX, and VPX instances and

> Citrix Gateway appliances.
>
> - The initiative in Citrix ADM service that uses this Citrix ADM service connect feature is ADM service connect based low-touch onboarding. For more information, see Low-touch on-boarding of Citrix ADC instances using Citrix ADM service connect.

For more information, see Data governance.

> **Important**
>
> Citrix ADM service connect fails to collect the probe data and cannot help in on-boarding the ADC appliance to ADM service if the following conditions are met:
>
> - `NSinternal` user account is disabled.
> - SSH public key is not set up.
>
> To overcome the preceding scenario, Citrix recommends you follow any one of the following:
>
> - Enable `internaluser` user account by using the `set ns param -internaluserlogin ENABLED`.
> - Configure the public key authentication. For more information, see Access a Citrix ADC appliance by using SSH keys and no password.

How does Citrix ADM service connect support with Citrix ADM service?

Here is a high-level workflow of how the Citrix ADM service connect feature on Citrix ADC interacts with Citrix ADM service.

1. Citrix ADM service connect feature on Citrix ADC appliance auto connects with Citrix ADM service using a periodic probe request.
2. This request has system, usage and telemetry data, using which the Citrix ADM service gives you some insights and recommendations on your Citrix ADC infrastructure. Like; quick identification of performance issues, high resource usage, and critical errors.
3. You can view the insights and recommendations and decide to onboard your ADC instances to the Citrix ADM service to start managing your Citrix ADC instances.
4. When you decide to onboard, the Citrix ADM service connect feature helps complete the on-boarding seamlessly.

What versions of Citrix ADC is Citrix ADM service connect supported on?

Citrix ADM service connect is supported on all Citrix ADC platforms and all appliance models (MPX, VPX, and SDX). Starting from Citrix ADC release 13.0 build 61.xx, Citrix ADM service connect is enabled by default for Citrix ADC appliances.

How to enable Citrix ADM service connect?

If you are an existing Citrix ADC customer, and upgrade to Citrix ADC release 13.0 build 61.xx, Citrix ADM service connect is enabled by default as part of the upgrade process.

If you are a new Citrix ADC customer, installing Citrix ADC release 13.0 build 61.xx, Citrix ADM service connect is enabled by default as part of the install process.

> **Note**
>
> Unlike the new Citrix ADC appliances, existing Citrix ADC appliances find the route through Citrix Insight Service (CIS) or Call Home.

How to enable and disable Citrix ADM service connect?

You can enable and disable Citrix ADM service connect from CLI, GUI, or NITRO API methods.

Using CLI

To enable the Citrix ADM service connect by using the CLI

At the command prompt, type:

```
set adm parameter -admserviceconnect ENABLED
```

To disable the Citrix ADM service connect by using the CLI

At the command prompt, type:

```
set adm parameter -admserviceconnect DISABLED
```

> **Important**
>
> If your Citrix ADC is on release 13.0 build 61.xx, the parameter name to enable or disable Citrix ADC service connect is “autoconnect.” For example, to enable service connect, use the `set adm parameter -autoconnect ENABLED` command.

Using the GUI

To disable the Citrix ADM service connect by using the Citrix ADC GUI

1. In a web browser, type the IP address of the Citrix ADC appliance (for example, http://192.0.2.10).
2. In **User Name** and **Password**, enter the administrator credentials.
3. Navigate to **System > Settings > Configure ADM Parameters**.

4. On the **Configure ADM Parameters** page, clear the **Enable Citrix ADM service connect** dialog box, and click **OK**.

Using the NITRO API

You can disable Citrix ADM service connect by using the **NITRO** command.

- In Citrix ADC release 13.0 build 61.xx, you can enable or disable the Citrix ADM service connect by using the following command:

  - `curl -X PUT -H "Content-Type:application/json"http://192.0.2.10/nitro/v1/config/admparameter> -d '{ "admparameter":{ "autoconnect":"enabled" } } ' -u nsroot:Test@1`

- From Citrix ADC release 13.0 build 64.xx, the “autoconnect” parameter name is renamed to `admserviceconnect`. You can disable the Citrix ADM service connect by using the following command:

  - `curl -X PUT -H “Content-Type:application/json” http://192.0.2.10/nitro/v1/config/admparameter -d '{ "admparameter":{ "admserviceconnect":"disabled" } } ' -u nsroot:Test@1`

> **Note**
>
> While onboarding the Citrix ADC instances onto Citrix ADM service, you might come across issues. You can troubleshoot the issues using the diagnostic tool. For more information, see Troubleshoot issues using the diagnostic tool.

Citrix ADM built-in agent behavior

From Citrix ADC release 13.0 build 61.xx and higher, the Citrix ADM built-in agent available on Citrix ADC instances communicates with ADM service. It communicates without the need for manual initialization on the respective ADC instance. After communication with ADM service is established, the built-in agent stays evergreen by auto-upgrading itself to the latest software version regularly.

Previously, you had to initialize the built-in agent on the ADC instances, using `mastools` commands, to establish communication with ADM service and for regular auto-upgrades.
For more information, see Configure the ADC built-in agent to manage instances.

References

For more information on Citrix ADM service connect, see the following topics:

- Data governance: Data governance.
- Citrix ADM service: Citrix Application Delivery Management Service.

Upgrade and downgrade a Citrix ADC appliance

September 14, 2021

> **Note**
>
> Citrix ADM service connect is enabled by default, after you install or upgrade Citrix ADC or Citrix Gateway to release 13.0 build 61.xx and above. For more information see, Data governance and Citrix ADM service connect.

Citrix ADC 13.0 offers new and updated features with increased functionality. A comprehensive list of enhancements is listed in the release notes accompanying the release announcement. Read the release notes document before you upgrade your software.

This section provides information about **upgrading and downgrading a Citrix ADC appliance** (MPX and VPX) firmware **by using the Citrix ADC GUI or CLI**.

You can also **use Citrix ADM to upgrade a Citrix ADC appliance**. For more information, see:

- 10 ways Citrix ADM service supports easier Citrix ADC upgrades
- Use Citrix ADM service to upgrade Citrix ADC instances
- Use Citrix ADM software to upgrade Citrix ADC instances

For information about **upgrading a Citrix ADC SDX appliance**, see Single Bundle Upgrade.

Before you begin

November 22, 2021

Before you start the upgrade or downgrade process, make sure you check the following:

- Time allocated for upgrading Citrix ADC appliances. Follow your organization's change control procedure. Allocate twice as much time to perform the upgrades. Allocate enough time to upgrade each of the Citrix ADC appliance.
- Evaluate your organization's support agreement. Document appliance serial number, support agreement, and contacts details for support from Citrix Technical Support or the Citrix Authorized Partner.
- The licensing framework and types of licenses. A software edition upgrade might require new licenses, such as:
  - upgrading from the standard edition to the advanced edition, or
  - the standard edition to the Premium edition, or

  - the advanced edition to the Premium edition.

  Existing Citrix ADC licenses continue to work when you upgrade to version 13.0. For more information, see Licensing

- Check for New and deprecated commands, parameters, and SNMP OIDs.
- Check for Citrix ADC MPX Hardware and Software Compatibility Matrix.
- If the Citrix ADC Gateway logon page is customized, then make sure that the UI theme is set to default.
- If you are upgrading LOM, review the LOM Firmware Upgrade page.
- Download the Citrix ADC firmware from the Citrix ADC Downloads. For the detailed steps to download the Citrix ADC firmware, see the Download a Citrix ADC release package.
- Back up files. Perform a backup of the configuration file, customization file, certificates, monitor scripts, license files, and so on either manually or refer to the following documentation for backup using Citrix ADC CLI or GUI - Backup and restore.
  - Refer to the following list for additional common customized files for back up.
    * `/nsconfig/monitors/*.pl`
    * `/nsconfig/htmlinjection/*`
    * `/nsconfig/rc.netscaler`
  - Back up the customization folder. This is usually under `/var/customizations`. An example of customization is a logon page with a logo. After you have copied the customizations folder, you have to delete it from the Citrix ADC appliance, before you upgrade the appliance. Upgrading with customization in place might cause some issues.

  > **Important:**
  > Citrix highly recommends reviewing the above backup procedures. Have an action plan in the event the update does not complete on the Citrix ADC appliance.

- Verify that there is adequate space in the /var and /flash directory for the Citrix ADC appliance before performing an upgrade. The /var requires 5 GB of free space (1 GB for the upgrade bundle + 4 GB for the upgrade process)
  The /flash requires enough space to copy over the new kernel, which differs between 140MB to 160MB approximately, ensure that there is atleast 250 MB free space available.
  For more information on clearing out the disk space in /var see, How to free space on /var directory for logging issues with a Citrix ADC appliance.
  For more information on clearing out the disk spaces in /flash see, https://support.citrix.com/article/CTX133587.
- Validate the integrity of the Citrix ADC appliance. If you have a Citrix ADC hardware appliance, Citrix highly recommends running `fsck` for running a disk check and validating the integrity of the Citrix ADC hard drive. In the event of an error, reset the hard disk drive and repeating the disk

check command. If the error message reappears, contact Citrix Support to further investigate the issue.

  - Validate the disk integrity of the hard drive using fsck command. For more information, see CTX122845.
  - Validate the integrity of Citrix ADC appliance using the diagnostic bundles files and uploading the logs to Citrix Insight Service for analysis. For more information, see How to collect a technical support bundle.

- Check the Citrix ADC VPX Support matrix and usage guidelines.
- Check the FAQ section.
- It is a best practice to upgrade to one major release at a time. Do not upgrade directly to the latest version.

  For example, if the Citrix ADC appliance is on release 12.0, and you want to upgrade to release 13.0, you must upgrade the appliance to release 12.1 first, and then to release 13.0.
- Verify the upgrading procedures with a test environment.

For more information about prerequisites for upgrading or downgrading the Citrix ADC appliance, see these support articles:

- CTX220371: Must Read Articles Before and After Upgrading Citrix ADC

Upgrade considerations - SNMP configuration

September 14, 2021

The timeout parameter for an SNMP alarm is an internal option that has no impact on the alarm configuration.

Timeout parameter might appear in the SNMP alarm configurations in the running configuration (sh running) and the saved configuration (ns.conf) even if you have not made any changes to these SNMP alarm configurations.

On upgrading to a release build with the fix for the timeout setting issue, the SNMP configurations are erroneously reset to default values.

The following SNMP alarms (if configured) are impacted during an upgrade:

- APPFW-BUFFER-OVERFLOW
- APPFW-COOKIE
- APPFW-CSRF-TAG
- APPFW-DENY-URL

- APPFW-FIELD-CONSISTENCY
- APPFW-FIELD-FORMAT
- APPFW-POLICY-HIT
- APPFW-REFERER-HEADER
- APPFW-SAFE-COMMERCE
- APPFW-SAFE-OBJECT
- APPFW-SQL
- APPFW-START-URL
- APPFW-VIOLATIONS-TYPE
- APPFW-XML-ATTACHMENT
- APPFW-XML-DOS
- APPFW-XML-SCHEMA-COMPILE
- APPFW-XML-SOAP-FAULT
- APPFW-XML-SQL
- APPFW-XML-VALIDATION
- APPFW-XML-WSI
- APPFW-XML-XSS
- APPFW-XSS
- CLUSTER-BACKPLANE-HB-MISSING
- CLUSTER-NODE-HEALTH
- CLUSTER-NODE-QUORUM
- CLUSTER-VERSION-MISMATCH
- COMPACT-FLASH-ERRORS
- CONFIG-CHANGE
- CONFIG-SAVE
- HA-BAD-SECONDARY-STATE
- HA-NO-HEARTBEATS
- HA-SYNC-FAILURE
- HA-VERSION-MISMATCH
- HARD-DISK-DRIVE-ERRORS
- HA-STATE-CHANGE
- HA-STICKY-PRIMARY
- PORT-ALLOC-FAILED
- SYNFLOOD

These SNMP alarms configurations are impacted when you upgrade the Citrix ADC to the following release builds:

- Release 11.1 build 61.2 or later
- Release 12.0 build 61.0 or later
- Release 12.1 build 30.1 or later

- Release 13.0 build 51.4 or later

Example

Let's consider an example of CLUSTER-NODE-HEALTH SNMP alarm.

```
CLUSTER-NODE-HEALTH SNMP alarm is set up by using the Citrix ADC
    command line:

> set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -
    severity Major

> save config
<!--NeedCopy-->
```

This SNMP alarm configuration appears in the saved configuration file (`ns.conf`) as:

```
set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity
    Major -timeout 86400

<!--NeedCopy-->
```

During an upgrade to any of the release builds mentioned above, the following error appears in ns.log file:

```
May 23 09:14:46 <local0.err> ns nsconfigd: __init_config_filter(): (
    null) line 0: No such argument [-timeout]>> set snmp alarm CLUSTER-
    NODE-HEALTH -time 111 -state DISABLED -severity Major -timeout
    86400.
<!--NeedCopy-->
```

After the upgrade, the SNMP alarm configurations are reset to the default values.

Workaround

Use one of the following workarounds:

- Before the upgrade, remove the timeout setting from the SNMP configurations in the saved configuration file (ns.conf).
- After the upgrade, reconfigure the SNMP alarms without the timeout parameter.

Download a Citrix ADC release package

September 14, 2021

Complete the following steps to download a Citrix ADC release package:

1. Open Citrix ADC Downloads page in a Web browser.
2. On the Citrix ADC Downloads page, expand the **Citrix ADC release** that you want to update to.
3. Expand one of the appropriate categories, and click the Citrix ADC build link. For example, for downloading a version of Citrix ADC firmware, expand **Firmware**, and click the Citrix ADC build that you want to download.
4. On the selected Citrix ADC build page, expand the **Build** section, click **Download File** to download the Citrix ADC build package.

> **Note:**
> The checksum is provided to ensure that you match the downloaded build package with the actual package which is hosted on the website. Checksum is an important check to ensure that you have the correct bits.

Upgrade a Citrix ADC standalone appliance

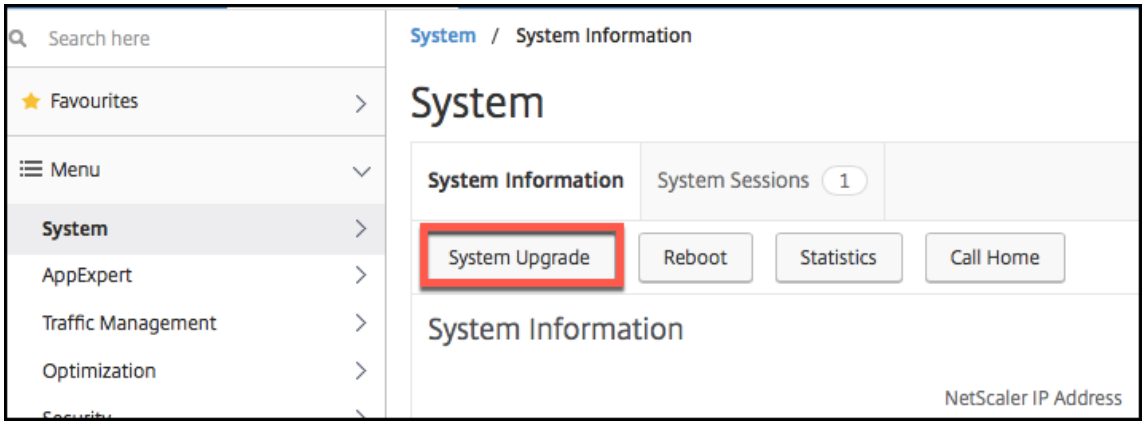
September 14, 2021

Before upgrading the system software, make sure that you read the Before you begin section and complete the prerequisites such as backing up the necessary files and downloading the Citrix ADC firmware.

Upgrade a Citrix ADC standalone appliance by using the GUI

Follow these steps to upgrade a standalone Citrix ADC to release 13.0 by using the GUI.

1. In a web browser, type the IP address of the Citrix ADC, for example `http://10.102.29.50`.
2. In User Name and Password, type the administrator credentials (nsroot/nsroot) and then click **Log On**.
3. From the GUI, click **System Upgrade**.



4. From the **Choose File** menu choose the appropriate option: **Local** or **Appliance**. If you want to use the Appliance option, the firmware needs to be uploaded to the Citrix ADC first. You can use any file transfer method such as WinSCP to upload the Citrix ADC firmware to the appliance.
5. Select the correct file and click **Upgrade**.
6. Follow the instructions to upgrade the software.
7. When prompted, select **Reboot**.

After the upgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Upgrade a Citrix ADC standalone appliance by using the CLI

Follow these steps to upgrade a standalone Citrix ADC to release 13.0 by using the CLI:

In the following procedure, `<release>` and `<releasenumber>` represent the release version you are upgrading to, and `<targetbuildnumber>` represents the build number that you are upgrading to. The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.

1. Use an SSH client, such as PuTTy, to open an SSH connection to the appliance.
2. Log on to the appliance by using the administrator credentials. Save the running configuration. At the prompt, type:

   ```
   save config
   ```

3. Switch to the shell prompt by running the following command:

   ```
   shell
   ```

4. Create a copy of the ns.conf file. At the shell prompt, type:

   - `cd /nsconfig`
   - `cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>`

You should backup the configuration file to another computer.

5. (Optional) If you have modified some of the following files in the /etc directory, and copied them to /nsconfig to maintain persistency, any updates that are pushed to the /etc directory during the upgrade might be lost:

   - ttys
   - resolv.conf
   - sshd_config
   - host.conf
   - newsyslog.conf
   - host.conf
   - httpd.conf
   - rc.conf
   - syslog.conf
   - crontab
   - monitrc

   To avoid losing these updates, create a `/var/nsconfig_backup` directory, and move the customized files to this directory. That is, move any files that you modified in /etc directory and copied to /nsconfig by running the following command:

   ```
   cp /nsconfig/<filename> /var/nsconfig_backup
   ```

   Example:

   ```
   cp /nsconfig/syslog.conf /var/nsconfig_backup
   ```

6. Create a location for the installation package. At the shell prompt type:

   - `cd /var/nsinstall`
   - `cd <releasenumber>`

     > **Note:**
     >
     > If the desired release number directory is not present, create one using the following command:
     >
     > `mkdir <releasenumber>`
     >
     > **Example:**
     >
     > `mkdir 13.0`

   - `mkdir build_<targetbuildnumber>`
   - `cd build_<targetbuildnumber>`

7. Copy the already downloaded Citrix ADC firmware to the build directory that you have created in the above step, by using any file transferring method such as WinSCP. See the Before You

Begin section for more information about downloading the Citrix ADC firmware.

8. Extract the contents of the installation package. Example:

   ```
   tar -xvzf build-13.0-37.2_nc_64.tgz
   ```

9. Run the installns script to install the new version of the system software.

   ```
   ./installns
   ```

10. When prompted, restart the Citrix ADC.
11. (Optionally) If you've created a copy of the ns.conf file in the Before You Begin section, do the following:

    a) Manually compare the files in /var/nsconfig_backup and /etc and make appropriate changes in /etc.
    b) To maintain persistency, move the updated files in /etc to /nsconfig.
    c) Restart the appliance to put the changes into effect.

Below is an example of Citrix ADC firmware upgrade.

```
login: nsroot

Password: nsroot

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

Done

> save config

> shell

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

root@NSnnn# cd /var/nsinstall

root@NSnnn# cd 13.0

root@NSnnn# mkdir build_43.1

root@NSnnn# cd build_43.1

root@NSnnn# ftp <FTP server IP address>

ftp> mget build-13.0-41.1_nc.tgz

```



```
ftp> bye

root@NSnnn# tar xzvf build-13.0-41.1_nc.tgz

root@NSnnn# ./installns

installns version (13.0-41.1) kernel (ns-13.0-41.1_nc.gz)

...

Copying ns-13.0-41.1_nc.gz to /flash/ns-13.0-41.1_nc.gz ...

...

Installation has completed.

Reboot NOW? [Y/N] Y
```

Upgrade a Citrix ADC standalone appliance by using NITRO API

To use NITRO API to upgrade or downgrade a Citrix ADC, see Automate Citrix ADC Upgrade and Downgrade with a Single API.

Verify entities status on the Citrix ADC appliance after upgrading

After the Citrix ADC appliance is upgraded, verify the status of the following entities:

- Virtual servers are in UP state
- Monitors are in UP state
- GSLB sites synchronise without any issues
- All certificates are present on the appliance
- All the licenses are present on the appliance

Check and install Citrix ADC 13.0 software update

Update the Citrix ADC software when an update is available, for better performance. A Citrix ADC update can include feature improvements, performance fixes, or enhancements. Make sure you read the release notes to see what fixes and enhancements are available in the update. To check and install a software update, do the following.

1. In the Citrix ADC home page, click **Check for Update** from the **nsroot** menu at the top right corner.

2. In the **Latest System Software Updates Available** page, check the available software update that you can install.
3. Click **Download** to download the installation package from the Citrix download website.
4. After you have downloaded the software package, install the update through either CLI or GUI procedure.

> **Note**
>
> The **Check for Update** link is accessible only if you log into the GUI through HTTP protocol and not through HTTPS protocol.

Related resources

The following resources provide related information about upgrading or downgrading a Citrix ADC appliance:

- Video tutorial - How to upgrade your Citrix ADC using CLI

Downgrade a Citrix ADC standalone appliance

September 14, 2021

You can downgrade to any earlier release on a standalone Citrix ADC by using the CLI or GUI.

> **Note:**
>
> Loss in configuration might occur when downgrading. Compare the configurations before and after the downgrade, and then manually reenter any missing entries.

Downgrade a Citrix ADC appliance by using the CLI

Follow the steps given below to downgrade a Citrix ADC standalone appliance running release 13.0 to an earlier release.

In this procedure, `<release>` and `<releasenumber>` represent the release version you are downgrading to, and `<targetbuildnumber>` represents the build number that you are downgrading to.

1. Open an SSH connection to the Citrix ADC by using an SSH client, such as PuTTY.
2. Log on to the Citrix ADC by using the administrator credentials. Save the running configuration. At the prompt, type:

   save config

3. Create a copy of the ns.conf file. At the shell prompt, type:

a) `cd /nsconfig`
b) `cp ns.conf ns.conf.NS<currentbuildnumber>`

You should backup a copy of the configuration file on another computer.

4. Copy the <releasenumber> configuration file (ns.conf.NS<releasenumber>) to ns.conf. At the shell prompt, type:

```
cp ns.conf.NS<releasenumber> ns.conf
<!--NeedCopy-->
```

> **Note:**
>
> `ns.conf.NS<releasenumber>` is the backup configuration file that is automatically created when the system software is upgraded from release version `<releasenumber>` to the current release version.
>
> There may be some loss in configuration when downgrading. After the appliance restarts, compare the configuration saved in step 3 with the running configuration, and make any adjustments for features and entities configured before the downgrade. Save the running configuration after making the changes.
>
> **Important:**
>
> If routing is enabled, perform step 5. Otherwise, skip to step 6.

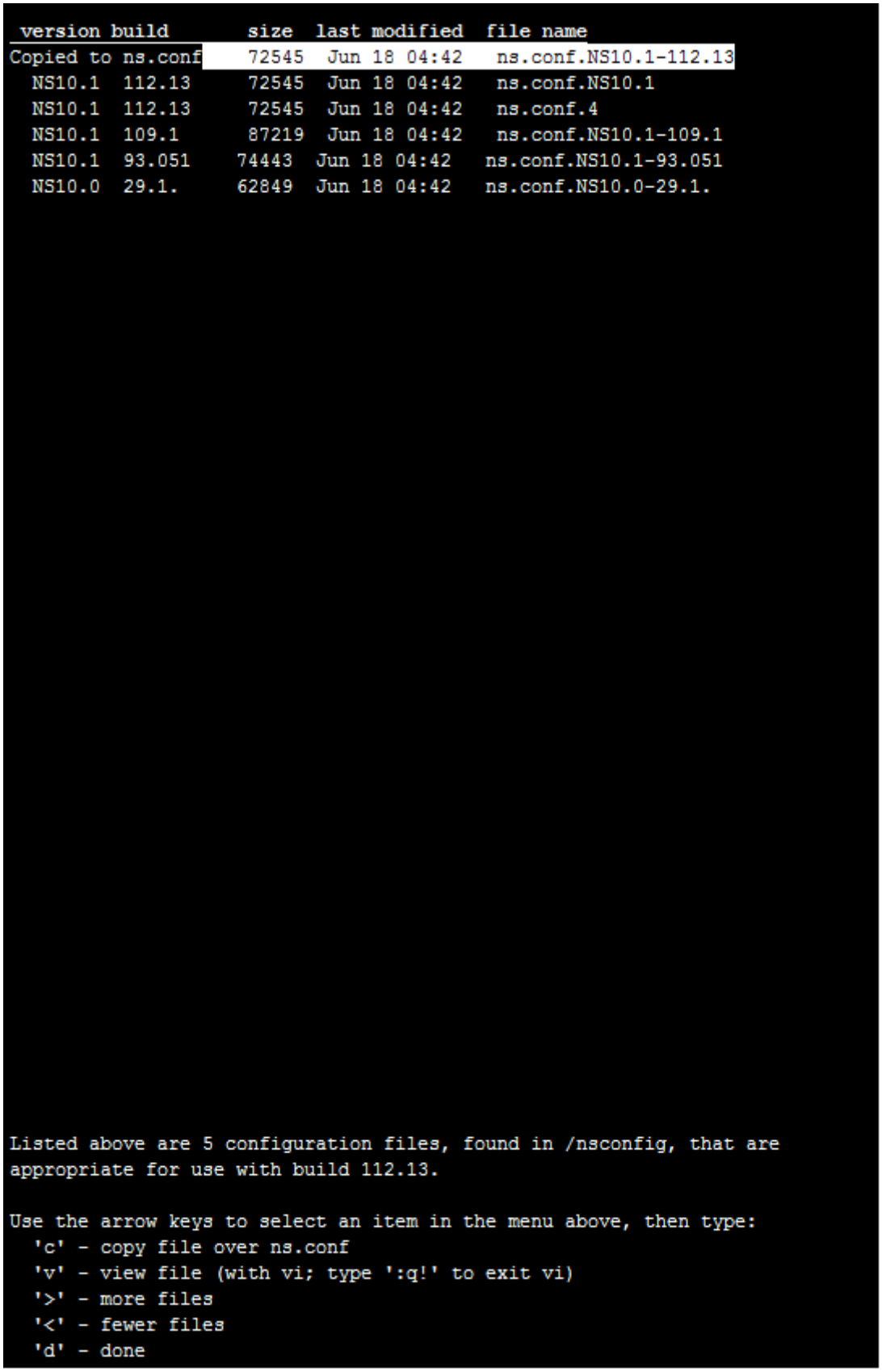
5. If routing is enabled, the ZebOS.conf file contains the configuration. At the shell prompt, type:

```
cd /nsconfig
cp ZebOS.conf ZebOS.conf.NS
cp ZebOS.conf.NS<targetreleasenumber> ZebOS.conf
<!--NeedCopy-->
```

6. Change directory to `/var/nsinstall/<releasenumber>nsinstall`, or create one if it does not exist.
7. Change directory to `build_<targetbuildnumber>`, or create one if it does not exist.
8. Download or copy the installation package (`build-<release>-<targetbuildnumber>.tgz`) to this directory and extract the contents of the installation package.
9. Run the `installns` script to install the new version of the system software. The script updates the `/etc` directory.

   If the configuration file for the build that you are downgrading to, exists on the appliance, you are prompted to load that configuration:

   Figure 1. Downgrade menu if configuration file exists



If the free space available on the flash drive is insufficient to install the new build, the Citrix ADC aborts the installation. Manually clean up the flash drive and restart the installation.

Example:

```
login: nsroot

Password: nsroot

Last login: Mon Apr 24 02:06:52 2017 from 10.102.29.9

Done

> save config

> shell

root@NSnnn# cp ns.conf.NS10.5 ns.conf

root@NSnnn# cd /var/nsinstall

root@NSnnn# mkdir 10.5nsinstall

root@NSnnn# cd 10.5nsinstall

root@NSnnn# mkdir build_57

root@NSnnn# cd build_57

root@NSnnn# ftp 10.102.1.1

ftp> mget build-10.5-57_nc.tgz

ftp> bye

root@NSnnn# tar -xzvf build-10.1-125_nc.tgz

root@NSnnn# ./installns

installns version (10.5-57) kernel (ns-10.5-57.gz)

...

...

```



```
...

Copying ns-10.5-57.gz to /flash/ns-10.5-57_nc.gz ...

Changing /flash/boot/loader.conf for ns-10.5-57 ...


Installation has completed.


Reboot NOW? [Y/N] Y
<!--NeedCopy-->
```

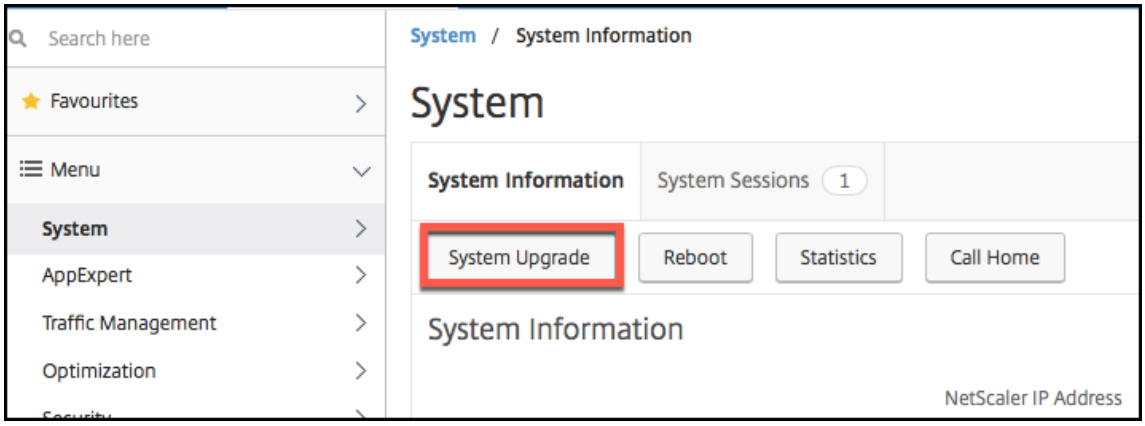
Downgrade a Citrix ADC appliance by using the GUI

You can use the upgrade wizard of the GUI to downgrade a Citrix ADC appliance running release 13.0 to an earlier release.

> **Notes:**
>
> You cannot downgrade a Citrix ADC appliance running release 13.0 directly to release 10.5 or earlier by using the GUI. Citrix recommends using the CLI for downgrading.
>
> Visit the Product Matrix site for more information on the Citrix ADC release lifecycle.
>
> It is a best practice to downgrade to one major release at a time.
>
> For example, if the Citrix ADC appliance is on release 13.0, and you want to downgrade to release 12.0, you must downgrade the appliance to release 12.1 first, and then to release 12.0.

Follow the steps given below to downgrade a Citrix ADC appliance running release 13.0 to an earlier release by using GUI.

1. In a web browser, type the IP address of the Citrix ADC, for example `http://10.102.29.50`.
2. In User Name and Password, type the administrator credentials and then click **Log On**.
3. From the GUI, click **System Upgrade**.



4. From the **Choose File** menu choose the appropriate option: **Local** or **Appliance**. If you want to use the Appliance option, the firmware must be uploaded to the Citrix ADC first. You can use any file transfer method such as WinSCP to upload the Citrix ADC firmware to the appliance.
5. Select the correct file and click **Upgrade**.
6. Follow the instructions to downgrade the software.
7. When prompted, select **Reboot**.

After the downgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Related resources

The following resources provide related information about upgrading or downgrading a Citrix ADC appliance:

- Video tutorial - How to upgrade your Citrix ADC using CLI

Upgrade a high availability pair

September 14, 2021

One of the requirements of Citrix ADC appliances in a high availability setup is to install the same Citrix ADC software release on both appliances of the setup. Therefore, when software on one appliance is upgraded, ensure that the software is upgraded on both the appliances.

You can follow the same procedure to upgrade a standalone appliance or each appliance in a high availability pair, although additional considerations apply to upgrading a high availability pair.

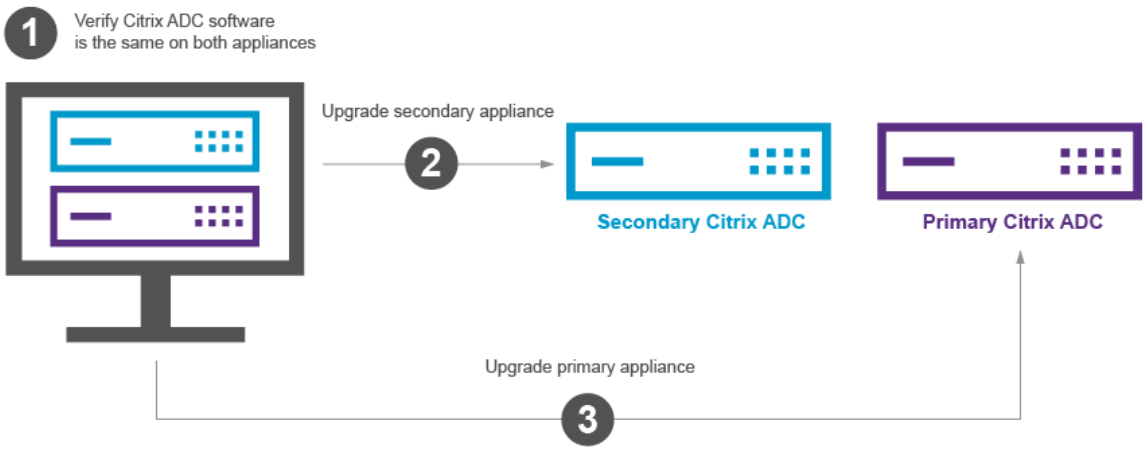
Before you start a Citrix ADC firmware upgrade on an HA pair, read the prerequisites mentioned in the Before you begin section. Also, you need to consider a few HA-specific points.

Points to note

- First upgrade the secondary node, and then the primary node. Upgrading software on the secondary appliance before the primary appliance ensures that the upgrade process is completed without any issues.
- If both the nodes in a high availability (HA) setup are running different Citrix ADC software releases, the following functionalities are disabled:
  - HA config synchronization
  - HA command propagation
  - HA synchronization of states services information
  - Connection mirroring (connection failover) of sessions
  - HA synchronization of persistence sessions information
- The above mentioned functionalities are disabled, if both the nodes in a high availability (HA) setup are running different builds of the same release but both the builds have different internal HA versions. The above mentioned functionalities works fine if both the nodes in a high availability (HA) setup are running different builds of the same release but both the builds have the same internal HA versions.

  Refer to the Points to note section of the release notes to check if the internal HA version has changed in Citrix ADC build.
- Synchronization of the files in the All mode of the Sync HA files command works successfully if the two nodes in an HA configuration are running different Citrix ADC software releases, or the two nodes are running different builds of the same release. For more information, see Synchronising Configuration Files in High Availability Setup.

Figure. Upgrade a high availability pair



You can upgrade using the Citrix ADC CLI or GUI.

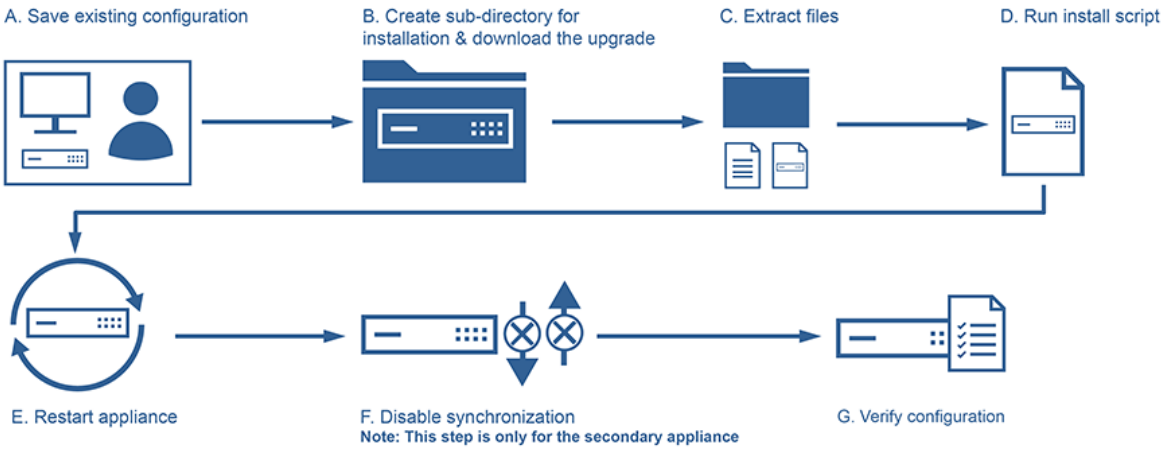
Upgrade a high availability pair by using the CLI

The upgrade process includes the following steps:

1. Upgrade software on the secondary appliance
2. Upgrade software on the primary appliance
3. Synchronize secondary appliance

Upgrade software on the secondary appliance

The following illustration depicts the procedure to upgrade software on the secondary appliance:



1. Log on to the secondary NetScaler appliance using an SSH utility, such as PuTTY and specifying the NetScaler IP (NSIP). Use the nsroot credentials to log on to the appliance.
2. From the command line interface of the appliance, type the following command to save the existing configuration: save config
3. Switch to the shell prompt.

```
login as: username
Using keyboard-interactive authentication.
Password:
Last login: Wed Jun 24 14:59:16 2015 from 10.252.252.65
Done
> shell
Copyright (c) 1992-20

<!--NeedCopy-->
```

4. Run the following command to change to the default installation directory: # cd /var/nsinstall
5. Run the following command to create a temporary subdirectory of the nsinstall directory: **# mkdir x_xnsinstall**

> Note: The text x_x is used to name the NetScaler version for future configurations. For example, the directory for the installation files of NetScaler 9.3 us called 9_3nsinstall. Do not use a period (.) in the folder name, it can cause failed upgrades.

6. Change to the **x_xnsinstall** directory.
7. Download the required installation package and documentation bundle, such as "ns-x.0-xx.x-doc.tgz," to the temporary directory created in Step 4.

   > **Note:**
   >
   > Some builds do not have a documentation bundle as it does not have to be installed.
   >
   > Click the **Documentation** tab from the GUI to access the documentation.

8. Before you run the install script, the files must be extracted and placed on the appliance. Use the following command to uncompress the bundle downloaded from Citrix website: **tar -zxvf ns-x.0-xx.x-doc.tgz**. The following is a quick explanation of the parameters used.

   x: Extract files

   v: Print the file names as they are extracted one by one

   z: The file is a "gzipped" file

   f: Use the following tar archive for the operation

9. Run the following command to install the downloaded software: # ./installns

   > Note: If the appliance does not have sufficient disk space to install the new kernel files, the installation process performs an automatic cleanup of the flash drive.

10. After the installation process is complete, the process prompts to restart the appliance. Press y to restart the appliance.
11. Log on to the appliance Command Line Interface using the nsroot credentials.
12. Run the following command from to display the state of the NetScaler appliance: **show ha node** The output of the preceding command should indicate that the appliance is a secondary node and synchronization is disabled.
13. Run the following command to perform a force failover and takeover as primary appliance: **force failover**

Here's a sample configuration in the new primary node.

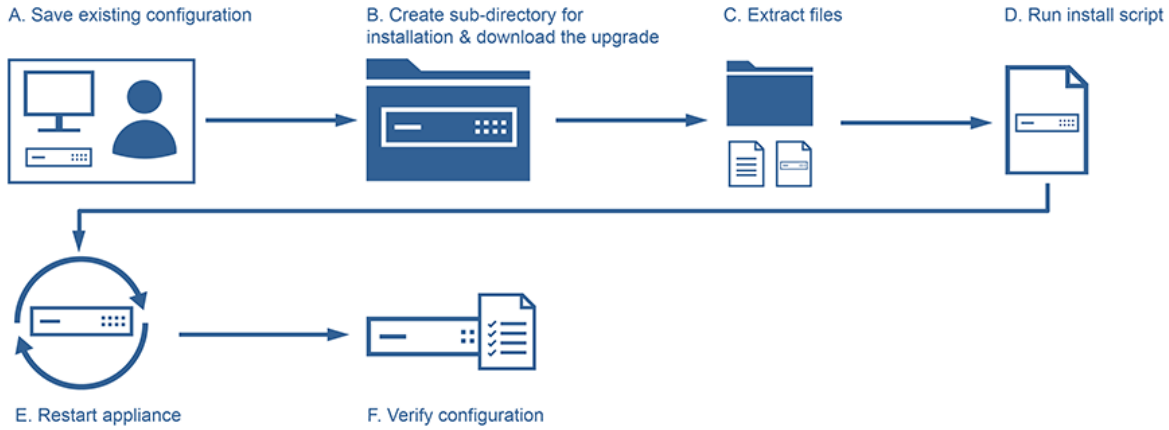
```
login: nsroot
Password: nsroot
Last login: Monday Apr  17 08:37:26 2017 from 10.102.29.9
Done
show ha node
```



```
        2 nodes:
1)      Node ID:      0
        IP:        10.0.4.2
        Node State: UP
        Master State: Primary
        ...
        Sync State: AUTO DISABLED
        Propagation: AUTO DISABLED
        ...
Done
<!--NeedCopy-->
```

Upgrade software on the primary appliance

The following illustration depicts the procedure to upgrade software on the primary appliance:



> Note: After completing the “Upgrade software on the secondary appliance” procedure, the original primary appliance is now a secondary appliance.

1. Log on to the secondary NetScaler appliance using an SSH utility, such as PuTTY. Use the nsroot credentials to log on to the appliance. Follow the same steps as mentioned in the above section to complete the installation process. We have to follow the same steps as mentioned in step 2 to step 9 in the previous section(Upgrade software of the secondary appliance)
2. After the installation process is complete, the process prompts to restart the appliance. Press y to restart the appliance.
3. Log on to the appliance Command Line Interface using the nsroot credentials.
4. Run the following command to display the state of the NetScaler appliance: **show ha node**. The output of the preceding command should indicate that the appliance is a secondary node and the status of the node state is marked as UP.

5. Run the following command to perform a force failover to ensure that the appliance is a primary appliance: **force failover**
6. Verify that the appliance is a primary appliance.

   Here's an example configuration of the new primary node and the new secondary node.

```
show ha node
    Node ID:      0
    IP:   10.0.4.11
    Node State: UP
    Master State: Primary
    ...
    ...
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    ...
    ...
    Local node information
    Critical Interfaces: 1/1
Done

Show ha node
    Node ID:      0
    IP:   10.0.4.2
    Node State: UP
    Master State: Secondary
    ..
    ..
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    . .
    . .
    Local node information:
    Critical Interfaces: 1/1
Done
<!--NeedCopy-->
```


Upgrade a high availability pair by using the GUI

Follow these steps to upgrade a Citrix ADC pair in a high availability setup, by using the ADC GUI. Consider an example of a high availability setup of Citrix ADC appliances CITRIX-ADC-A (primary) and CITRIX-ADC-B (secondary).

1. **Upgrade the secondary node**. Log on to the secondary node GUI using administrator credentials, and perform the upgrade as described in Upgrade a Citrix ADC standalone appliance by using the GUI.
2. **Force failover**. Perform a force failover on the secondary node using GUI as described in Forcing a node to fail over.

   After failover operation, the secondary node takes over as primary and the primary node becomes the new secondary node. After the failover operation in the example HA setup:

   - CITRIX-ADC-B becomes the new primary
   - CITRIX-ADC-A becomes the new secondary
3. **Upgrade the original primary node (new secondary node)**. Log on to the new secondary node GUI (CITRIX-ADC-A) and perform the upgrade as described in Upgrade a Citrix ADC standalone appliance by using the GUI.
4. **Force failover**. Perform a force failover on the new secondary node (CITRIX-ADC-A) using GUI as described in Forcing a node to fail over.

   After this second failover operation, the state of both the nodes return to the same state as before starting the HA upgrade operation. After the failover operation in the example HA setup:

   - CITRIX-ADC-A becomes primary
   - CITRIX-ADC-B becomes secondary
5. **Verify the upgrade process**. Log on to the GUI of both the nodes. Navigate to **System > High Availability**, on the details page, verify the HA state of both the nodes. Also, verify the upgraded release details displayed on the top pane of the GUI.

Related resources

The following resources provide related information about upgrading a Citrix ADC high availability setup:

- Video tutorial - How to upgrade your Citrix ADC HA pair using the GUI

In Service Software Upgrade support for high availability for performing zero downtime upgrade

September 14, 2021

During a regular upgrade process in a high availability setup, at some point, both nodes run different software builds. These two builds can have the same or different internal high availability version numbers.

If both the builds have different high availability version numbers, connection failover (even if it is enabled) for existing data connections is not supported. In other words, all existing data connections are lost, which leads to downtime.

To address this issue, in Service Software Upgrade (ISSU) can be used for high availability set-ups. ISSU introduces a migration functionality, which replaces the force failover operation step in the upgrade process. The migration functionality takes care of honoring the existing connections and includes the force failover operation.

After migration operation is performed, the new primary node always receives traffic (request and response) related to the existing connections but steers them to the old primary node. The old primary node processes the data traffic and then sends them directly to the destination.

How the enhanced ISSU works

The regular upgrade process in a high availability setup consist of the following sequential steps:

1. **Upgrade the secondary node**. This step includes software upgrade of the secondary node and restart of the node.
2. **Force Failover**. Running the force failover makes the upgraded secondary node to primary, and the primary node to secondary.
3. **Upgrade the new secondary node**. This step includes software upgrade of the new secondary node and restart of the node.

During the time frame between step 1 and step 3, both nodes run different software builds. These two builds can have the same or different internal high availability versions.

If both the builds have different high availability version numbers, connection failover (even if it is enabled) for existing data connections is not supported. In other words, all existing data connections are lost, which leads to downtime.

The ISSU upgrade process in a high availability setup consists of the following steps:

1. **Upgrade the secondary node**. This step includes software upgrade of the secondary node and restart of the node.

2. **ISSU migration operation**. The step includes the force failover operation and takes care of the existing connections. After you perform the migration operation, the new primary node always receives traffic (request and response) related to the existing connections but steers them to the old primary node through the configured SYNC VLAN in GRE tunnel. The old primary node processes the data traffic and then sends them directly to the destination. The ISSU migration operation is completed when all the existing connections are closed.
3. **Upgrade the new secondary node**. This step includes software upgrade of the new secondary node and restart of the node.

Before you begin

Before you start performing the ISSU process in a high availability setup, go through the following pre-requisites and limitations:

- Make sure the SYNC VLAN is configured on both the nodes of the high availability setup. For more information, see Restricting high availability synchronization traffic to a VLAN.
- ISSU is not supported in Microsoft Azure cloud because Microsoft Azure does not support GRE tunneling.
- High availability config propagation and synchronization do not work during ISSU.
- ISSU is not supported for IPv6 high availability setup.
- ISSU is not supported for following sessions:
  - Jumbo frames
  - IPv6 sessions
  - Large scale NAT (LSN)

Configuration steps

ISSU includes a migration feature, which replaces the force failover operation in the regular upgrade process of a high availability setup. The migration functionality takes care of honoring the existing connections and includes the force failover operation.

During the ISSU process of a high availability setup, you run the migration operation just after you upgraded the secondary node. You can perform the migration operation from either of the two nodes.

CLI Procedure

To perform the high availability migration operation by using the CLI:

At the command prompt type:


```
start ns migration
<!--NeedCopy-->
```

GUI Procedure

To perform the high availability migration operation by using the GUI:

Navigate to **System**, click **System Information** tab, click **Migration tab**, and then click **Start Migration**.

Display ISSU statistics

You can view the ISSU statistics for monitoring the current ISSU process in a high availability setup. The ISSU statistics displays the following information:

- Current status of ISSU migration operation
- Start time of the ISSU migration operation
- End time of the ISSU migration operation
- Start time of the ISSU rollback operation

You can view the ISSU statistics on either of HA nodes by using CLI or GUI.

CLI Procedure

To display the ISSU statistics by using the CLI:

At the command prompt type:

```
show ns migration
<!--NeedCopy-->
```

GUI Procedure

To display the ISSU statistics by using the GUI:

Navigate to **System**, click **System Information** tab, click **Migration tab**, and then click **Show Migration**.

Rollback of the ISSU process

High availability (HA) setups now support rollback of the In Service Software Upgrade (ISSU) process. The ISSU rollback feature is helpful if you observe that the HA setup during the ISSU migration operation is not stable, or is not performing at an optimum level as expected.

The ISSU rollback is applicable when the ISSU migration operation is in progress. The ISSU rollback does not work if ISSU migration operation is already completed. In other words, you must run the ISSU rollback operation when ISSU migration operation is in progress.

The ISSU rollback functions differently based on the state of the ISSU migration operation when the ISSU rollback operation is triggered:

- **Force failover has not yet happened during ISSU migration operation**. The ISSU rollback stops the ISSU migration operation, and removes any internal data related to the ISSU migration stored in both the nodes. The current primary node remains as primary node and continues to process data traffic related to existing and new connections.
- **Force failover has happened during ISSU migration operation**. If the high availability failover has happened during the ISSU migration operation, then the new primary node (say it is N1) processes traffic related to the new connections. The old primary node (new secondary node, say it is N2) processes traffic related to the old connections (existing connections before the ISSU migration operation).

  The ISSU rollback stops the ISSU migration operation and triggers a force failover. The new primary node (N2) now starts processing traffic related to the new connections. The new primary node (N2) also continues to processes traffic related to old connections (existing connections established before the ISSU migration operation). In other words, the existing connections established before the ISSU migration operation are not lost.

  The new secondary node (N1) removes all the existing connections (new connections created during the ISSU migration operation) and does not process any traffic. In other words, any existing connections that were established after the force failover of ISSU migration operation are lost forever.

Configuration steps

You can use Citrix ADC CLI or GUI to perform the ISSU rollback operation.

CLI Procedure

To perform the ISSU rollback operation by using the CLI:

At the command prompt type:

```
1 stop ns migration
2 <!--NeedCopy-->
```


GUI Procedure

To perform the ISSU rollback operation by using the GUI:

Navigate to **System**, click **System Information** tab, click **Migration tab**, and then click **Stop Migration**.

SNMP traps for In Service Software Upgrade process

The In Service Software Upgrade (ISSU) process for a high availability setup supports the following SNMP trap messages at the start and end of the ISSU migration operation.

SNMP Trap	Description
migrationStarted	This SNMP trap is generated and sent to the configured SNMP trap listeners when the ISSU migration operation starts.
migrationComplete	This SNMP trap is generated and sent to the configured SNMP trap listeners when the ISSU migration operation completes.

The primary node (before the start of the ISSU process) always generates these two SNMP traps and sends them to the configured SNMP trap listeners.

There are no SNMP alarms associated with the ISSU SNMP traps. In other words, these traps are generated irrespective of the any SNMP alarm. You only have to configure the trap SNMP listeners.

For more information on configuring SNMP trap listeners, see SNMP traps on Citrix ADC.

Downgrade a high availability pair

September 14, 2021

You can downgrade to any release on a high availability pair by using the command line interface. The GUI does not support the downgrade process.

To downgrade the system software on a Citrix ADC pair in a high availability pair, you need to downgrade the software first on the secondary node and then on the primary node. For instructions on downgrading each node separately, see Downgrade a Citrix ADC standalone appliance.

> **Important**
>
> Loss in configuration might occur when downgrading. You should compare the configurations before and after the downgrade, and then manually re-enter any missing entries.

Troubleshooting issues related to the installation, upgrade, and downgrade processes

September 14, 2021

If the appliance does not work as expected after you complete the installation, upgrade, or downgrade process, the first thing to do is to check for the most common causes of the problem.

Resources for troubleshooting

For best results, use the following resources to troubleshoot an issue related to installing, upgrading, or downgrading a Citrix ADC:

- The configuration files from the appliance. In case of a High Availability pair, the configuration files from both appliances.
- The following files from the appliance(s):
  - The relevant newnslog files.
  - The ns.log file.
  - The messages file.
- A network topology diagram.

Issues and resolutions

Following are the most common installation, upgrade, and downgrade issues, and tips for resolving them:

1. **Issue**

   Upgrading a Citrix ADC MPX appliance fails due to hardware and software incompatibility.

   Resolution

   See the Citrix ADC MPX hardware-software compatibility matrix and check whether the software release is supported on the Citrix ADC MPX hardware.

2. **Issue**

   Upgrading a Citrix ADC VPX appliance fails due to Citrix ADC VPX appliance and hypervisor incompatibility.

Resolution

See the Citrix ADC VPX appliance and hypervisor compatibility matrix and check whether the Citrix ADC VPX appliance model is supported on the hypervisor.

3. **Issue**

   Upgrading a Citrix ADC appliance fails due to hardware errors.

   Resolution

   Validate the integrity of the Citrix ADC appliance. If you have a Citrix ADC hardware appliance, Citrix recommends running `fsck` for running a disk check and validating the integrity of the Citrix ADC hard drive.

   For more information, see How to Verify the File System Integrity of a Citrix ADC appliance.

4. **Issue**

   Upgrading a Citrix ADC appliance by using the GUI stalls.

   Resolution

   Refresh the browser to check whether the upgrade is progressing or not.

5. **Issue**

   Upgrading a Citrix ADC appliance fails due to low Space in /var Directory

   Resolution

   Free up space on the /var directory. For more information, see
   Refer to How to free space on /var directory.

6. **Issue**

   The Citrix ADC is not accessible after the software downgrade

   Cause

   During the software downgrade process, if the configuration file of the existing release and build does not match the configuration file of the earlier release and build, the appliance cannot load the configuration, and the default IP address is assigned to the appliance.

   Resolution

   - Verify that the appliance is accessible from the console.
   - Verify the NSIP address and the routes on the appliance.
     - If the IP address has changed to the default 192.168.100.1 IP address, change the IP address as required.
     - Verify that the appliance is accessible.

7. **Issue**

   During an upgrade, if I run the command for synchronizing, the following message appears:

   Command failed on the secondary node but succeeded on the primary node.

   Resolution

   Do not run any dependent commands (set /unset /bind /unbind) when High Availability (HA) synchronization is in progress.

8. **Issue**

   During an upgrade process, traffic does not pass through the new primary node when you run the force failover command.

   Resolution

   - Check for problems with the network topology and the switch configurations.
   - Run the set L2param -garpreply ENABLED command to enable the GARP reply.
   - Try using virtual MAC if not already used.
   - Run the sendarp –a command from the primary node.

9. **Issue**

   After upgrading or downgrading a Citrix ADC appliance, connecting to the appliance fails through SSH.

   Resolution

   Perform the following operations in the Citrix ADC appliance:

   - Remove old or insecure host keys at `/nsconfig/ssh/ssh_host_*`.
   - Review the custom SSHD configuration at `/nsconfig/sshd_config` and check if it is still relevant and compatible. Rename or remove the custom SSHD configuration accordingly.
   - Cold reboot the Citrix ADC appliance

10. **Issue**

    In an HA pair, after you run the force HA failover command the devices keep rebooting. The secondary device does not come up after an upgrade.

    Resolution

    Check to see if the /var directory is full. If so, remove the old installation files. Run the df –h command to show the available disk space.

11. **Issue**

    After upgrading an HA pair, one of the nodes is listed as state UNKNOWN.

    Resolution

- Check to see if both nodes are running the same build. If the builds are not same and HA nodes have a version mismatch, some of the fields are shown as UNKNOWN when you run the show ha node command.
- Check to see if the secondary appliance is reachable.

12. **Issue**

    After you upgrade the Citrix ADC, the interface shows most of the load balancing virtual servers and services are DOWN.

    Resolution

    Verify that the SNIP address is active on the secondary appliance. Also, type the show service command to see if the service is running.

13. **Issue**

    After performing an upgrade, all virtual servers are down on the secondary appliance.

    Resolution

    Enable the HA state and HA synchronization by running the following commands:

    - set node hastate enable
    - set node hasync enable

    Disabling HA is not recommended.

14. **Issue**

    After performing a downgrade, the Citrix ADC does not boot up properly.

    Resolution

    Check to see if the correct license has been installed.

15. **Issue**

    In an HA pair, some features do not get synchronized after an upgrade is performed.

    Resolution

    Run the sync ha file misc command to synchronize the configurations files from the primary node to secondary node.

16. **Issue**

    During reboot, the following error message appears:

    One or more commands in ns.conf failedWhat should I do?

    Resolution

Make sure that no command in the ns.conf file exceeds the 255 byte limit. In commands that create policies that are too long for the 255-byte limit, you can use pattern sets to shorten the policies.

Example:

```
add cs policy p11 -rule 'HTTP.REQ.URL.ENDSWITH_ANY("
    ctx_file_extensions")'
Done
<!--NeedCopy-->
```

ctx_file_extensions is a default patset that covers a large number of extensions. In addition to the default pattern sets, you can create user-defined pattern sets. Add a patset by running the following command:

```
add patset <name>
<!--NeedCopy-->
```

Note: Patsets are supported only in release 9.3 or later.

17. **Issue**

    When upgrading a Citrix ADC VPX appliance, I am told to free up space in /var. What files do I remove?

    Resolution

    Remove the old installation files from /var/tmp/ directory. Also remove unwanted files from /flash.

18. **Issue**

    There is no connectivity to the graphical user interface (GUI) when you run the force HA failover command on the secondary appliance.

    Resolution

    Log on to the secondary appliance using the command line interface and enable the access to GUI by running the set ns ip <IP> -gui enabled command.

19. **Issue**

    After performing an upgrade, and when I click on any link on the GUI that has to load a java applet (Upgrade Wizard or license Wizard), the following error message appears: **GUI version does not match with the kernel version. Please close this instance, clear java plug-in cache and reopen**.

    Resolution

    - Log on to the Citrix ADC using the GUI.

- Navigate to Citrix ADC Gateway > Global Settings.
- Click Change Global Settings under Settings.
- In the details pane, under Client Experience, select Default from the UI theme list.
- Click OK.

20. **Issue**

    If upgrading a Citrix ADC appliance failed because of any reason, how to restore the appliance using the backed-up files?

    Resolution

    If the upgrade is unsuccessful then restore the appliance to the previous version of the Citrix ADC appliance using the backed-up files. For more information, see Backup and restore a Citrix ADC appliance.

    For more information on backup and restore of a Citrix ADC cluster setup, see Backup and restore of a cluster setup.

21. **Issue**

    If licenses are missing after a failed upgrade of a Citrix ADC appliance, how to resolve the issue?

    Resolution

    If any license goes missing or if you want to reallocate the licenses, refer to the following topic Licensing overview.

> **Note**
>
> These troubleshooting steps also apply to issues with configuration loss when downgrading the software across multiple releases.
>
> For any other issue, see the release notes, Knowledge Center articles, and FAQs.

FAQs

September 14, 2021

For answers to the questions that you might have about upgrading the Citrix ADC firmware, see the Installing, Upgrading, and Downgrading FAQs.

New and deprecated commands, parameters, and SNMP OIDs

September 14, 2021

This section lists the new and deprecated commands, parameters, and SNMP OIDs.

New commands

The following table lists the new commands in release 13.0.

Command group	Command
Authentication, authorization, and auditing	add aaa ssoprofile; rm aaa ssoprofile; show aaa ssoprofile; set aaa ssoprofile; add authentication citrixAuthAction; rm authentication citrixAuthAction; set authentication citrixAuthAction; unset authentication citrixAuthAction; show authentication citrixAuthAction; lock aaa user; set aaa otpparameter; show aaa otpparameter; add authentication emailAction; rm authentication emailAction; set authentication emailAction; show authentication emailAction; add authentication noAuthAction; rm authentication noAuthAction; set authentication noAuthAction; show authentication noAuthAction; add authentication captchaAction; rm authentication captchaAction; set authentication captchaAction; show authentication captchaAction; add authentication adfsProxyProfile; rm authentication adfsProxyProfile; set authentication adfsProxyProfile; show authentication adfsProxyProfile
AppFlow	bind appflow action; unbind appflow action
Application Firewall	stat rnat6 and stat MapBmr

Command group	Command
Content Inspection	add contentInspection profile; rm contentInspection profile; set contentInspection profile; show contentInspection profile; add contentInspection callout; rm contentInspection callout; set contentInspection callout; show contentInspection callout; count contentInspection callout; set contentInspection parameter; unset contentInspection parameter; show contentInspection parameter
LSN	add lsn appsattributes; rm lsn appsattributes; set lsn appsattributes; show lsn appsattributes
Network	add rnat; rename rnat; bind rnat; unbind rnat; rm rnat
SSL	add ssl caCertGroup; bind ssl caCertGroup; rm ssl caCertGroup; unbind ssl caCertGroup; show ssl caCertGroup
System	enable system autorestorefeature ; rm system restorepoint; show system restorepoint; migrate ns; show ns timezone; disable system autorestorefeature; create system restorepoint
URL Filtering	add urlfiltering Categorization; clear urlfiltering Categorization; show urlfiltering Categorization
VPN	add vpn urlPolicy; rm vpn urlPolicy; set vpn urlPolicy; show vpn urlPolicy; stat vpn urlPolicy; rename vpn urlPolicy; add vpn urlAction; rm vpn urlAction; set vpn urlAction; show vpn urlAction; rename vpn urlAction

New parameters

Command group: Authentication, authorization, and auditing

Command:

- set aaa parameter [-maxKBQuestions]
- unset aaa parameter [-maxKBQuestions]
- show aaa parameter [-maxKBQuestions]

Command group: Admin Partition

Command:

- add ns ip6 [-advertiseOnDefaultPartition]
- set ns ip6 [-advertiseOnDefaultPartition]
- show ns ip6 [-advertiseOnDefaultPartition]
- add ns ip [-advertiseOnDefaultPartition]
- set ns ip [-advertiseOnDefaultPartition]
- show ns ip [-advertiseOnDefaultPartition]

Command group: AppFlow

Command:

- bind appflow action [-analyticsProfile]
- unbind appflow action [-analyticsProfile]
- show appflow action [-analyticsProfile]
- set appflow param [-gxSessionReporting] [-usageRecordInterval] [-metrics] [-events [-auditlogs] [-observationPointId] [-distributedTracing] [-distTracingSamplingRate] [-tcpAttackCounterInterval]
- show appflow param [-observationPointId] [-subscriberIdObfuscationAlgo] [-gxSessionReporting] [-usageRecordInterval] [-metrics] [-events] [-auditlogs [-distributedTracing] [-distTracingSamplingRate] [-tcpAttackCounterInterval]

Command group: Application Firewall

Command:

- add appfw profile [-postBodyLimitSignature]
- add appfw profile [-rfcprofile]
- set appfw profile [-postBodyLimitSignature] [-rfcprofile]
- show appfw profile [-postBodyLimitSignature] [-rfcprofile]
- set appfw settings [-malformedReqAction]
- show appfw settings [-malformedReqAction]
- import appfw signatures [-preservedefactions]

Command group: Audit

Command:

- add audit syslogAction [-ContentInspectionLog]

- set audit syslogAction [-ContentInspectionLog]
- unset audit syslogAction [-ContentInspectionLog]
- show audit syslogAction [-ContentInspectionLog]
- add audit nslogAction [-ContentInspectionLog]
- set audit nslogAction [-ContentInspectionLog]
- unset audit nslogAction [-ContentInspectionLog]
- show audit nslogAction [-ContentInspectionLog]
- set audit syslogParams [-ContentInspectionLog]
- unset audit syslogParams [-ContentInspectionLog]
- show audit syslogParams [-ContentInspectionLog]
- set audit nslogParams [-ContentInspectionLog]
- unset audit nslogParams [-ContentInspectionLog]
- show audit nslogParams [-ContentInspectionLog]

Command group: Authentication

Command:

- add authentication ldapAction [-KBAttribute] [-alternateEmailAttr]
- set authentication ldapAction [-KBAttribute] [-alternateEmailAttr]
- show authentication ldapAction [-KBAttribute] [-alternateEmailAttr]
- add authentication tacacsAction [-Attributes]
- set authentication tacacsAction [-Attributes]
- show authentication tacacsAction [-Attributes]
- add authentication samlAction [-Attributes] [-storeSAMLResponse]
- set authentication samlAction [-Attributes] [-storeSAMLResponse]
- show authentication samlAction [-Attributes] [-storeSAMLResponse]
- add authentication vserver [-certkeyNames]
- set authentication vserver [-certkeyNames]
- show authentication vserver [-certkeyNames]- show authentication loginSchema [-feature]- - add authentication OAuthIDPProfile [-configservice] [-signatureAlg] [-Attributes][-sendPassword]
- set authentication OAuthIDPProfile [-configservice] [-signatureAlg] [-Attributes] [-sendPassword]
- show authentication OAuthIDPProfile [-configservice] [-signatureAlg] [-Attributes] [-sendPassword]
- add authentication pushService [-CertEndpoint] [-signingKeyName] [-signingKey] [-clientID] [-clientSecret] [-CustomerID] [-refreshInterval] [-trustService]
- set authentication pushService [-CertEndpoint] [-signingKeyName] [-signingKey] [-clientID] [-clientSecret] [-CustomerID] [-refreshInterval] [-trustService]
- show authentication pushService [-clientID] [-clientSecret] [-CustomerID] [-CertEndpoint] [-refreshInterval] [-pushServiceStatus] [-trustService] [-pushCloudServerStatus] [-signingKeyName] [-signingKey]
- show authentication adfsProxyProfile [-adfsTrustStatus]

Command group: Basic

Command:

- add server [-queryType]
- show server [-queryType] [-serviceGroupEntName2] [-svcitmPriority] [-svcitmActSvcs] [-svcitmBoundSvcs] [-weight]
- add service [-contentInspectionProfileName]
- set service[-contentInspectionProfileName]
- unset service [-contentInspectionProfileName]- show service [-contentInspectionProfileName]- bind serviceGroup [-nameServer] [-dbsTTL]
- show serviceGroup [-numOfCurConnections] [-numOfLastConnections] [-nameServer] [-dbsTTL] [-svcitmActSvcs] [-svcitmPriority] [-svcitmBoundSvcs]

Command group: Cache Redirection

Command:

- add cr vserver [-UseOriginIpPortForCache]
- set cr vserver [-UseOriginIpPortForCache]
- show cr vserver [-UseOriginIpPortForCache]

Command group: Clustering

Command:

- show cluster instance [-heterogeneousFlag]

Command group: Content Switching

Command:

- add cs vserver [-persistenceType] [-persistMask] [-v6persistmasklen] [-timeout] [-cookieName] [-persistenceBackup] [-backupPersistenceTimeout]
- set cs vserver [-persistenceType] [-persistMask] [-v6persistmasklen] [-timeout] [-cookieName] [-persistenceBackup] [-backupPersistenceTimeout]
- show cs vserver [-persistenceType] [-persistMask] [-v6persistmasklen] [-timeout] [-cookieName] [-persistenceBackup] [-backupPersistenceTimeout]

Command group: DNS

Command:

- set dns nameServer [-type]
- rm dns nameServer [-type]
- enable dns nameServer [-type]
- disable dns nameServer [-type]
- show dns action64
- set dns parameter [-maxUDPPacketSize]

- show dns parameter [-maxUDPPacketSize]

Command group: GSLB

Command:

- show gslb service [-gslbsvcHealth] [-glsbsvcHealthdescr]
- show gslb parameter [-builtin]

Command group: High Availability

Command:

- show HA node [-haSyncFailureReason]

Command group: ICA

Command:

- set ica parameter [-HDXInsightNonNSAP]
- show ica parameter [-HDXInsightNonNSAP]

Command group: Load Balancing

Command:

- add lb vserver [-adfsProxyProfile]
- set lb vserver [-adfsProxyProfile]
- unset lb vserver [-adfsProxyProfile]
- show lb vserver [-adfsProxyProfile]
- set lb parameter [-dbsTTL]
- show lb parameter [-dbsTTL]

Command group: LSN

Command:

- add lsn logprofile [-analyticsProfile] [-logSessDeletion]
- set lsn logprofile [-analyticsProfile] [-logSessDeletion]
- show lsn logprofile [-analyticsProfile] [-logSessDeletion]
- bind lsn appsprofile [-appsattributesname]
- unbind lsn appsprofile [-appsattributesname]
- show lsn appsprofile [-appsattributesname]

Command group: Network

Command:

- add route [-vlan]
- rm route [-vlan]
- add netProfile [-proxyProtocol] [-proxyProtocoltxversion]

- set netProfile [-proxyProtocol] [-proxyProtocoltxversion]
- show netProfile [-proxyProtocol] [-proxyProtocoltxversion]
- set rnat [-name]
- unset rnat [-name]
- add rnat [-name]
- rename rnat [-name] [-newName]
- bind rnat [-name] [-natIP]
- unbind rnat [-name] [-natIP]
- rm rnat [-name]
- show rnat [-name] [-stateflag]

Command group: Policy

Command:

- show policy expression
- show policy patset
- show policy urlset [-imported]
- import policy urlset [-subdomainExactMatch]

Command group: RDP

Command:

- add rdp clientprofile [-randomizeRDPFilename] [-rdpLinkAttribute]
- set rdp clientprofile [-randomizeRDPFilename] [-rdpLinkAttribute]
- show rdp clientprofile [-randomizeRDPFilename] [-rdpLinkAttribute]
- add rdp serverprofile [-rdpRedirection]
- set rdp serverprofile [-rdpRedirection]
- show rdp serverprofile [-rdpRedirection]

Command group: SSL

Command:

- create ssl ecdsaKey [-pkcs8]
- ssl certKey [-deletefromdevice]
- clear ssl certKey [-ocspstaplingCache]
- add ssl action [-caCertGrpName]
- show ssl action [-clientCertVerification] [-caCertGrpName]
- show ssl ocspResponder [-port]
- create ssl rsakey [-pkcs8]
- set ssl parameter [-ndcppComplianceCertCheck]
- show ssl parameter [-ndcppComplianceCertCheck]
- set ssl vserver [-preload] [-preload]

- show ssl dtlsProfile
- add ssl profile [-preload]
- set ssl profile [-preload]
- show ssl profile [-preload]

Command group: System

Command:

- create system backup [-useLocalTimezone] [-includekernel]
- show system backup [-useLocalTimezone]
- add ns tcpProfile [-taillossprobe]
- set ns tcpProfile [-taillossprobe]
- show ns tcpProfile [-taillossprobe]
- add ns icapProfile [-insertHTTPRequest]
- set ns icapProfile [-insertHTTPRequest]
- show ns icapProfile [-insertHTTPRequest]
- add ns httpProfile [-markTraceReqInval]
- set ns httpProfile [-markTraceReqInval]
- unset ns httpProfile [-markTraceReqInval]
- show ns httpProfile[- markTraceReqInval]
- save ns config all
- set ns param [-mgmthttpport] [-mgmthttpsport] [-proxyProtocol]
- show ns param [-mgmthttpport] [-mgmthttpsport] [-proxyProtocol]
- set ns httpParam [-ignoreConnectCodingScheme]
- show ns httpParam [-ignoreConnectCodingScheme]
- add ns icapProfile [-logAction]
- set ns icapProfile [-logAction]
- show ns icapProfile [-logAction]

Command group: Subscriber

Command:

- set subscriber gxInterface [-healthCheck] [-healthCheckTTL] [-cerRequestTimeout] [-negativeTTLLimitedSuccess] [-purgeSDBonGxFailure] [-gxReportingAvp1] [-gxReportingAvp1VendorId] [-gxReportingAvp1Type] [-gxReportingAvp2] [-gxReportingAvp2VendorId] [-gxReportingAvp2Type] [-gxReportingAvp3] [-gxReportingAvp3VendorId] [-gxReportingAvp3Type] [-gxReportingAvp4] [-gxReportingAvp4VendorId] [-gxReportingAvp4Type] [-gxReportingAvp5] [-gxReportingAvp5VendorId] [-gxReportingAvp5Type]
- show subscriber gxInterface [-healthCheck] [-healthCheckTTL] [-cerRequestTimeout] [-negativeTTLLimitedSuccess] [-purgeSDBonGxFailure] [-gxReportingAvp1] [-gxReportingAvp1VendorId] [-gxReportingAvp1Type] [-gxReportingAvp2] [-gxReportingAvp2VendorId] [-gxReportingAvp2Type] [-gxReportingAvp3] [-gxReportingAvp3VendorId] [-gxReportingAvp3Type] [-gxReportingAvp4]

[-gxReportingAvp4VendorId] [-gxReportingAvp4Type] [-gxReportingAvp5] [-gxReportingAvp5VendorId] [-gxReportingAvp5Type]

Command group: Traffic Management

Command:

- show tm sessionPolicy
- show tm sessionAction
- show tm global
- show tm sessionParameter [-tmsessionpolicyBindtype] [-tmsessionpolicyCount]

Command group: URL Filtering

Command:

- set urlfiltering parameter [-CloudHost] [-SeedDBPath]
- show urlfiltering parameter [-CloudHost] [-SeedDBPath]

Command group: VPN

Command:

- show vpn sessionPolicy
- add vpn sessionAction [-fqdnSpoofedIP]
- set vpn sessionAction [-fqdnSpoofedIP]
- show vpn sessionAction [-feature] [-fqdnSpoofedIP]
- show vpn clientlessAccessPolicy
- bind vpn global [-userDataEncryptionKey]
- unbind vpn global [-userDataEncryptionKey]
- show vpn global [-userDataEncryptionKey]
- set vpn parameter [-fqdnSpoofedIP] [-netmask]
- unset vpn parameter [-fqdnSpoofedIP]
- show vpn parameter [-fqdnSpoofedIP]

Deprecated parameters

Command group: AppFlow

Command:

- add appflow action [-MetricsLog]
- show appflow action [-MetricsLog]

Command group: LSN

Command:

- add lsn transportprofile [-stuntimeout]
- set lsn transportprofile [-stuntimeout]
- show lsn transportprofile [-stuntimeout]

Command group: Network

Command:

- clear rnat

Solutions for Telecom Service Providers

September 14, 2021

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

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

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

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

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

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

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

Large Scale NAT

September 14, 2021

> **Note**
>
> This feature is available with a Citrix ADC Advanced or Premium edition license.

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

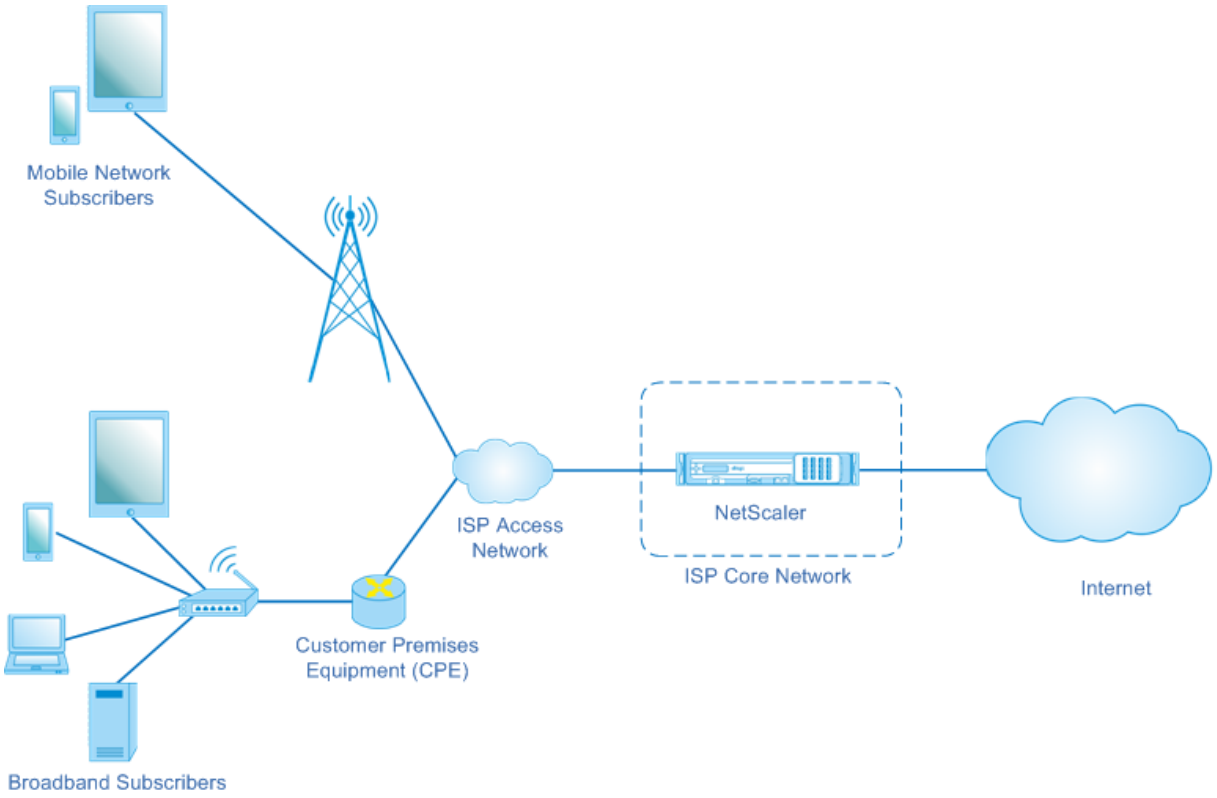
LSN translates private IPv4 addresses into public IPv4 addresses. It includes network address and port translation methods to aggregate many private IP addresses into fewer public IPv4 addresses. LSN is designed to handle NAT on a large scale. The Citrix ADC LSN feature is very useful for Internet Service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

LSN Architecture

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

Noncommercial subscribers, however, are typically behind customer-premises equipment (CPE), such as routers and modems, that also implements NAT. These two levels of NAT create the NAT444

model. Deploying a Citrix ADC appliance in an ISP's core network for LSN functionality is transparent to the subscribers and requires no configuration changes to subscribers or the CPEs.



Large Scale NAT

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

LSN Features Supported on Citrix ADC appliance

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

NAT Resource Allocation

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

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

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

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

IP Pooling

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

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

Reusing LSN Mappings

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

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

LSN Filtering

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

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

Quotas

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

- **Port quota.** The Citrix ADC appliance can limit the LSN NAT ports to be used at a time by each subscriber for a specified protocol. For example, you could limit each subscriber to a maximum of 500 TCP NAT ports. When the LSN NAT mappings for a subscriber reach the limit, the Citrix ADC appliance does not allocate additional NAT ports of the specified protocol to that subscriber.
- **Subscriber Session Limit.** The number of concurrent session for a subscriber can be more than it port quota. The Citrix ADC appliance can limit the LSN sessions allowed for each subscriber for a specified protocol. When the number of LSN sessions reaches the limit for a subscriber, the Citrix ADC appliance does not allow the subscriber to open additional sessions of the specified protocol.
- **Group Session Limit.** The Citrix ADC appliance can limit the total number of LSN sessions allowed for a subscriber group for a specified protocol. When the total number of LSN sessions reaches the limit for a group for a specified protocol, the Citrix ADC appliance does not allow any subscriber of the group to open additional sessions of the specified protocol. For example, You limit a group to a maximum of 10000 UDP sessions. When the total number of UDP sessions for this group reaches 10000, the Citrix ADC appliance does not allow any subscriber of the group to open additional UDP sessions.

Application Layer Gateways

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

The Citrix ADC appliance supports ALG for the following protocols:

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

Hairpin Support

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

Points to Consider before Configuring LSN

September 14, 2021

Consider the following points before configuring LSN on a Citrix ADC appliance:

- Make sure that you understand the different components of Large Scale NAT, described in RFCs 6888, 5382, 5508, and 4787.
- Endpoint independent mapping (EIM) and endpoint independent filtering (EIF) are disabled by default. These options must be enabled for proper functioning of VoIP and peer-to-peer (P2P) applications.
- **Logging LSN**: Following are the consideration points for logging LSN information:
  - Citrix recommends logging the LSN information on external log servers instead of on the Citrix ADC appliance. Logging on external servers facilitates optimal performance when the appliance creates large numbers of LSN log entries (in order of millions).
  - Citrix recomends using SYSLOG over TCP, or NSLOG. By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data.
  - The following limitations apply to SYSLOG over TCP:
    * The Syslog over TCP solution does not provide authentication, integrity check, and privacy.
    * The Citrix ADC appliance relies on the TCP protocol to provide confirmation of SYSLOG message delivery to external log servers.
- **High Availability**: Following are the consideration points for high availability of Citrix ADC appliances for LSN:
  - Citrix recommends configuring the LSN feature in a high availability deployment of two Citrix ADC appliances for uninterrupted and seamless operation of all LSN sessions.
  - In a high availability deployment, Citrix recommends:
    * Setting the SYNC VLAN parameter for dedicating a VLAN for all HA related communication.
    * Synchronizing the symmetric RSS key of the primary node to the secondary node for stateful synchronization of a large number of LSN mappings and sessions.
    * Binding the subnet of LSN IP addresses to a VLAN to avoid flooding of GARP broadcasts on all VLANs after a failover.

  - In a high availability deployment of Citrix ADC appliances, ALG-related sessions are not mirrored to the secondary appliance.
- **Application Layer Gateways (ALGs)**: Following are the consideration points related for ALGs on a Citrix ADC appliance:
  - The following are not supported for SIP ALG:
    - Multicast IP addresses
    - Encrypted SDP
    - SIP messages over TLS
    - FQDN translation in SIP messages
    - Authentication of SIP messages
    - Traffic domains, admin partitions, andCitrix ADC clusters.
    - SIP messages with multipart bodies.
  - The following are not supported for RTSP ALG:
    - Multicast RTSP sessions
    - RTSP session over UDP
    - Citrix ADC traffic domains, admin partitions, andCitrix ADC clusters
  - The Citrix ADC appliance does not support ALG for the IPSec protocol.
- If you disable the LSN feature when some LSN sessions exist on the Citrix ADC appliance, these sessions continue to exist for the duration of the configured timeout interval.
- LSN takes precedence over RNAT. If a packet from a specified LSN subscriber also matches a RNAT rule, the packet is translated according to the LSN configuration.
- Forwarding of packets related only to the LSN sessions is based on the Citrix ADC appliance's routing table.
- Unlike with subnet IP addresses, selection of an LSN NAT IP address for a subscriber's connection is not based on the routing entry for the destination IP address.
- For inbound packets, static LSN mappings take precedence over dynamic LSN mappings.
- For outbound packets, LSN application profiles take precedence over static mapping.
- When a large number of LSN sessions (> 1 million) exist on the Citrix ADC appliance, Citrix recommends displaying selected LSN sessions instead of all of them. In the command line interface or the configuration utility, use the selection parameters for showing LSN session operation.
- To reduce the amount of active memory allocated to the LSN feature, you must warm restart the Citrix ADC appliance after changing the configured-memory setting. Without a warm restart, you can only increase the amount of active memory.

Configuration Steps for LSN

September 14, 2021

Configuring LSN on a Citrix ADC appliance consists of the following tasks:

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

utility combines these two operations on a single screen.

6. **Create an LSN Group and bind LSN pools, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group.** An LSN group is an entity consisting of an LSN client, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. For Dynamic NAT, an LSN pool can be bound to multiple LSN groups. For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN group and binding LSN pools, LSN transport profiles, LSN application profiles to the LSN group. The configuration utility combines these two operations in a single screen.

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

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

LSN entities and bindings	Limit
Port ranges that can be bound to an LSN application profile	8

Configuration Using the Command Line Interface

To create an LSN client by using the command line interface

At the command prompt, type:

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn client
<!--NeedCopy-->
```

To create an LSN pool by using the command line interface

At the command prompt, type:

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

show lsn pool
<!--NeedCopy-->
```

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

At the command prompt, type:


```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

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

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

At the command prompt, type:

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

show lsn transportprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn appsprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```


To create an LSN group by using the command line interface

At the command prompt, type:

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

show lsn group
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn group
<!--NeedCopy-->
```

Configuration Using the Configuration Utility

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

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

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

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

To configure an LSN transport profile by using the configuration utility

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

To configure an LSN application profile by using the configuration utility

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

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

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn client
  - clientname

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

    This is a mandatory argument. Maximum Length: 127

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn client
  - clientname

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

    This is a mandatory argument. Maximum Length: 127

  - network

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

  - netmask

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

    Default value: 255.255.255.255

- td

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

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

  Default value: 0

  Minimum value: 0

  Maximum value: 4094

- aclname

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn pool
  - poolname

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

    This is a mandatory argument. Maximum Length: 127

  - nattype

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

    Available options function as follows:

    * **Deterministic**—Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The Citrix ADC appliance sequentially allocates NAT resources to these subscribers. The Citrix ADC appliance assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on

the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port.

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

  Possible values: DYNAMIC, DETERMINISTIC

  Default value: DYNAMIC

– portblockallocation

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

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

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

– portrealloctimeout

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

  * Deterministic NAT

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

  In these cases, ports are immediately reallocated.

  Default value: 0

  Maximum value: 600

- maxPortReallocTmq

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

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

  Default value: 65536

  Maximum value: 65536

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn pool
  - poolname

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

    This is a mandatory argument. Maximum Length: 127

  - lsnip

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

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn transportprofile
  - transportprofilename

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

    This is a mandatory argument. Maximum Length: 127
  - transportprotocol

    Protocol for which to set the LSN transport profile parameters.

    This is a mandatory argument.

    Possible values: TCP, UDP, ICMP
  - sessiontimeout

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

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

    Default value: 120

    Minimum value: 60
  - finrsttimeout

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

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

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

    Default value: 30
  - portquota

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

Default value: 0

Minimum value: 0

Maximum value: 65535

- sessionquota

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

  Default value: 0

  Minimum value: 0

  Maximum value: 65535

- portpreserveparity

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- portpreserverange

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

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

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- syncheck

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

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

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

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

  Possible values: ENABLED, DISABLED

  Default value: ENABLED

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn appsprofile
  - appsprofilename

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

    This is a mandatory argument. Maximum Length: 127

  - transportprotocol

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

    This is a mandatory argument.

    Possible values: TCP, UDP, ICMP

  - ippooling

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

    Available options function as follows:

    * **Paired**—The Citrix ADC appliance allocates the same NAT IP address for all sessions associated with the same subscriber. When all the ports of a NAT IP address are used

in LSN sessions (for same or multiple subscribers), the Citrix ADC appliance drops any new connection from the subscriber.

* **Random**—The Citrix ADC appliance allocates random NAT IP addresses, from the pool, for different sessions associated with the same subscriber.

This parameter is applicable to dynamic NAT allocation only.

Possible values: PAIRED, RANDOM

Default value: RANDOM

– mapping

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

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

Available options function as follows:

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

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

Default value: ADDRESS-PORT-DEPENDENT

– filtering

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

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

Available options function as follows:

* **ENDPOINT INDEPENDENT**—Filters out only packets not destined to the subscriber IP address and port X:x, regardless of the external host IP address and port source (Z:z). The Citrix ADC appliance forwards any packets destined to X:x. In other words, sending packets from the subscriber to any external IP address is sufficient to allow packets from any external hosts to the subscriber.

* **ADDRESS DEPENDENT**—Filters out packets not destined to subscriber IP address and port X:x. In addition, the appliance filters out packets from Y:y destined for the subscriber (X:x) if the client has not previously sent packets to Y:anyport (external port independent). In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external host's IP address.
* **ADDRESS PORT DEPENDENT** (the default)—Filters out packets not destined to subscriber IP address and port (X:x). In addition, the Citrix ADC appliance filters out packets from Y:y destined for the subscriber (X:x) if the subscriber has not previously sent packets to Y:y. In other words, receiving packets from a specific external host requires that the subscriber first send packets first to that external IP address and port.

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

Default value: ADDRESS-PORT-DEPENDENT

– tcpproxy

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

Possible values: ENABLED, DISABLED

Default value: DISABLED

– td

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

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

Default value: 65535

Maximum value: 65535

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn appsprofile
  - appsprofilename

    Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the

CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “lsn application profile1” or ‘lsn application profile1’).

This is a mandatory argument. Maximum Length: 127

- lsnport

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn group
  - groupname

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

    This is a mandatory argument. Maximum Length: 127

  - clientname

    Name of the LSN client entity to be associated with the LSN group. You can associate only one LSN client entity with an LSN group. You cannot remove this association or replace with another LSN client entity once the LSN group is created.

    This is a mandatory argument. Maximum Length: 127

  - nattype

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

    Available options function as follows:

    * **Deterministic**—Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The Citrix ADC appliance sequentially allocates NAT resources to these subscribers. The Citrix ADC appliance assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on

the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port.

* **Dynamic**—Allocate a random NAT IP address and a port from the LSN NAT pool for a subscriber's connection. If port block allocation is enabled (in LSN pool) and a port block size is specified (in the LSN group), the Citrix ADC appliance allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The appliance allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscribers connections) from the subscribers allocated port block, the appliance allocates a new random port block for the subscriber.

Possible values: DYNAMIC, DETERMINISTIC

Default value: DYNAMIC

– portblocksize

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

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

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

– logging

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

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

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

* NSIP address of the Citrix ADC appliance

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- sessionLogging

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

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

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- sessionSync

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

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

Possible values: ENABLED, DISABLED

Default value: ENABLED

- snmptraplimit

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

  Default value: 100

  Minimum value: 0

  Maximum value: 10000

- ftp

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

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

  Possible values: ENABLED, DISABLED

  Default value: ENABLED

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn group
  - groupname

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

    This is a mandatory argument. Maximum Length: 127

  - poolname

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

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

- transportprofilename

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

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

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

- appsprofilename

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

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

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

Sample LSN Configurations

September 14, 2021

The following are examples of configuring LSN through command line interface.

Create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, and default settings:

```
1 add lsn client LSN-CLIENT-1
2
3 Done
4
5 bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
```



```
6
7  Done
8
9  add lsn pool LSN-POOL-1
10
11 Done
12
13 bind lsn pool LSN-POOL-1 203.0.113.3
14
15 Done
16
17 add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1
18
19 Done
20
21 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
22
23 Done
24 <!--NeedCopy-->
```

Create an LSN configuration with an extended ACL for identifying LSN subscribers:

```
1  add ns acl LSN-ACL-2 ALLOW -srcIP 192.0.2.10-192.0.2.20
2
3  Done
4
5  apply acls
6
7  Done
8
9  add lsn client LSN-CLIENT-2
10
11 Done
12
13 bind lsn client LSN-CLIENT-2  - aclname LSN-ACL-2
14
15 Done
16
17 add lsn pool LSN-POOL-2
18
19 Done
20
21 bind lsn pool LSN-POOL-2 203.0.113.5-203.0.113.10
22
23 Done
```



```
24
25 add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2
26
27 Done
28
29 bind lsn group LSN-GROUP-2 -poolname LSN-POOL-2
30
31 Done
32 <!--NeedCopy-->
```

Create an LSN configuration with endpoint-independent mapping for HTTP protocol (port 80) and address-port dependent mapping for SSH protocol (port 22). Also, restrict each subscriber to use a maximum of 1000 NAT ports for TCP protocol and 100 NAT ports for UDP protocol. Restrict each subscriber to have a maximum of 2000 concurrent sessions for TCP protocol. Restrict the group to have a maximum of 30000 concurrent sessions for TCP protocol:

```
1  add lsn client LSN-CLIENT-3
2
3  Done
4
5  bind lsn client LSN-CLIENT-3 -network 192.0.3.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-3
10
11 Done
12
13 bind lsn pool LSN-POOL-3 203.0.113.11
14
15 Done
16
17 add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-3
18
19 Done
20
21 bind lsn group LSN-GROUP-3 -poolname LSN-POOL-3
22
23 Done
24
25 add lsn appsprofile LSN-APPS-HTTPPROFILE-3 TCP -mapping ENDPOINT-
      INDEPENDENT
26
27 Done
```



```
28
29 bind lsn appsprofile LSN-APPS-HTTPPROFILE-3 80
30
31 Done
32
33 bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-HTTPPROFILE
       -3
34
35 Done
36
37 add lsn appsprofile LSN-APPS-SSHPROFILE-3 TCP -mapping ADDRESS-PORT-
       DEPENDENT
38
39 Done
40
41 bind lsn appsprofile LSN-APPS-SSHPROFILE-3 22
42
43 Done
44
45 bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-SSHPROFILE
       -3
46
47 Done
48
49 add lsn transportprofile LSN-TRANS-PROFILE-TCP-3 TCP -portquota 1000 -
       sessionquota 2000 -groupSessionLimit 30000
50
51 Done
52
53 bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-TCP
       -3
54
55 Done
56
57 add lsn transportprofile LSN-TRANS-PROFILE-UDP-3 UDP -portquota 100
58
59 Done
60
61 bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-UDP
       -3
62
63 Done
64 <!--NeedCopy-->
```

Create an LSN configuration for a large set of subscribers:


```
add lsn client LSN-CLIENT-4

Done

bind lsn client LSN-CLIENT-4 -network 192.0.4.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.5.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.6.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.7.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.8.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-4

Done

bind lsn pool LSN-POOL-4 203.0.113.30-203.0.113.40

Done

bind lsn pool LSN-POOL-4 203.0.113.45-203.0.113.50

Done

bind lsn pool LSN-POOL-4 203.0.113.55-203.0.113.60

Done

add lsn group LSN-GROUP-4 -clientname LSN-CLIENT-4

Done

```



```
bind lsn group LSN-GROUP-4 -poolname LSN-POOL-4

Done

add lsn appsprofile LSN-APPS-WELLKNOWNPROFILE-4 TCP -mapping ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 1- 1023

Done

bind lsn group LSN-GROUP-4 -applicationprofilename LSN-APPS-WELLKNOWN-PORTS-PROFILE-4

Done
<!--NeedCopy-->
```

Create an LSN configuration with sharing of NAT resources among multiple LSN groups. In this example, LSN pool LSN-POOL-5 is shared with LSN groups LSN-GROUP-5 and LSN-GROUP-6:

```
add lsn client LSN-CLIENT-5

Done

bind lsn client LSN-CLIENT-5 -network 192.0.15.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-5

Done

bind lsn pool LSN-POOL-5 203.0.113.12-203.0.113.14

Done

add lsn group LSN-GROUP-5 -clientname LSN-CLIENT-5

Done

bind lsn group LSN-GROUP-5 -poolname LSN-POOL-5

Done
```



```
24
25 add lsn client LSN-CLIENT-6
26
27 Done
28
29 bind lsn client LSN-CLIENT-6 -network 192.0.16.0 -netmask 255.255.255.0
30
31 Done
32
33 add lsn pool LSN-POOL-6
34
35 Done
36
37 bind lsn pool LSN-POOL-6 203.0.113.15-203.0.113.18
38
39 Done
40
41 add lsn group LSN-GROUP-6 -clientname LSN-CLIENT-6
42
43 Done
44
45 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-6
46
47 Done
48
49 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-5
50
51 Done
52 <!--NeedCopy-->
```

Create an LSN configuration with deterministic NAT resource allocation:

```
1  add lsn client LSN-CLIENT-7
2
3  Done
4
5  bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
10
11 Done
12
13 bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
```



```
14
15 Done
16
17 add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
      DETERMINISTIC -portblocksize 1024
18
19 Done
20
21 bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
22
23 Done
24 <!--NeedCopy-->
```

Create an LSN configuration with multiple subscriber networks having the same network address but each network belonging to a different traffic domain. Also, restrict the outbound traffic related to HTTP protocol (port 80), sending it through a particular traffic domain (td 5):

```
1  add lsn client LSN-CLIENT-8
2
3  Done
4
5  bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 1
6
7  Done
8
9  bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 2
10
11 Done
12
13 bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 3
14
15 Done
16
17 add lsn pool LSN-POOL-8
18
19 Done
20
21 bind lsn pool LSN-POOL-8 203.0.113.80-203.0.113.86
22
23 Done
24
25 add lsn group LSN-GROUP-8 -clientname LSN-CLIENT-8
```



```
26
27 Done
28
29 bind lsn group LSN-GROUP-8 -poolname LSN-POOL-8
30
31 Done
32
33 add lsn appsprofile LSN-APPS-HTTP-PROFILE-8 TCP -td 5
34
35 Done
36
37 bind lsn appsprofile LSN-APPS-HTTP-PROFILE-8 80
38
39 Done
40
41 bind lsn group LSN-GROUP-8 -applicationprofilename LSN-APPS-HTTP-
      PROFILE-8
42
43 Done
44 <!--NeedCopy-->
```

Create an LSN configuration that restricts the outbound traffic of a specific protocol (TCP), sending it through a particular traffic domain (td 5). With endpoint-independent filtering, receive inbound traffic related to this protocol (TCP) on any traffic domain:

```
1  add lsn client LSN-CLIENT-9
2
3  Done
4
5  bind lsn client LSN-CLIENT-9 -network 192.0.9.0 -netmask 255.255.255.0
       -td 1
6
7  Done
8
9  add lsn pool LSN-POOL-9
10
11 Done
12
13 bind lsn pool LSN-POOL-9 203.0.113.90
14
15 Done
16
17 add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
18
19 Done
```



```
20
21 bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
22
23 Done
24
25 add lsn appsprofile LSN-APPS-PROFILE-9 TCP -filtering ENDPOINT-
      INDEPENDENT -td 5
26
27 Done
28
29 bind lsn group LSN-GROUP-9 -approfile LSN-APPS-PROFILE-9
30
31 Done
32 <!--NeedCopy-->
```

Create an LSN configuration that restricts outbound HTTP (port 80) traffic, sending it through a particular traffic domain (td 10). With address-dependent filtering, receive inbound traffic related to this protocol (HTTP) on the specified traffic domain (td 10):

```
1  add lsn client LSN-CLIENT-10
2
3  Done
4
5  bind lsn client LSN-CLIENT-10 -network 192.0.10.0 -netmask
      255.255.255.0 -td 1
6
7  Done
8
9  add lsn pool LSN-POOL-10
10
11 Done
12
13 bind lsn pool LSN-POOL-10 203.0.113.100
14
15 Done
16
17 add lsn group LSN-GROUP-10 -clientname LSN-CLIENT-10
18
19 Done
20
21 bind lsn group LSN-GROUP-10 -poolname LSN-POOL-10
22
23 Done
24
25 add lsn appsprofile LSN-APPS-PROFILE-10 TCP -mapping ENDPOINT -
```



```
      INDEPENDENT -filtering ADDRESS-DEPENDENT -td 10
26
27 Done
28
29 bind lsn appsprofile LSN-APPS-PROFILE-10 80
30
31 Done
32
33 bind lsn group LSN-GROUP-10 -approfile LSN-APPS-PROFILE-10
34
35 Done
36 <!--NeedCopy-->
```

Configuring Static LSN Maps

September 14, 2021

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

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

At the command prompt, type:

```
1 add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
     <positive_integer>] [<natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd
      <positive_integer>]]
2 -  show lsn static
3 <!--NeedCopy-->
```

To create a static LSN mapping by using the configuration utility

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

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn static name

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

transportprotocol

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

subscrIP

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

subscrPort

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

td

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

natIP

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

natPort

NAT port for this LSN mapping entry.

destIP

Destination IP address for the LSN mapping entry.

dsttd

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

Wildcard Port Static Maps

A static mapping entry is usually a one-to-one LSN mapping between a subscriber IP address:port and a NAT IP address:port. A one-to-one static LSN mapping entry exposes only one port of the subscriber to the Internet.

Some situations might require exposing all ports (64K) of a subscriber to the Internet (for example, a server hosted on an internal network and running a different service on each port). To make these internal services accessible through the Internet, you have to expose all the ports of the server to the Internet.

One way to meet this requirement is to add 64K one-to-one static mapping entries, one mapping entry for each port. Creating 64K entries is very cumbersome and a big task. Also, this large number of configuration entries might lead to performance issues in the Citrix ADC appliance.

Another simple method is to use wildcard ports in a static mapping entry. You just need to create one static mapping entry with NAT-port and subscriber-port parameters set to the wildcard character (*), and the protocol parameter set to ALL, to expose all the ports of a subscriber to the Internet. For a subscriber's inbound or outbound connections matching a wildcard static mapping entry, the subscriber's port does not change after the NAT operation.

When a subscriber-initiated connection to the Internet matches a wildcard static mapping entry, the Citrix ADC appliance assigns a NAT port that has the same number as the subscriber port from which the connection is initiated. Similarly, an Internet host gets connected to a subscriber's port by connecting to the NAT port that has the same number as the subscriber's port.

Configuring the Citrix ADC appliance to Provide Access to All Ports of an IPv4 Subscriber

To configure the Citrix ADC appliance to provide access to all ports of an IPv4 subscriber, create a wildcard static map with the following mandatory parameter settings:

- Protocol=ALL
- Subscriber port = *
- NAT port = *

In a wildcard static map, unlike in a one-to-one static map, setting the NAT IP parameter is mandatory. Also, the NAT IP address assigned to a wildcard static map cannot be used for any other subscribers.

To create a wildcard static map by using the command line interface

At the command prompt, type:

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
   positive_integer>] [-destIP <ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

Sample Configuration

In the following sample configuration of a wildcard static map, all ports of a subscriber whose IP address is 192.0.2.10 are made accessible through NAT IP 203.0.11.33.

Sample configuration:

```
add lsn static NAT44-WILDCARD-STATIC-1 ALL  192.0.2.10 * 203.0.113.33 *

Done
<!--NeedCopy-->
```

Configuring Application Layer Gateways

September 14, 2021

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

The Citrix ADC appliance supports ALG for the following protocols:

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

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

September 14, 2021

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

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

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

Sample LSN Configuration for FTP ALG:
In the following sample LSN configuration, FTP ALG is enabled for subscribers that have IP address in the range 192.0.2.30-192.0.2.100.

```
add ns acl LSN-ACL-1 ALLOW -srcIP 192.0.2.30-192.0.2.100

Done

apply acls

Done

add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1  -aclname LSN-ACL

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.10

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -FTP ENABLED

Done

```



```
bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```

Sample LSN Configuration for TFTP ALG:

In the following sample LSN configuration, endpoint-independent mapping and endpoint-independent filtering are enabled for TFTP protocol (UDP port 69). The Citrix ADC appliance automatically enables TFTP ALG for this LSN configuration.

```
add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2 -network 198.51.100.0 -netmask
   255.255.255.0

Done

add lsn pool LSN-POOL-2

Done

bind lsn pool LSN-POOL-2 203.0.113.10-203.0.113.11

Done

add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

bind lsn group LSN-GROUP-2 -poolname pool1 LSN-POOL-2

Done

add lsn appsprofile LSNAPPSPROFILE-TFTP-2 UDP -mapping ENDPOINT-
   INDEPENDENT - filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSNAPPSPROFILE-TFTP-2 69

Done

```



```
33 bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TFTP
       -2
34
35 Done
36 <!--NeedCopy-->
```

Application Layer Gateway for PPTP Protocol

September 14, 2021

The Citrix ADC appliance supports Application Layer Gateways (ALGs) for the Point-to-Point Tunneling Protocol (PPTP).

PPTP is a network protocol that enables secure transfer of data from a remote client to an enterprise server by creating a tunnel across TCP/IP-based data networks. PPTP encapsulates PPP packets into IP packets for transmission over the Internet. PPTP establishes a tunnel for each communicating PPTP network server (PNS)-PPTP Access Concentrator (PAC) pair. After the tunnel is set up, enhanced generic routing encapsulation (GRE) is used to exchange PPP packets. A call ID in the GRE header indicates the session to which a particular PPP packet belongs.

The Citrix ADC appliance recognizes PPTP packets that arrive on the default TCP port, 1723. The appliance parses PPTP control packets, translates the call ID, and assigns a NAT IP address. For two-way data communication between the client and server, the Citrix ADC appliance creates an LSN session entry based on the server call ID, and an LSN session based on the client call ID. The appliance then parses the GRE data packets and translates call IDs on the basis of the two LSN session entries.

For PPTP protocol, the Citrix ADC appliance also includes timeout setting for any idle PPTP LSN sessions. If a PPTP LSN session is idle for a time that exceeds the timeout setting, the Citrix ADC appliance removes the session.

Limitations:

The following are the limitations of PPTP ALG on a Citrix ADC appliance:

- PPTP ALG is not supported for hairpin LSN flow.
- PPTP ALG is not supported to work with any RNAT configuration.
- PPTP ALG is not supported in Citrix ADC clusters.

Configuring PPTP ALG

Configuring PPTP ALG on the Citrix ADC appliance consist of the following tasks:

- Create an LSN configuration and enable PPTP ALG on it. In an LSN configuration, the LSN group includes the PPTP ALG setting. For instructions on creating an LSN configuration, see Configuration Steps for LSN.
- (Optional) Set the global timeout for idle PPTP LSN sessions.

To enable PPTP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-pptp ( ENABLED |
   DISABLED )]

show lsn group
<!--NeedCopy-->
```

To set the global timeout for idle PPTP LSN sessions by using the CLI

At the command prompt, type:

```
set appAlgParam -pptpGreIdleTimeout <positive_integer>

show appAlgParam
<!--NeedCopy-->
```

Example:

In the following sample LSN configuration, PPTP ALG is enabled for subscribers in the 192.0.2.0/24 network.

Also idle PPTP LSN session timeout is set to 200 secs.

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3
```



```
14
15 Done
16
17 add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -pptp ENABLED
18
19 Done
20
21 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
22
23 Done
24
25 set appAlgParam -pptpGreIdleTimeout 200
26
27 Done
28 <!--NeedCopy-->
```

Application Layer Gateway for SIP Protocol

September 14, 2021

Using Large Scale NAT (LSN) with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media.

SIP ALG adheres to the following RFCs:

- RFC 3261
- RFC 3581
- RFC 4566
- RFC 4475

> **Note**
>
> SIP ALG is supported in a Citrix ADC standalone appliance, in a Citrix ADC high availability setup, as well as in a Citrix ADC cluster setup.

How SIP ALG Works

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

For an outgoing message, the private IP address and port number of the SIP client are replaced with the Citrix ADC owned public IP address and port number, called the *LSN pool IP address and port number*, specified during LSN configuration. For an incoming message, the LSN pool IP address and the port number are replaced with the private address of the client. If the message contains any public IP addresses, the Citrix ADC SIP ALG retains them. Also, a pinhole is created on the:

- LSN pool IP address and port on behalf of the private client, so that the messages that arrive at this IP address and port from the public network are treated as SIP messages.
- Public IP address and port on behalf of the public clients, so that the messages that arrive at this IP address and port from the private network are treated as SIP messages.

When a SIP message is sent out across the network, the SIP Application Layer Gateway (ALG) collects information from the message and translates the IP addresses in the following headers into LSN pool IP addresses:

- Via
- Contact
- Route
- Record-Route

In the following sample SIP request message, LSN replaces the IP addresses in the header fields to hide them from the outside network.

```
1 INVITE adam@10.102.185.156 SIP/2.0 Via: SIP/2.0/UDP 192.170.1.161:62914
      From: eve@10.120.210.3 To: adam@10.102.185.156 Call-ID: a12abcde@10
     .120.210.3 Contact: adam@10.102.185.156 Route: <sip:netscreen@10
     .150.20.3:5060> Record-Route: <sip:netscreen@10.150.20.3:5060>
2 <!--NeedCopy-->
```

When a message containing SDP information arrives, the SIP ALG collects information from the message and translates the IP addresses in the following fields into LSN pool IP addresses and port numbers:

- c= (connection information)

  This field can appear at the session or media level. It appears in the following format:

  c=<network-type><address-type><connection-address>

  If the destination IP address is a unicast IP address, the SIP ALG creates pinholes by using the IP address and port numbers specified in the m= field.

- m= (media announcement)

  This field appears at the media level and contains the description of the media. It appears in the following format:

  m=<media><port><transport><fmt list>

- a= (information about the media field)

  This field can appear at the session or media level, in the following format:

  a=<attribute>

  a=<attribute>:<value>

The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

o=user 2344234 55234434 IN IP4 10.150.20.3

c=IN IP4 10.150.20.3

m=audio 43249 RTP/AVP 0

The following table shows how SIP payload is translated.

Inbound Request (from public to private)	To:	None
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace LSN pool IP address with private IP address
	Contact:	None
	Record-Route	None
	Route:	None
Outbound Response (from private to public)	To:	None
	From:	None
	Call-ID:	None
	Via:	None

	Request-URI:	Replace private IP address with LSN pool IP address
	Contact:	Replace private IP address with LSN pool IP address
	Record-Route	None
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	None
	Call-ID:	None
	Via:	Replace private IP address with LSN pool IP address
	Request-URI:	None
	Contact:	Replace private IP address with LSN pool IP address
	Record-Route	None
	Route:	None
Inbound Response (from public to private)	To:	None
	From:	None
	Call-ID:	None
	Via:	Replace LSN pool IP address with private IP address
	Request-URI:	None
	Contact:	Retain public IP address, if present
	Record-Route	None
	Route:	None

Limitations of SIP ALG

A SIP ALG has the following limitations:

- Only SDP payload is supported.
- The following are not supported:
  - Multicast IP addresses
  - Encrypted SDP
  - SIP TLS
  - FQDN translation
  - SIP layer authentication
  - TD/partitioning
  - Multipart body
  - SIP messages over IPv6 network
  - Line folding

Tested SIP Clients and Proxy Servers

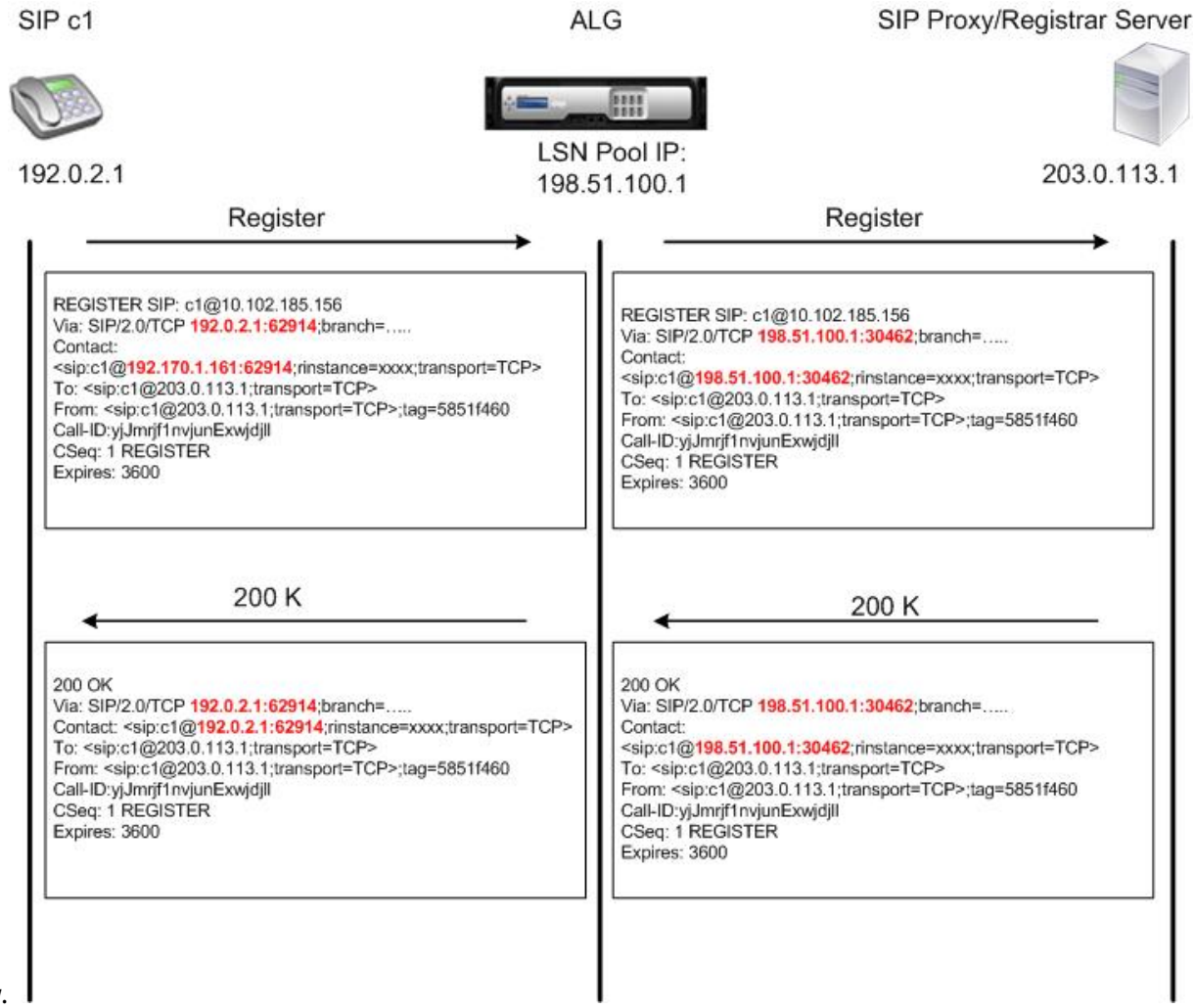
The following SIP clients and proxy server have been tested with SIP ALG:

- **SIP Clients:** X-Lite, Zoiper, Ekiga. Avaya
- **Proxy Server:** openSIPS

LSN SIP Scenario: SIP Proxy Outside the Private Network (Public Network)

SIP Client Registration

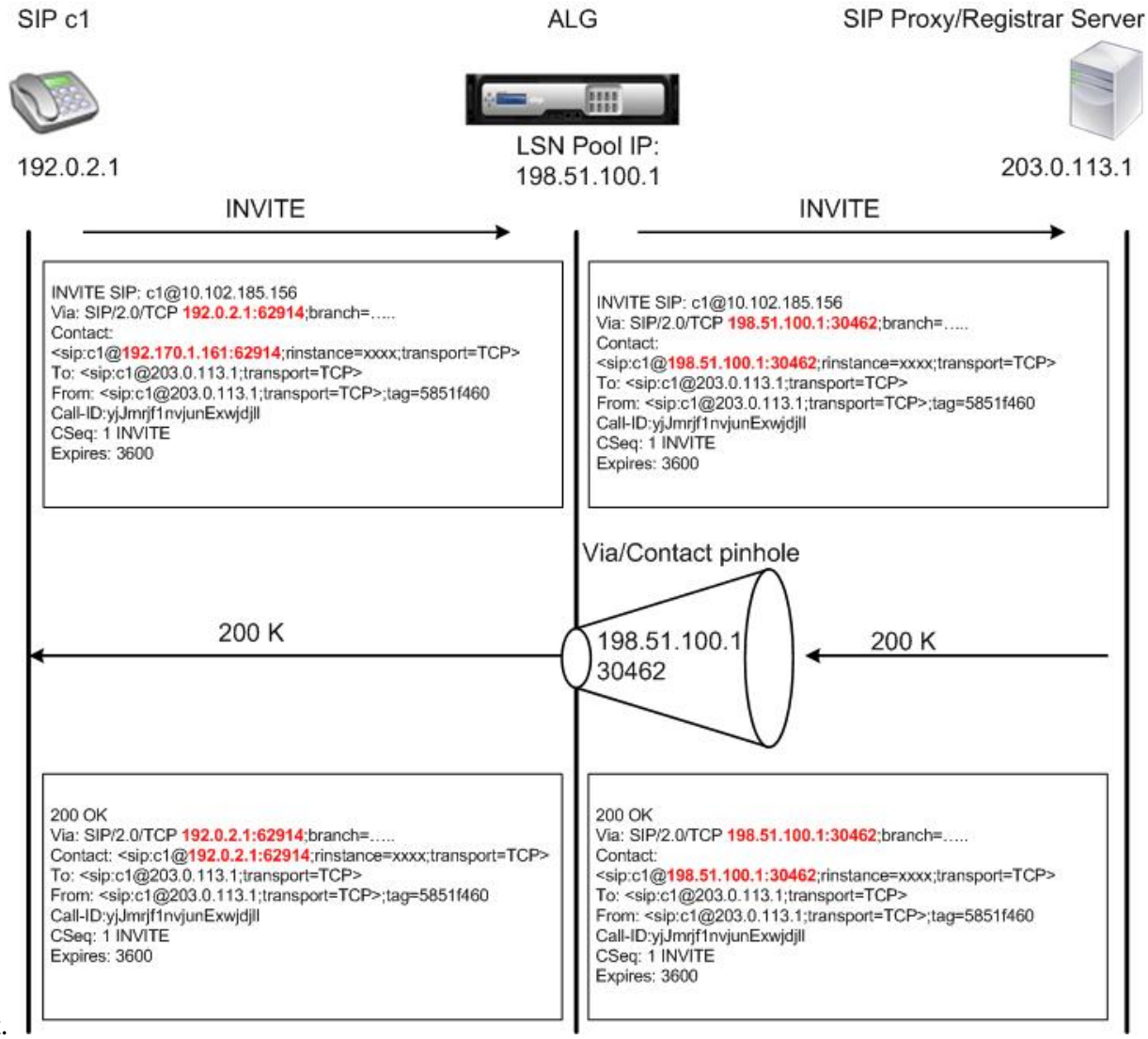
For a typical SIP call, SIP client must register with the SIP registrar by composing a REGISTER request and sending it to the SIP registrar. The Citrix ADC appliance's SIP ALG intercepts the request, replaces the IP address and port number in the request with the LSN pool IP address and port number provided in the LSN configuration, and forwards the request to the SIP registrar. The SIP ALG then opens a pinhole in the Citrix ADC configuration to allow further SIP communication between the SIP client and the SIP registrar. The SIP registrar sends a 200 OK response to the SIP client over the LSN pool IP address and port number. The Citrix ADC appliance captures this response in the pinhole, and the SIP ALG replaces the SIP header, putting the original Contact, Via, Route, and Record-Route SIP fields back into the message. The SIP ALG then forwards the message to the SIP client. The following figure shows how SIP ALG uses LSN in a SIP call registration



flow.

Outgoing Calls

A SIP call is initiated with a SIP INVITE message sent from the internal to the external network. The SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, replacing them with the LSN pool IP address and port number. LSN stores these mappings for subsequent SIP messages in the SIP call. The SIP ALG then opens separate pinholes in the Citrix ADC configuration to allow SIP and media through the Citrix ADC appliance on the dynamically assigned ports specified in the SDP and SIP headers. When a 200 OK message arrives at the Citrix ADC, it is captured by one of the created pinholes. The SIP ALG replaces the SIP header, restoring the original Contact, Via, Route, and Record-Route SIP fields, and then forwards the message to the internal SIP

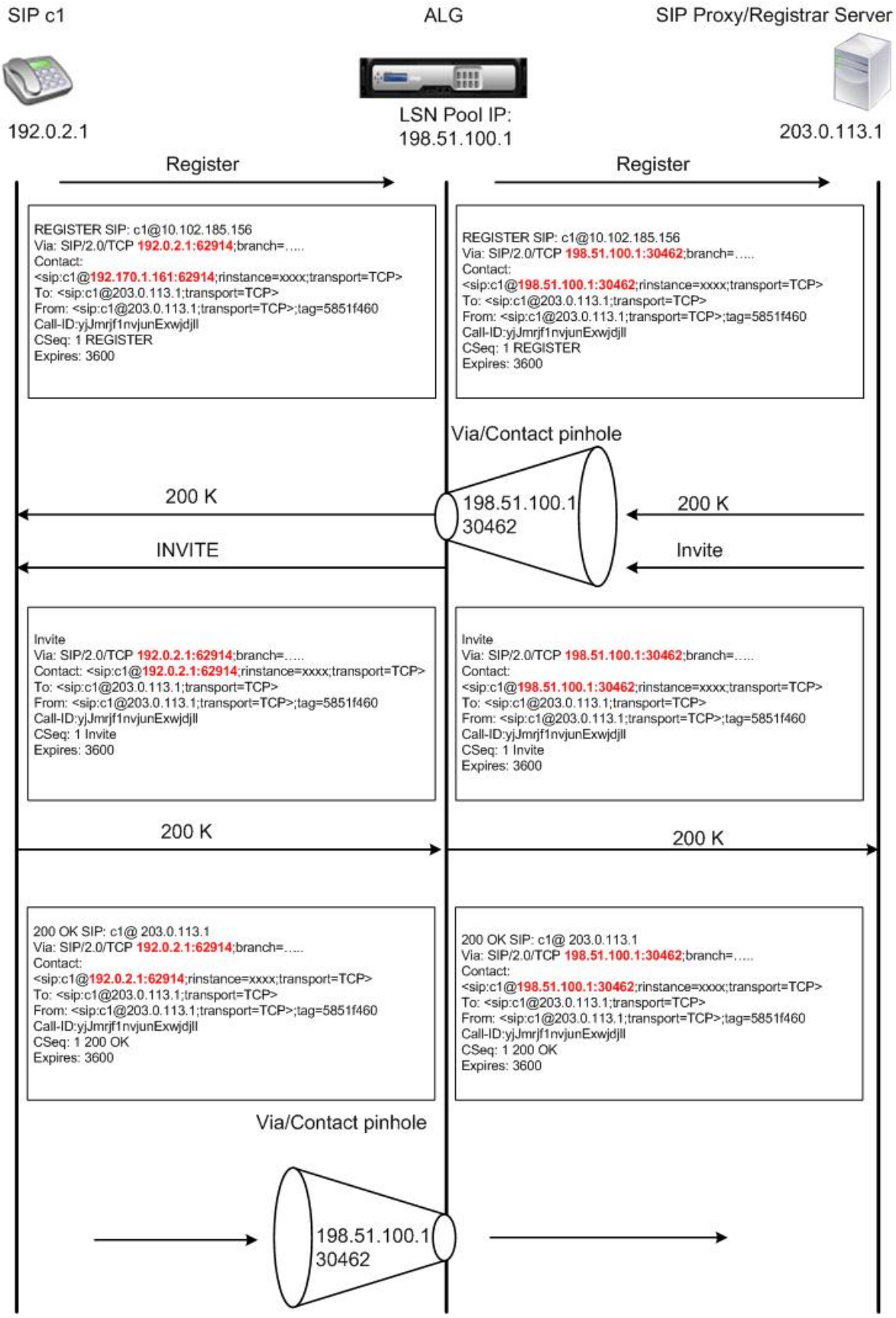


client.

Incoming Calls

A SIP incoming call is initiated with a SIP INVITE message from the external client to the internal network. The SIP registrar forwards the INVITE message to the SIP client in the internal network, using the pinhole that was created when the Internal SIP client registered with the SIP registrar.

The SIP ALG performs NAT on the LSN IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, translating them to the IP address and port number of the internal SIP client, and forwards the request to the SIP client. When the 200 OK response message sent by the internal SIP client arrives at the Citrix ADC appliance, the SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, translating them to the LSN pool IP address and port number, forwards the response message to the SIP registrar, and then opens a pinhole in the outbound direction for further SIP communica-



tion.

Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields in the message just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for 15 seconds to allow time for transmission of the 200 OK.

Call Between Clients in the Same Network

When both client A and client B in the same network initiate a call, the SIP messages are routed through the SIP proxy in the outside network. The SIP ALG processes the INVITE from client A as a normal outgoing call. Since client B is in the same network, the SIP proxy sends the INIVITE back to the Citrix ADC appliance. The SIP ALG examines the INIVITE message, determines that it contains the NAT IP address of client A, and replaces that address with the private IP address of client A before sending the message to client B. Once the call is established between the clients, the Citrix ADC is not involved in the media transmission between the clients.

More LSN SIP Scenarios: SIP Proxy Inside the Private Network

If you want to host the SIP Proxy server inside the private network, Citrix recommends that you do one of the following:

- Configure a static LSN Mapping for the private SIP proxy. For more information, see Configuring Static LSN Maps. Make sure that the NAT port is the same as the port configured in the SIP ALG profile.
- Configure the SIP Proxy server inside a demilitarized zone (DMZ).

Figure 1. SIP Call Registration

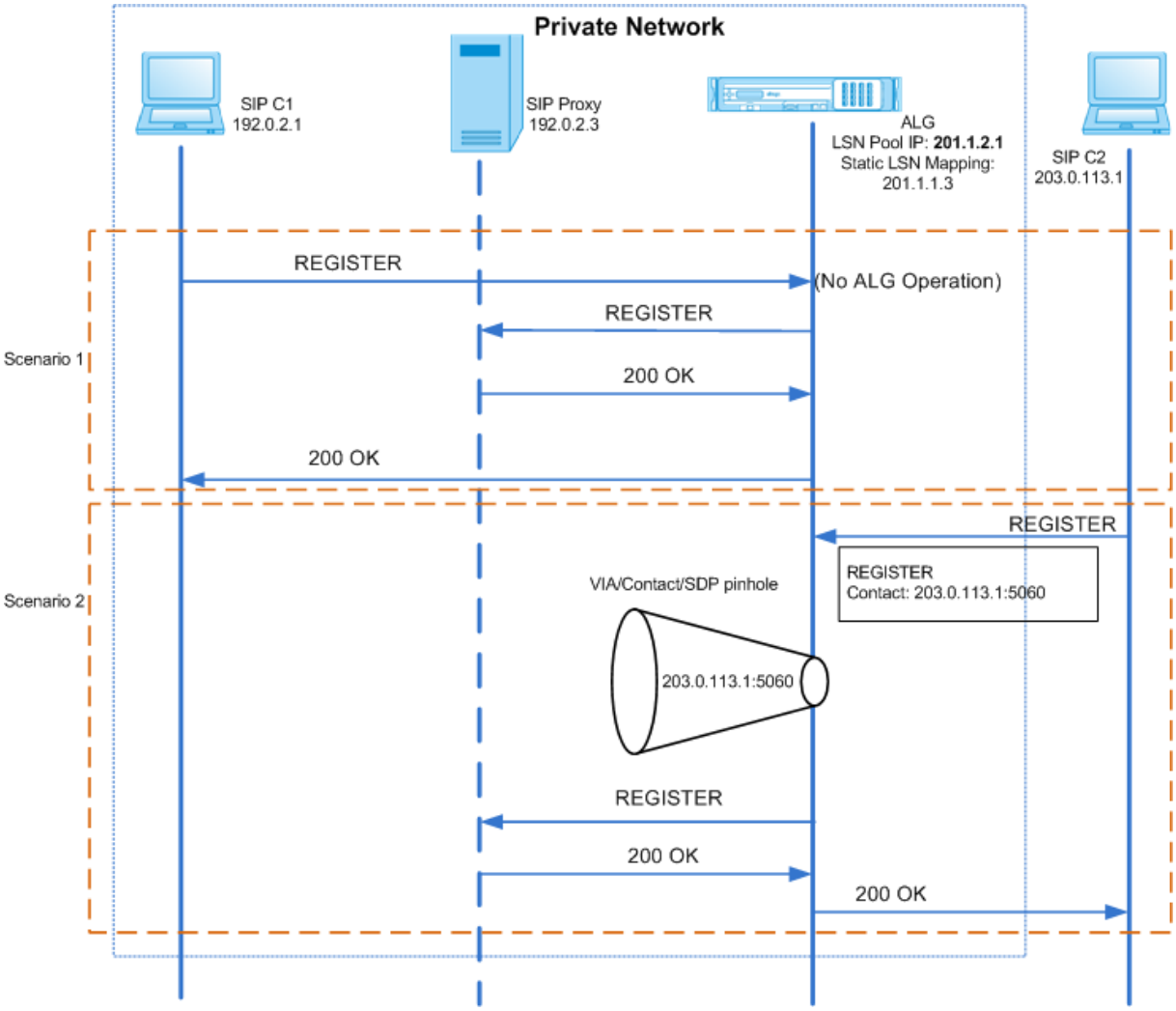
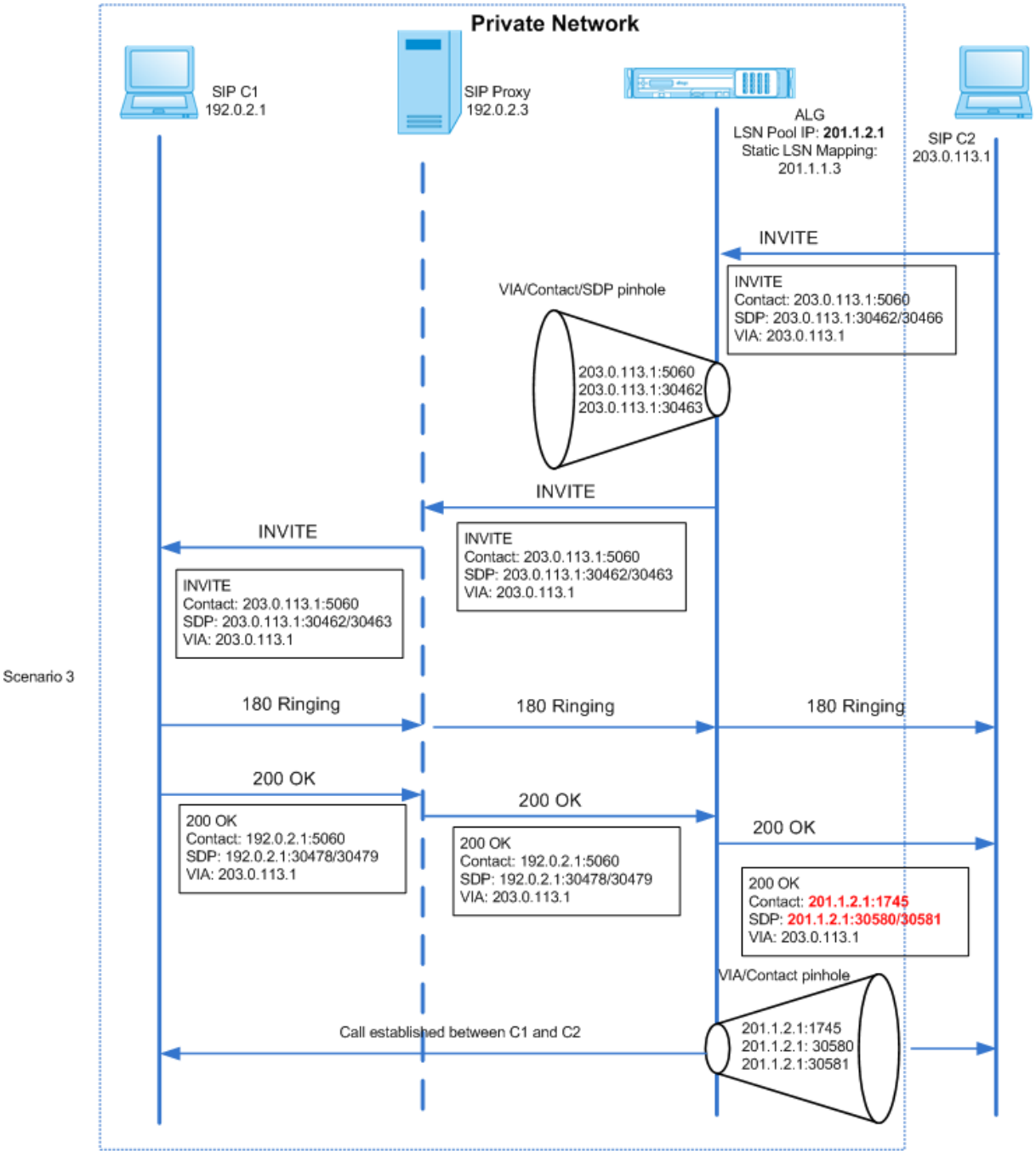


Figure 2. SIP Incoming Call Flow



Figures 1 and 2 show the following scenarios:

- Scenario 1—SIP client in the private network registers with the SIP proxy server in the same network. ALG operations are not performed, because the SIP client and SIP proxy server are in the same network.
- Scenario 2—SIP client in the public network registers with the SIP proxy server in the private network. The REGISTER message from the public SIP client is sent to the Citrix ADC appliance by using the static LSN mapping configured on the appliance, and the appliance creates a pinhole

for further SIP operations.

- Scenario 3— SIP Incoming call flow. A SIP incoming call is initiated with a SIP INVITE message from the external to the internal network. The Citrix ADC appliance receives the INVITE message from SIP client C2, which is in the external network, through the static LSN maps configured on the Citrix ADC appliance.
  The appliance creates a pinhole and forwards the INVITE message to the SIP proxy. The SIP proxy then forwards the INVITE message to SIP client C1 in the internal network. SIP client C1 then sends 180 and 200 OK messages to the SIP proxy, which in turn forwards the message to SIP client C2 through the Citrix ADC appliance.
  When the 200 OK response message sent by internal SIP client C1 arrives at the Citrix ADC, the SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, and in the SDP fields, replacing them with the LSN pool IP address and port number. The SIP ALG then forwards the response message to SIP client C2 and opens a pinhole in the outbound direction for further SIP communication.

Support for Audit Logs

You can log ALG information as part of LSN logging by enabling ALG in the LSN audit logging configuration. For more information on LSN logging, see Logging and Monitoring LSN. A log message for an ALG entry in the LSN log consists of the following information:

- Time stamp
- Type of SIP message (for example, SIP request)
- Source IP address and port of the SIP client
- Destination IP address and port of the SIP proxy
- NAT IP address and port
- SIP method
- Sequence number
- Whether or not the SIP client is registered
- Caller's user name and domain
- Receiver's user name and domain

Sample audit log:

Request:

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 169 0 : Infomsg: "SIP request" - Group: g2
     - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. - Transport
    : TCP - Source_IP: 192.169.1.165 - Source_port: 57952 -
    Destination_IP: 10.102.185.156 - Destination_port: 5060 - Natted_IP:
     10.102.185.191 - Natted_port: 10313 - Method: REGISTER -
    Sequence_Number: 3060 - Register: YES - Content_Type: -
```



```
    Caller_user_name: 156_pvt_1 - Callee_user_name: 156_pvt_1 -
    Caller_domain_name: - Callee_domain_name: -
<!--NeedCopy-->
```

Response:

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 170 0 : Infomsg: "SIP response" - Group:
    g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. -
    Transport: TCP - Response_code 200 - Source_IP: 10.102.185.156 -
    Source_port: 5060 - Destination_IP: 192.169.1.165 - Destination_port
    : 57952 - Natted_IP: 10.102.185.191 - Natted_port: 10313 -
    Sequence_Number: 3060 - Content_Type: - Caller_user_name: 156_pvt_1
    - Callee_user_name: 156_pvt_1 - Caller_domain_name: -
    Callee_domain_name: -
<!--NeedCopy-->
```

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Steps for LSN. While configuring LSN, make sure that you:

- Set the following parameters while adding the LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT

Important: For the SIP ALG to work, a full cone NAT configuration is mandatory.

Example:

```
add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT
<!--NeedCopy-->
```

- Create a SIP ALG profile and make sure that you define either the source port range or destination port range.

Example:

```
add lsn sipalgprofile sipalgprofile_tcp -sipsrcportrange 1-65535 -
    sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole
     ENABLED  - sipTransportProtocol TCP
<!--NeedCopy-->
```


- Set SIP ALG = ENABLED, while creating the LSN group.

Example:

```
add lsn group g1 -clientname c1 -sipalg ENABLED
<!--NeedCopy-->
```

- Bind the SIP ALG profile to the LSN group.

Sample SIP ALG Configuration:

The following sample configuration shows how to create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, SIP ALG specific setting, and configure SIP ALG:

```
add lsn pool p1

Done

bind lsn pool p1 10.102.185.190

Done

add lsn client c1

Done

bind lsn client c1 -network 192.170.1.0 -netmask 255.255.255.0

Done

add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app_udp UDP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app_tcp 1-65535

Done

bind lsn appsprofile app_udp 1-65535

Done
```



```
32
33 add lsn sipalgprofile sipalgprofile_tcp -sipdstportrange 5060 -
      openViaPinhole ENABLED -openRecordRoutePinhole ENABLED  -
      sipTransportProtocol TCP
34
35 Done
36
37 add lsn sipalgprofile sipalgprofile_udp -sipdstportrange 5060 -
      openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -
      sipTransportProtocol UDP
38
39 Done
40
41 add lsn group g1 -clientname c1 -sipalg ENABLED
42
43 Done
44
45 bind lsn group g1 -poolname p1
46
47 Done
48
49 bind lsn group g1 -appsprofilename app_tcp
50
51 Done
52
53 bind lsn group g1 -appsprofilename app_udp
54
55 Done
56
57 bind lsn group g1 -sipalgprofilename sipalgprofile_tcp
58
59 Done
60
61 bind lsn group g1 -sipalgprofilename sipalgprofile_udp
62
63 Done
64 <!--NeedCopy-->
```

Application Layer Gateway for RTSP Protocol

September 14, 2021

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. Citrix ADC functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to Citrix ADC owned public IP address called an LSN pool IP address.
- Inbound response—LSN pool IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

> **Note**
>
> RTSP ALG is supported in a Citrix ADC standalone appliance, in a Citrix ADC high availability setup, as well as in a Citrix ADC cluster setup.

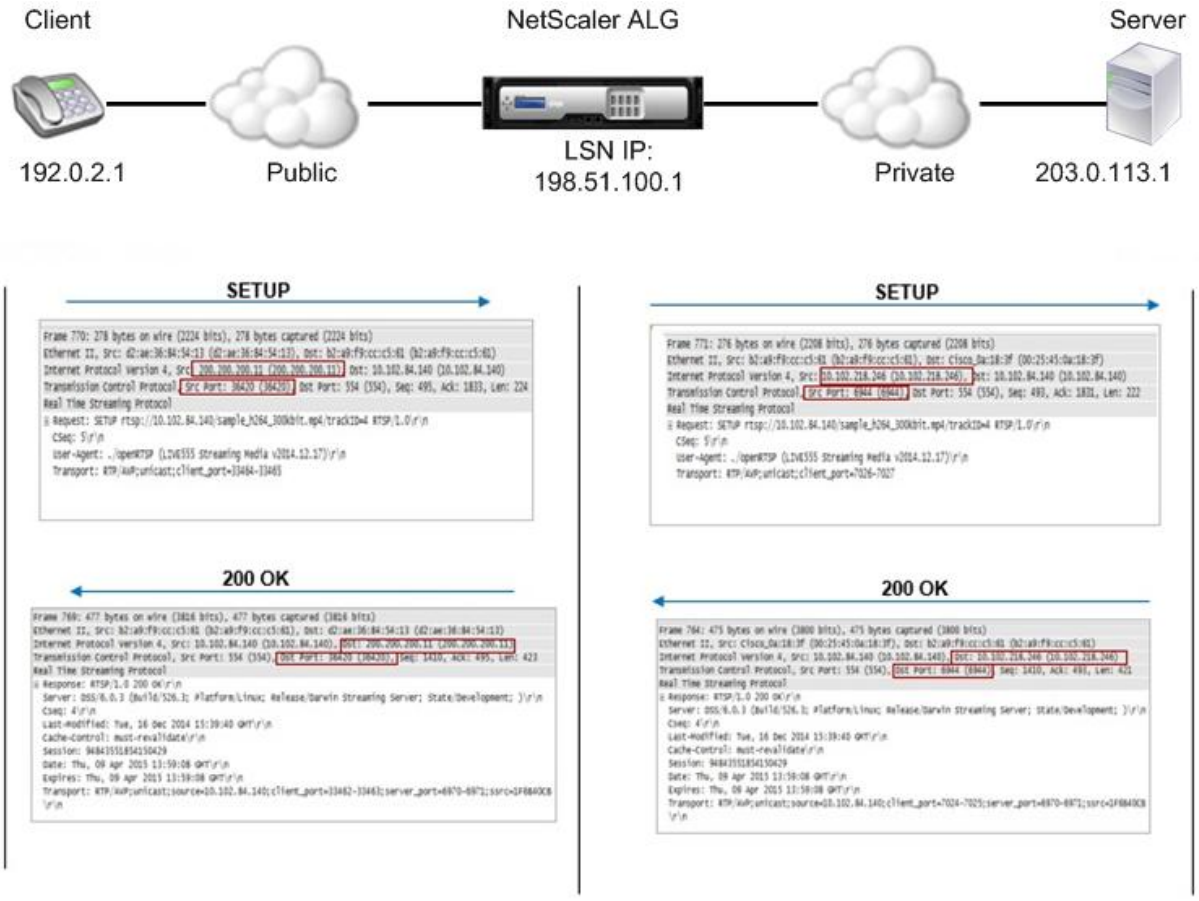
Limitations of RTSP ALG

The RTSP ALG does not support the following:

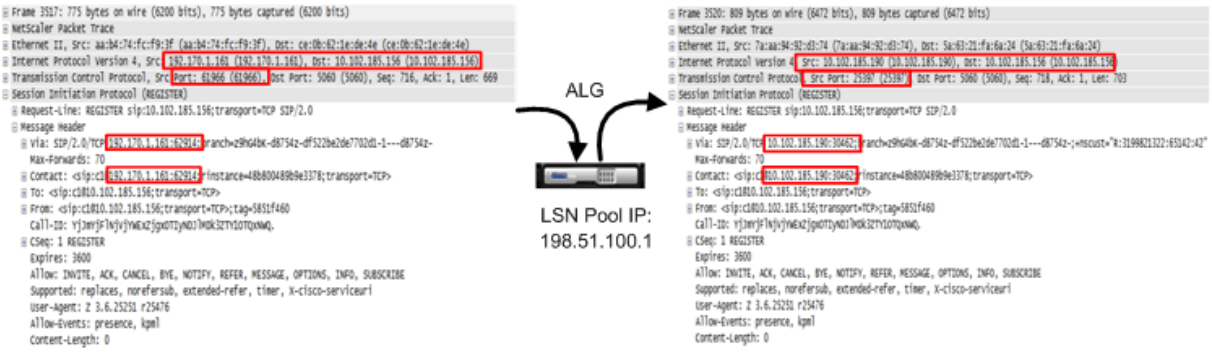
- Multicast RTSP sessions
- RTSP session over UDP
- TD/admin partitioning
- RSTP Authentication
- HTTP tunneling

RTSP and LSN scenario

The following figure shows an RTSP SETUP request flow. Typically, a SETUP request specifies how a single media stream must be transported. The request contains the media stream URL and a transport specifier. This specifier typically includes one local port for receiving RTP data (audio or video), and another for receiving RTCP data (meta information). The server reply usually confirms the chosen parameters and fills in the missing parts, such as the server's chosen ports. Each media stream must be configured by using the SETUP command before an aggregate play request can be sent.

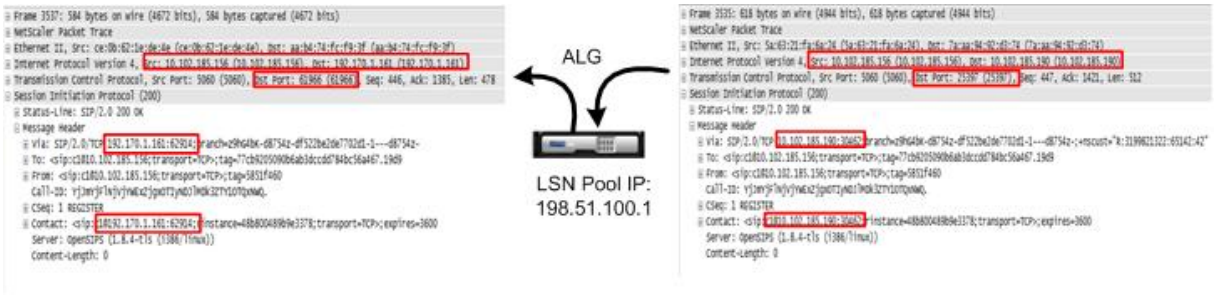


In a typical RTSP communication, the media client in the public network sends a SETUP request to the media server in the private network. RSTP ALG intercepts the request and, in the media stream, replaces the public IP address and port number with the LSN pool IP address and LSN port number. The following figure shows the translation performed by a Citrix ADC appliance in the media stream for an outbound request:



The media server in the private network uses the LSN pool IP address and LSN port number to send a 200 OK response to the media client in the public network. The Citrix ADC RTSP ALG intercepts the response and replaces the LSN pool IP address and LSN port number with the public IP address and port number of the media client. The following figure shows the translation performed by a Citrix ADC

appliance in the media stream for an inbound response:



Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Steps for LSN. While configuring LSN, make sure that you:

- Set the **NAT Type** as DETERMINSTIC or DYNAMIC while adding the LSN pool.
- Set the following parameters while adding the LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

Sample RTSP ALG Configuration:

The following sample configuration shows how to create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, and RTSP ALG settings:

```
1  enable ns feature WL SP LB CS LSN
2
3  Done
4
5  add lsn pool pool1 -nattype DETERMINISTIC
6
7  Done
8
9  bind lsn pool pool1 10.102.218.246
10
11 Done
12
13 add lsn client client1
14
15 Done
16
17 bind lsn client client1 -network 200.200.200.11 -netmask 255.255.255.0
18
```



```
Done

add lsn appsprofile app1 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app2 UDP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app1 1-65535

Done

bind lsn appsprofile app2 1-65535

Done

add lsn rtspalgprofile rtspalgprofiledefault -rtspIdleTimeout 1000 -rtspportrange 554

Done

add lsn group group1 -clientname client1 -nattype DETERMINISTIC -portblocksize 512 -rtspalg ENABLED

Done

bind lsn group group1 -poolname pool1

Done

bind lsn group group1 -appsprofilename app1

Done

bind lsn group group1 -appsprofilename app2

Done

bind lsn group group1 -rtspalgprofilename rtspalgprofiledefault

Done
```



```
<!--NeedCopy-->
```

Application Layer Gateway for IPSec Protocol

September 14, 2021

If communication between two network devices (for example, client and server) uses the IPSec protocol, IKE traffic (which is over UDP) uses port fields, but Encapsulating Security Payload (ESP) traffic does not. If a NAT device on the path assigns the same NAT IP address (but different ports) to two or more clients at the same destination, the NAT device is unable to distinguish and properly route the return ESP traffic does not contain port information. Therefore, IPSec ESP traffic fails at the NAT device.

NAT-Traversal (NAT-T) capable IPSec endpoints detect the presence of an intermediate NAT device during IKE phase 1 and switch to UDP port 4500 for all subsequent IKE and ESP traffic (encapsulating ESP in UDP). Without NAT-T support on the peer IPSec endpoints, IPSec protected ESP traffic is transmitted without any UDP encapsulation. Therefore, IPSec ESP traffic fails at the NAT device.

The Citrix ADC appliance supports IPSec application layer gateway (ALG) functionality for large scale NAT configurations. The IPSec ALG processes IPSec ESP traffic and maintains session information so that the traffic does not fail when the IPSec endpoints do no support NAT-T (UDP encapsulation of ESP traffic).

How IPSec ALG Works

An IPSec ALG monitors IKE traffic between a client and the server, and permits only one IKE phase 2 message exchange between the client and the server at any given time.

Once the two-way ESP packets are received for a particular flow, the IPSec ALG creates a NAT session for this particular flow so that subsequent ESP traffic can flow smoothly. The ESP traffic is identified by Security Parameters Indexes (SPIs), which are unique for a flow and for each direction. An IPSec ALG uses ESP SPIs in place of source and destination ports for performing large scale NAT.

If a gate receives no traffic, it times out. After both gates time out, another IKE phase 2 exchange is permitted.

IPSec ALG Timeouts

IPsec ALG on a Citrix ADC appliance has three timeout parameters:

- **ESP Gate Timeout**. Maximum time that the Citrix ADC appliance blocks an IPSec ALG gate for a particular client on a specific NAT IP address for a given server if no two-way ESP traffic is exchanged between the client and the server.
- **IKE Session Timeout**. Maximum time that the Citrix ADC appliance keeps the IKE session information before removing it if there is no IKE traffic for that session.
- **ESP Session Timeout**. Maximum time that Citrix ADC appliance keeps the ESP session information before removing it if there is no ESP traffic for that session.

Points to Consider before Configuring IPSec ALG

Before you start configuring IPSec ALG, consider the following points:

- You must understand the different components of IPSec protocol.
- IPSec ALG is not supported for DS-Lite and Large scale NAT64 configurations.
- IPSec ALG is not supported for hairpin LSN flow.
- IPSec ALG does not work with RNAT configurations.
- IPSec ALG is not supported in Citrix ADC clusters.

Configuration Steps

Configuring IPSec ALG for large scale NAT44 on a Citrix ADC appliance consists of the following tasks:

- **Create an LSN application profile and bind it to the LSN configuration.** Set the following parameters while configuring an application profile:
  - Protocol=UDP
  - IP Pooling = PAIRED
  - Port=500

Bind the application profile to the LSN group of an LSN configuration. For instructions on creating an LSN configuration, see Configuration Steps for LSN.

- **Create an IPSec ALG profile**. An IPSec profile includes various IPSec timeouts, such as IKE session timeout, ESP session timeout, and ESP gate timeout. You bind an IPSec ALG profile to an LSN group. An IPSec ALG profile has the following default settings:
  - IKE session timeout = 60 minutes
  - ESP session timeout = 60 minutes
  - ESP gate timeout = 30 seconds
- **Bind the IPSec ALG profile to the LSN configuration**. IPSec ALG is enabled for an LSN configuration when you bind an IPSec ALG profile to the LSN configuration. Bind the IPSec ALG profile to the LSN configuration by setting the IPSec ALG profile parameter to the name of the created profile in the LSN group. An IPSec ALG profile can be bound to multiple LSN groups, but an LSN group can have only one IPSec ALG profile.

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

At the command prompt, type:

```
add lsn appsprofile <appsprofilename> UDP -ippooling PAIRED

show lsn appsprofile
<!--NeedCopy-->
```

To bind destination port to the LSN application profile by using the command line interface

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

To bind an LSN application profile to an LSN group by using the command line interface

At the command prompt, type:

```
bind lsn group <groupname> -appsprofilename <string>

show lsn group
<!--NeedCopy-->
```

To create an IPSec ALG profile by using the CLI

At the command prompt, type:

```
add ipsecalg profile <name> [-ikeSessionTimeout <positive_integer>] [-
    espSessionTimeout <positive_integer>] [-espGateTimeout <
    positive_integer>] [-connfailover ( ENABLED | DISABLED)

show ipsecalg profile <name>
<!--NeedCopy-->
```

To bind an IPSec ALG profile to an LSN configuration by using the CLI

At the command prompt, type:


```
bind lsn group <groupname> -poolname <string> - ipsecAlgProfile <string
    >

show lsn group <name>
<!--NeedCopy-->
```

To create an LSN application profile and bind it to an LSN configuration by using the GUI

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **Application** tab, add an LSN application profile and bind it to an LSN group.

To create an IPSec ALG profile by using the GUI**

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **IPSEC ALG** tab, and then add an IPSec ALG profile.

To bind an IPSec ALG profile to an LSN configuration by using the GUI**

1. Navigate to **System** > **Large Scale NAT** > **LSN Group**, open the LSN group.
2. In **Advanced Settings**, click **+ IPSEC ALG Profile** to bind the created IPSec ALG profile to the LSN group.

Sample Configuration

In the following sample large scale NAT44 configuration, IPSec ALG is enabled for subscribers in the 192.0.2.0/24 network. IPSec ALG profile IPSECALGPROFILE-1 with various IPSec timeout settings is created and is bound to LSN group LSN Group -1.

Sample configuration:

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

```



```
bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.9

Done

add lsn appsprofile LSN-APPSPROFILE-1 UDP -ippooling PAIRED

Done

bind lsn appsprofile LSN-APPSPROFILE-1 500

Done

add ipsecalg profile  IPSECALGPROFILE-1 -ikeSessionTimeout 45  -
   espSessionTimeout 40  - espGateTimeout 20 -connfailover ENABLED

Done

bind lsn group LSN-GROUP-1 -appsprofilename LSN-APPSPROFILE-1

Done

bind lsn group LSN-GROUP-1 -poolname LSN-POOL-1

Done

bind lsn group LSN-GROUP-1 - ipsecAlgProfile IPSECALGPROFILE-1

Done
<!--NeedCopy-->
```

Logging and Monitoring LSN

September 14, 2021

You can log LSN information to diagnose, troubleshoot problems, and to meet legal requirements. You can monitor the performance of the LSN feature by using LSN statistical counters and displaying current LSN sessions.

Logging LSN

Logging LSN information is one of the important functions required by the ISPs to meet legal requirements and for identifying the source of traffic at any given time.

A Citrix ADC appliance logs LSN mapping entries and the LSN sessions created or deleted for each LSN group. You can control logging of LSN information for an LSN group by using the logging and session logging parameters of the LSN group. These are group level parameters and are disabled by default. The Citrix ADC appliance logs LSN sessions for an LSN group only when both logging and session logging parameters are enabled.

The following table displays the logging behavior for an LSN group for various settings of logging and session logging parameters.

Logging	Session Logging	Logging Behavior
Enabled	Enabled	Logs LSN mapping entries as well as LSN sessions.
Enabled	Disabled	Logs LSN mapping entries but not LSN sessions.
Disabled	Enabled	Logs neither mapping entries nor LSN sessions.

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

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (MAPPING)
- Whether the LSN mapping entry was created or deleted
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

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

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (SESSION)
- Whether the LSN session is created or removed

- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

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

For logging LSN information, Citrix recommends:

- Logging the LSN information on external log servers instead of on the Citrix ADC appliance. Logging on external servers facilitates optimal performance when the appliance creates large amounts of LSN log entries (in order of millions).
- Using SYSLOG over TCP, or NSLOG. By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data.

  Note:

  - The SYSLOG generated on Citrix ADC appliance are dynamically sent to the external log servers.
  - When using SYSLOG over TCP, if the TCP connection is down or the SYSLOG server is busy, then the Citrix ADC appliances stores the logs in buffer and send the data once the connection is active.

For more information about configuring logging, see Audit Logging.

Configuring LSN logging consists of the following tasks:

- **Configuring the Citrix ADC appliance for logging.** This task involves creating and setting various entities and parameters of the Citrix ADC appliance:
  - **Create a SYSLOG or NSLOG audit logging configuration**. Creating an audit logging configuration involves the following tasks:
    * Create a NSLOG or SYSLOG audit action and enable the LSN parameter. Audit actions specify the IP addresses of log servers.
    * Create a SYSLOG or NSLOG audit policy and bind the audit action to the audit policy. Audit actions specify the IP addresses of log servers. Optionally, you can set the transport method for log messages that are sent to the external log servers. By default UDP is selected, you can set the transport method as TCP for a reliable transport mechanism.Bind the audit policy to system global.
    * Create a SYSLOG or NSLOG audit policy and bind the audit action to the audit policy.

    * Bind the audit policy to system global.
      Note: For an existing audit logging configuration, just enable the LSN parameter for logging LSN information in the server specified by the audit action.
  - **Enable logging and session logging parameters.** Enable logging and session logging parameters either as you add LSN groups or after you have created the groups. The Citrix ADC appliance generates log messages related to these LSN groups and sends them to the server of those audit actions that have the LSN parameter enabled.
- **Configuring log servers.** This task involves installing SYSLOG or NSLOG packages on the desired servers. This task also involves specifying the NSIP address of the Citrix ADC appliance in the configuration file of SYSLOG or NSLOG. Specifying the NSIP address enables the server to identify the log information sent by the Citrix ADC appliance for storing them in a log file.

For more information about configuring logging, see Audit Logging.

SYSLOG Configuration Using the Command Line Interface

To create a SYSLOG server action for LSN logging by using the command line interface

At the command prompt, type:

```
add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel>... [-transport (TCP)] [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

To create a SYSLOG server policy for LSN logging by using the command line interface

At the command prompt, type:

```
add audit syslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

To bind a SYSLOG server policy to system global for LSN logging by using the command line interface

At the command prompt, type:

```
bind system global [<policyName> [-priority <positive_integer>]]
<!--NeedCopy-->
```

SYSLOG Configuration Using the Configuration Utility

To configure a SYSLOG server action for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Syslog** and, on the Servers tab, add a new auditing server or edit an existing server.
2. To enable LSN logging, select the **Large Scale NAT Logging** option.
3. (Optional) To enable SYSLOG over TCP, select the **TCP Logging** option.

To configure a SYSLOG server policy for LSN logging by using the configuration utility

Navigate to **Systems** > **Auditing** > **Syslog** and, on the **Policies** tab, add a new policy or edit an existing policy.

To bind a SYSLOG server policy to system global for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Syslog**.
2. On the **Policies** tab, in the **Action** list, click **Global Bindings** to bind the audit global policies.

NSLOG Configuration Using the Command Line Interface

To create a NSLOG server action for LSN logging by using the command line interface

At the command prompt, type:

```
add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> ... [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

To create a NSLOG server policy for LSN logging by using the command line interface

At the command prompt, type:

```
add audit nslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

To bind a NSLOG server policy to system global for LSN logging by using the command line interface

At the command prompt, type:

```
bind system global [<policyName>]
<!--NeedCopy-->
```


NSLOG Configuration Using the Configuration Utility

To configure a NSLOG server action for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Nslog** and, on the **Servers** tab, add a new auditing server or, edit an existing server.
2. To enable LSN logging, select the **Large Scale NAT Logging** option.

To configure a NSLOG server policy for LSN logging by using the configuration utility

Navigate to **Systems** > **Auditing** > **Nslog** and, on the **Policies** tab, add a new policy or edit an existing policy.

To bind a NSLOG server policy to system global for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Nslog**.
2. On the **Policies** tab, in the **Action** list, click **Global Bindings** to bind the audit global policies.

Example

The following configuration specifies two SYSLOG and two NSLOG servers for storing log entries including LSN logs. LSN Logging is configured for LSN groups LSN-GROUP-2 and LSN-GROUP-3.

The Citrix ADC appliance generates log messages related to LSN mappings and LSN sessions of these LSN groups, and sends them to the specified log servers.

```
add audit syslogAction SYS-ACTION-1 198.51.101.10 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done
bind system global SYSLOG-POLICY-1
Done

add audit syslogAction SYS-ACTION-2 198.51.101.20 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-2 ns_true SYS-ACTION-2
Done
bind system global SYSLOG-POLICY-2
Done

add audit nslogAction NSLOG-ACTION-1 198.51.101.30 -logLevel ALL -lsn
    ENABLED
```



```
Done
add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1
Done
bind system global NSLOG-POLICY-1
Done
add audit nslogAction NSLOG-ACTION-2 198.51.101.40 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-2 ns_true NSLOG-ACTION-2
Done
bind system global NSLOG-POLICY-2
Done

add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-2  - logging ENABLED  -
    sessionLogging ENABLED
Done
set lsn group LSN-GROUP-2   - logging ENABLED  - sessionLogging ENABLED
Done
<!--NeedCopy-->
```

The following configuration specifies SYSLOG configuration for sending log messages to the external SYSLOG server 192.0.2.10 using TCP.

```
add audit syslogAction SYS-ACTION-1 192.0.2.10 -logLevel ALL -transport
    TCP
Done

add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done

bind system global SYSLOG-POLICY-1
Done
<!--NeedCopy-->
```

The following table displays sample LSN log entries of each type stored on the configured log servers. These LSN log entries are generated by a Citrix ADC appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry
LSN session creation	Local4.Informational 10.102.37.115 08/05/2014:09:59:48 GMT 0-PPE-0 : LSN LSN_SESSION 2581750 : SESSION CREATED Client IP:Port:TD 192.0.2.10: 15136:0, NatIP:NatPort 203.0.113.6: 6234, Destination IP:Port:TD 198.51.100.9: 80:0, Protocol: TCP
LSN session deletion	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_SESSION 3871790 : SESSION DELETED Client IP:Port:TD 192.0.2.11: 15130:0, NatIP:NatPort 203.0.113.6: 7887, Destination IP:Port:TD 198.51.101.2:80:0, Protocol: TCP
LSN mapping creation	Local4.Informational 10.102.37.115 08/05/2014:09:59:47 GMT 0-PPE-0 : LSN LSN_MAPPING 2581580 : EIM CREATED Client IP:Port 192.0.2.15: 14567, NatIP:NatPort 203.0.113.5: 8214, Protocol: TCP
LSN mapping deletion	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_MAPPING 3871700 : EIM DELETED Client IP:Port 192.0.3.15: 14565, NatIP:NatPort 203.0.113.11: 8217, Protocol: TCP

Minimal Logging

Deterministic LSN configurations and Dynamic LSN configurations with port block significantly reduces the LSN log volume. For these two types of configuration, the Citrix ADC appliance allocates a NAT IP address and a block of ports to a subscriber. The Citrix ADC appliance generates a log message for a port block at the time of allocation to a subscriber. The Citrix ADC appliance also generates a log message when a NAT IP address and port block is freed. For a connection, a subscriber can be identified just by its mapped NAT IP address and port block. Because of this reason, the Citrix ADC appliance does not log any LSN session created or deleted. The appliance also neither logs any mapping entry created for a session nor when the mapping entry gets removed.

The minimal logging feature for deterministic LSN configurations and dynamic LSN configurations with port block is enabled by default and there is no provision to disable it. In other words, the Citrix ADC appliance automatically do minimal logging for deterministic LSN configurations and dynamic

LSN configurations with port block. There is no option available for disabling this feature. The appliance sends the log messages to all the configured log servers.

A log message for each port block consists of the following information:

- NSIP address of the Citrix ADC appliance
- Time stamp
- Entry type as DETERMINISTIC or PORTBLOCK
- Whether a port block is allocated or is freed
- Subscriber's IP address and the assigned NAT IP address and port block
- Protocol name

Minimal Logging for Deterministic LSN Configuration

Consider an example of a simple deterministic LSN configuration for four subscribers having the IP address 192.0.17.1, 192.0.17.2, 192.0.17.3, and 192.0.17.4.

In this LSN configuration, the port block size is set to 32768 and LSN NAT IP address pool has IP addresses in the range 203.0.113.19-203.0.113.23.

```
1  add lsn client LSN-CLIENT-7
2  Done
3  bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask
       255.255.255.253
4  Done
5  add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
6  Done
7  bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
8  Done
9  add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
       DETERMINISTIC -portblocksize 32768
10 Done
11 bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
12 Done
13 <!--NeedCopy-->
```

The Citrix ADC appliance sequentially preallocates, from the LSN NAT IP pool and on the basis of the set port block size, an LSN NAT IP address and a block of ports to each subscriber. It assigns the first block of ports (1024-33791) on the beginning NAT IP address (203.0.113.19) to the beginning subscriber IP address (192.0.17.1). The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT IP address is assigned to the subscriber, and so on. The appliance logs the NAT IP address and the block of ports allocated for each subscriber.

The Citrix ADC appliance does not log any LSN session created or deleted for these subscribers. The appliance generates the following log messages for the LSN configuration.

```
1) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201453 0 :  Dtrstc ALLOC Client 12.0.0.241,
   NatInfo 50.0.0.2:59904 to 60415
2) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201454 0 :  Dtrstc ALLOC Client 12.0.0.242,
   NatInfo 50.0.0.2:60416 to 60927
3) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
   NatInfo 50.0.0.2:60928 to 61439
4) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
   NatInfo 50.0.0.2:60928 to 61439
<!--NeedCopy-->
```

When you remove the LSN configuration, the allocated NAT IP address and block of ports is freed from each subscriber. The appliance logs NAT IP address and block of ports freed from each subscriber. The appliance generates the following log messages for each subscriber when you remove the LSN configuration.

```
1) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201706 0 :  Dtrstc FREE Client 12.0.0.238,
   NatInfo 50.0.0.2:58368 to 58879
2) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201707 0 :  Dtrstc FREE Client 12.0.0.239,
   NatInfo 50.0.0.2:58880 to 59391
3) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201708 0 :  Dtrstc FREE Client 12.0.0.240,
   NatInfo 50.0.0.2:59392 to 59903
<!--NeedCopy-->
```

Minimal Logging for Dynamic LSN Configuration with Port Block

Consider an example of a simple dynamic LSN configuration with port block for any subscriber in the network 192.0.2.0/24. In this LSN configuration, the port block size is set to 1024 and LSN NAT IP address pool has IP addresses in the range 203.0.113.3-203.0.113.4.

```
set lsn parameter -memLimit 4000
Done
add lsn client LSN-CLIENT-1
Done
bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
```



```
6  Done
7  add lsn pool LSN-POOL-1
8  Done
9  bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
10 Done
11 add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
12 Done
13 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
14 Done
15 <!--NeedCopy-->
```

The Citrix ADC appliance allocates a random NAT IP address and a block of ports, from the LSN NAT IP pool and on the basis of the set port block size, for a subscriber when it initiates a session for the first time. The Citrix ADC logs the NAT IP address and block of ports allocated to this subscriber. The appliance does not log any LSN session created or deleted for this subscriber. If all the ports are allocated (for different subscriber's sessions) from the subscriber's allocated port block, the appliance allocates a new random NAT IP address and port block for the subscriber for additional sessions. The Citrix ADC logs every NAT IP address and port block allocated to a subscriber.

The appliance generates the following log message when the subscriber, having the IP address 192.0.2.1, initiates a session. The log message shows that the appliance has allocated NAT IP address 203.0.113.3 and port block 1024-2047 to the subscriber.

```
1 03/23/2015:00:07:12 GMT  Informational 0-PPE-3 : default LSN
     LSN_PORTBLOCK 106725793 0 :  Portblock ALLOC Client 12.0.2.72,
     NatInfo 203.0.113.3:1024 to 2047, Proto:TCP
2 <!--NeedCopy-->
```

Once there are no more sessions left that is using the allocated NAT IP address and one of the ports in the allocated port block, the allocated NAT IP address and block of ports is freed from the subscriber. The Citrix ADC logs that the NAT IP address and the block of ports is freed from the subscriber. The appliance generates the following log messages for the subscriber, having the IP address 192.0.2.1 , when no more sessions are left that is using the allocated NAT IP address (203.0.113.3) and a port from the allocated port block (1024-2047). The log message shows that the NAT IP address and port block are freed from the subscriber.

```
1 03/23/2015:00:11:09 GMT Informational 0-PPE-3 : default LSN
     LSN_PORTBLOCK 106814342 0 :  Portblock FREE Client 12.0.3.122,
     NatInfo 203.0.113.3: 1024 to 2047, Proto:TC
2 <!--NeedCopy-->
```


Load Balancing SYSLOG Servers

The Citrix ADC appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the Citrix ADC appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

```
add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
    SYSLOGUDP)> <port>
<!--NeedCopy-->
```

Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```

Bing the service to the load balancing virtual server.

```
Bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
     <logLevel>]
<!--NeedCopy-->
```

Add a SYSLOG policy by specifying the rule and action.

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

Bind the SYSLOG policy to the system global for the policy to take effect.

```
bind system global <policyName>
<!--NeedCopy-->
```


Load balancing of SYSLOG servers using the configuration utility

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

   Navigate to Traffic Management > Services, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

   Navigate to Traffic Management > Virtual Servers, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

3. Bing the service to the load balancing virtual server to the service.

   Bing the service to the load balancing virtual server.

   Navigate to Traffic Management > Virtual Servers, select a virtual server and then selectAUDITLOGHASH in the Load Balancing Method.

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   Navigate to System > Auditing, click Servers and add a server by selecting LB Vserver option inServers.

5. Add a SYSLOG policy by specifying the rule and action.

   Navigate to System > Syslog, click Policies and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   Navigate to System > Syslog, select a SYSLOG policy and click Action, and then click Global Bindings and bind the policy to system global.

Example:

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. The Citrix ADC appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

```
add service service1 192.0.2.10 SYSLOGUDP 514
Done

add service service2 192.0.2.11 SYSLOGUDP 514
Done

add service service3 192.0.2.11 SYSLOGUDP 514
Done

```



```
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
Done

bind lb vserver lbvserver1 service1
Done

bind lb vserver lbvserver1 service2
Done

bind lb vserver lbvserver1 service3
Done

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
Done

add syslogpolicy syspol1 ns_true sysaction1
Done

bind system global syspol1
Done
<!--NeedCopy-->
```

Logging HTTP Header Information

The Citrix ADC appliance can now log request header information of an HTTP connection that is using the LSN fucntionality of the Citrix ADC. The following header information of an HTTP request packet can be logged:

- URL that the HTTP request is destined to.
- HTTP Method specified in the HTTP request.
- HTTP version used in the HTTP request.
- IP address of the subscriber that sent the HTTP request.

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set a subscribers. For example, an ISP can use this feature to find out the most popular websites among a set of subscribers.

An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging. The HTTP header log profile is then bound to an LSN group. The Citrix ADC appliance then logs HTTP header attributes, which are enabled in the bound HTTP header log profile for logging, of any HTTP requests related to the LSN group. The appliance then sends the log messages to the configured log servers.

An HTTP header log profile can be bound to multiple LSN groups but an LSN group can have only one

HTTP header log profile.

To create an HTTP header log profile by using the the command line interface

At the command prompt, type:

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED | DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion ( ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

To bind an HTTP header log profile to an LSN group by using the the command line interface

At the command prompt, type:

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

Example

In the following example of an LSN configuration, HTTP header log profile HTTP-Header-LOG-1 is bound to LSN group LSN-GROUP-1. The log profile has all the HTTP attributes (URL, HTTP method, HTTP version, and HOST IP address) enabled for logging so that all these attributes are logged for any HTTP requests from subscribers (in the network 192.0.2.0/24) related to the LSN group.

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done

set lsn parameter -memLimit 4000
Done

add lsn client LSN-CLIENT-1
Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done

add lsn pool LSN-POOL-1
Done
```



```
15
16 bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
17 Done
18
19 add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
20 Done
21
22 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
23 Done
24
25 bind lsn group  LSN-GROUP-1  -httphdrlogprofilename HTTP-HEADER-LOG-1
26 Done
27 <!--NeedCopy-->
```

The Citrix ADC generates the following HTTP header log message when one of the subscriber belonging to the LSN configuration example sends an HTTP request.

The log message tells us that a client having the IP address 192.0.2.33 sends an HTTP request to URL example.com using HTTP method GET and HTTP version 1.1.

```
1 03/19/2015:16:24:04 GMT  Informational 0-PPE-1 : default LSN Message 59
      0 :  "LSN Client IP:TD 10.102.37.118:0 URL: example.com Host:
     192.0.2.33 Version: HTTP1.1 Method: GET"
2 <!--NeedCopy-->
```

Logging MSISDN Information

A Mobile Station Integrated Subscriber Directory Number (MSISDN) is a telephone number uniquely identifying a subscriber across multiple mobile networks. The MSISDN is associated with a country code and a national destination code identifying the subscriber's operator.

You can configure a Citrix ADC appliance to include MSISDNs in LSN log entries for subscribers in mobile networks. The presence of MSISDNs in the LSN logs helps the administrator in faster and accurate back tracing of a mobile subscriber who has violated a policy or law, or whose information is required by lawful interception agencies.

The following sample LSN log entries include MSISDN information for a connection from a mobile subscriber in an LSN configuration. The log entries show that a mobile subscriber whose MSISDN is E164:5556543210 was connected to destination IP:port 23.0.0.1:80 through the NAT IP:port 203.0.113.3:45195.

Log Entry Type	Sample Log Entry
LSN session creation	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN mapping creation	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN session deletion	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN mapping	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP

Perform the following tasks for including MSISDN information in LSN logs

- **Create an LSN log profile**. An LSN log profile includes the log subscriber ID parameter, which specifies whether to or not to include the MSISDN information in the LSN logs of an LSN configuration. Enable the log subscriber ID parameter when creating the LSN log profile.
- **Bind the LSN log profile to an LSN group of an LSN configuration**. Bind the created LSN log profile to an LSN group of an LSN configuration by setting the log profile name parameter to the

created LSN log profile name. For instructions on configuring Large Scale NAT, see Configuration Steps for LSN.

To create an LSN log profile by using the CLI

At the command prompt, type:

```
add lsn logprofile <logprofilename -logSubscriberID ( ENABLED |
   DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind an LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

Sample Configuration:

In this example of LSN configuration, the LSN log profile has the log subscriber ID parameter enabled. The profile is bound to LSN group LSN-GROUP-9. MSISDN information is included in the LSN session and LSN mapping logs for connections from mobile subscribers (in the network 192.0.2.0/24).

```
add lsn logprofile  LOG-PROFILE-MSISDN-9   -logSubscriberID ENABLED

Done
add lsn client LSN-CLIENT-9

Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done
add lsn pool LSN-POOL-9

Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
```



```
17
18 Done
19 bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
20
21 Done
22 bind lsn group  LSN-GROUP-9  -logprofilename LOG-PROFILE-MSISDN-9
23
24 Done
25 <!--NeedCopy-->
```

Displaying Current LSN Sessions

You can display the current LSN sessions for detecting any unwanted or inefficient LSN sessions on the Citrix ADC appliance. You can display all or some LSN sessions on the basis of selection parameters.

Note: When more than a million LSN sessions exist on the Citrix ADC appliance, Citrix recommends displaying selected LSN sessions instead of all by using the selection parameters.

Configuration Using the Command Line Interface

To display all LSN sessions by using the command line interface

At the command prompt, type:

```
1 show lsn session
2 <!--NeedCopy-->
```

To display selective LSN sessions by using the command line interface

At the command prompt, type:

```
1 show lsn session [-clientname <string>] [-network <ip_addr> [-netmask <
     netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
     port>]]
2 <!--NeedCopy-->
```

Example

To display all LSN sessions existing on a Citrix ADC


```
> show lsn session
          SubscrIP             SubscrPort      SubscrTD              DstIP                      DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10              15136           0              198.51.100.9            80              0       203.0.113.6     6234            TCP     OUT
2.      192.0.2.11              15130           0              198.51.101.2            80              0       203.0.113.6     7887            TCP     OUT
3.      192.0.2.12              16136           0              198.51.100.3            80              0       203.0.113.6     9807            TCP     OUT
4.      192.0.2.13              18148           0              198.51.101.6            80              0       203.0.113.6     4657            TCP     OUT
5.      192.0.2.14              13560           0              198.51.101.7            80              0       203.0.113.7     9341            TCP     OUT
6.      192.0.2.15              14567           0              198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
7.      192.0.2.15              16890           0              198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
8.      192.0.2.16              12345           0              198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
9.      192.0.2.19              19876           0              198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
10.     192.0.2.20              10989           0              198.51.104.19           80              0       203.0.113.11    1343            TCP     OUT
11.     192.0.3.13              18149           0              198.51.101.61           80              0       203.0.113.11    4653            TCP     OUT
12.     192.0.3.14              13510           0              198.51.101.74           80              0       203.0.113.11    9344            TCP     OUT
13.     192.0.3.15              14565           0              198.51.100.82           80              0       203.0.113.11    8217            TCP     OUT
14.     192.0.3.15              16899           0              198.51.101.12           80              0       203.0.113.11    8219            TCP     OUT
15.     192.0.3.16              12343           0              198.51.102.99           80              0       203.0.113.11    1673            TCP     OUT

Done
```

To display all LSN sessions related to an LSN client entity LSN-CLIENT-2

```
> show lsn session -clientname LSN-CLIENT-2
        SubscrIP               SubscrPort      SubscrTD              DstIP                    DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10              15136           0              198.51.100.9            80              0       203.0.113.6     68234           TCP     OUT
2.      192.0.2.11              15130           0              198.51.101.2            80              0       203.0.113.6     7887            TCP     OUT
3.      192.0.2.12              16136           0              198.51.100.3            80              0       203.0.113.6     9807            TCP     OUT
4.      192.0.2.13              18148           0              198.51.101.6            80              0       203.0.113.6     4657            TCP     OUT
5.      192.0.2.14              13560           0              198.51.101.7            80              0       203.0.113.7     9341            TCP     OUT
6.      192.0.2.15              14567           0              198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
7.      192.0.2.15              16890           0              198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
8.      192.0.2.16              12345           0              198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
9.      192.0.2.19              19876           0              198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
10.     192.0.2.20              10989           0              198.51.104.19           80              0       203.0.113.11    1343            TCP     OUT

Done
```

To display all LSN sessions that uses 203.0.113.5 as the NAT IP address

```
> show lsn session –natIP 203.0.113.5
SubscrIP               SubscrPort      SubscrTD        DstIP                   DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.15              14567           0              198.51.100.8            80              0       203.0.113.5     8214            TCP     OUT
2.      192.0.2.15              16890           0              198.51.101.1            80              0       203.0.113.5     8214            TCP     OUT
3.      192.0.2.16              12345           0              198.51.102.9            80              0       203.0.113.5     1678            TCP     OUT
4.      192.0.2.19              19876           0              198.51.103.8            80              0       203.0.113.5     1567            TCP     OUT
Done
```

Configuration Using the Configuration Utility

To display all or selected LSN sessions by using the configuration utility

1. Navigate to System > Large Scale NAT > Sessions, and click the NAT44 tab.
2. For displaying LSN sessions on the basis of selection parameters, click Search.

Parameter Descriptions (of commands listed in the CLI procedure)

- show lsn session
  - clientname

    Name of the LSN Client entity. Maximum Length: 127
  - network

    IP address or network address of subscriber(s).

- netmask

  Subnet mask for the IP address specified by the network parameter.

  Default value: 255.255.255.255

- td

  Traffic domain ID of the LSN client entity.

  Default value: 0

  Minimum value: 0

  Maximum value: 4094

- natIP

  Mapped NAT IP address used in LSN sessions.

Displaying LSN Statistics

You can display statistics related to the LSN feature for evaluating the performance of the LSN feature or to troubleshoot problems. You can display a summary of statistics of the LSN feature or of a particular LSN group. The statistical counters reflect events since the Citrix ADC appliance was last restarted. All these counters are reset to 0 when the Citrix ADC appliance is restarted.

To display all LSN statistics by using the command line interface

At the command prompt, type:

```
stat lsn
<!--NeedCopy-->
```

To display statistics for a specified LSN group by using the command line interface

At the command prompt, type:

```
stat lsn group [<groupname>]
<!--NeedCopy-->
```

Example

```
> stat lsn

Large Scale NAT statistics
```



```
4                                                    Rate(/s)
                                                        Total
5  LSN TCP Received Packets                            0
       40
6  LSN TCP Received Bytes                              0
       3026
7  LSN TCP Transmitted Packets                         0
       40
8  LSN TCP Transmitted Bytes                           0
       3026
9  LSN TCP Dropped Packets                             0
                            0
10 LSN TCP Current Sessions                            0
                            0
11 LSN UDP Received Packets                            0
                            0
12 LSN UDP Received Bytes                              0
                            0
13 LSN UDP Transmitted Packets                         0
                            0
14 LSN UDP Transmitted Bytes                           0
                            0
15 LSN UDP Dropped Packets                             0
                            0
16 LSN UDP Current Sessions                            0
                            0
17 LSN ICMP Received Packets                           0
       982
18 LSN ICMP Received Bytes                             0
       96236
19 LSN ICMP Transmitted Packets                        0
                            0
20 LSN ICMP Transmitted Bytes                          0
                            0
21 LSN ICMP Dropped Packets                            0
       982
22 LSN ICMP Current Sessions                           0
                            0
23 LSN Subscribers                                     0
                            1
24
25 Done
26
27 > stat lsn group LSN-GROUP-1
28
```



```
29 LSN Group Statistics
30                                                    Rate (/s)
                                                         Total
31 TCP Translated Pkts                                  0
      40
32 TCP Translated Bytes                                 0
      3026
33 TCP Dropped Pkts                                     0
                          0
34 TCP Current Sessions                                 0
                          0
35 UDP Translated Pkts                                  0
                          0
36 UDP Translated Bytes                                 0
                          0
37 UDP Dropped Pkts                                     0
                          0
38 UDP Current Sessions                                 0
                          0
39 ICMP Translated Pkts                                 0
                          0
40 ICMP Translated Bytes                                0
                          0
41 ICMP Dropped Pkts                                    0
                          0
42 ICMP Current Sessions                                0
                          0
43 Current Subscribers                                  0
                          1
44
45 Done
46 <!--NeedCopy-->
```

Parameter Descriptions (of commands listed in the CLI procedure)

- stat lsn group
  - groupname

    Name of the LSN Group. Maximum Length: 127
  - detail

    Specifies detailed output (including more statistics). The output can be quite voluminous. Without this argument, the output will show only a summary.

- fullValues

  Specifies that numbers and strings should be displayed in their full form. Without this option, long strings are shortened and large numbers are abbreviated.

- ntimes

  The number of times, in intervals of seven seconds, the statistics should be displayed.

  Default value: 1

- logFile

  The name of the log file to be used as input.

- clearstats

  Clear the statsistics / counters

  Possible values: basic, full

Compact Logging

Logging LSN information is one of the important functions needed by ISPs to meet legal requirements and be able to identify the source of traffic at any given time. This eventually results in a huge volume of log data, requiring the ISPs to make large investments to maintain the logging infrastructure.

Compact logging is a technique for reducing the log size by using a notational change involving short codes for event and protocol names. For example, C for client, SC for session created, and T for TCP. Compact logging results in an average of 40 percent reduction in log size.

The following examples of NAT44 mapping creation log entries show the advantage of compact logging.

Default logging format	02/02/2016:01:13:01 GMT Informational 0-PPE-2 : default LSN LSN_ADDRPORT_MAPPING 85 0 : A&PDM CREATED ClientIP:Port:TD1.1.1.1:6500:0,NatIP:NatPort8.8.8.8:47902, DestinationIP:Port:TD2.2.2.2:80:0, Protocol: TCP
Compact logging format	02/02/2016:01:14:57 GMT Info 0-PE2:default LSN 87 0:A&PDMC|C-1.1.1.1:6500:0|N-8.8.8.9:51066|D-2.2.2.2:80:0|T

Configuration Steps

Perform the following tasks for logging LSN information in compact format:

- **Create an LSN log profile.** An LSN log profile includes the Log Compact parameter, which specifies whether to or not to log information in compact format for an LSN configuration.
- **Bind the LSN log profile to an LSN group of an LSN configuration.** Bind the created LSN log profile to an LSN group of an LSN configuration by setting the Log Profile Name parameter to the created LSN log profile name. All sessions and mappings for this LSN group are logged in compact format.

To create an LSN log profile by using the CLI

At the command prompt, type:

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

To bind an LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

Sample configuration:

```
add lsn logprofile  LOG-PROFILE-COMPACT-9 -logCompact ENABLED

Done
add lsn client LSN-CLIENT-9
Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-9
Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
Done
```



```
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
Done
bind lsn group  LSN-GROUP-9  - logProfileName LOG-PROFILE-COMPACT-9
Done
<!--NeedCopy-->
```

IPFIX Logging

The Citrix ADC appliance supports sending information about LSN events in Internet Protocol Flow Information Export (IPFIX) format to the configured set of IPFIX collector(s). The appliance uses the existing AppFlow feature to send LSN events in IPFIX format to the IPFIX collectors.

IPFIX based logging is available for the following large scale NAT44 related events:

- Creation or deletion of an LSN session.
- Creation or deletion of an LSN mapping entry.
- Allocation or de-allocation of port blocks in the context of deterministic NAT.
- Allocation or de-allocation of port blocks in the context of dynamic NAT.
- Whenever subscriber session quota is exceeded.

Points to Consider before you Configure IPFIX logging

Before you start configuring IPSec ALG, consider the following points:

- You must configure the AppFlow feature and IPFIX collector(s) on the Citrix ADC appliance. For instructions, see Configuring the AppFlow feature topic.

Configuration Steps

Perform the following tasks for logging LSN information in IPFIX format:

- **Enable LSN logging in the AppFlow configuration**. Enable the LSN logging parameter as part of AppFlow configuration.
- **Create an LSN log profile**. An LSN log profile includes the IPFIX parameter that enables or disables the log information in IPFIX format.
- **Bind the LSN log profile to an LSN group of an LSN configuration**. Bind the LSN log profile to one or multiple LSN group(s). Events related to the bound LSN group will be logged in IPFIX format.

To enable LSN logging in the AppFlow configuration by using the CLI

At the command prompt, type:


```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

To create an LSN log profile by using the CLIAt the command prompt

At the command prompt, type:

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind the LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

To create an LSN log profile by using the GUI

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **Log** tab, and then add a log profile.

To bind the LSN log profile to an LSN group of an LSN configuration by using the GUI

1. Navigate to **System** > **Large Scale NAT** > **LSN Group,** open the **LSN** group.
2. In **Advanced Settings,** click **+ Log Profile** to bind the created Log profile to the LSN group.

TCP SYN Idle Timeout

September 14, 2021

SYN idle timeout is the timeout for establishing TCP connections that use LSN on the Citrix ADC appliance. If a TCP session is not established within the configured timeout period, the Citrix ADC removes

the session. SYN idle timeout is useful in providing protection against SYN flood attacks. In an LSN configuration, the LSN group entity includes the SYN idle timeout setting.

Example:

In the following sample LSN configuration, SYN idle timeout is set to 30 secs for TCP connections related to subscribers from the 192.0.2.0/24 network.

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  - synidletimeout 30

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
<!--NeedCopy-->
```

Overriding LSN configuration with Load Balancing Configuration

September 14, 2021

An LSN configuration takes precedence over any load balancing configuration by default. For overriding the large scale networking (LSN) configuration with the load balancing configuration for traffic matching both configurations, create a net profile with Override LSN parameter enabled and bind this profile to the virtual server of the load balancing configuration. USNIP or USIP settings of the load balancing configuration are applied to the traffic, instead of applying the LSN IP address of the LSN configuration.

This option is useful in an LSN deployment that includes Citrix ADC appliances and value added services, such as firewall and optimization devices. In this type of deployment, the ingress traffic on the Citrix ADC appliance is required to pass through these value-added services before an LSN configuration on the appliance is applied to the traffic. For the Citrix ADC appliance to send the ingress traffic to a value added service, a load balancing configuration is created and override LSN is enabled on the appliance. The load balancing configuration includes value added services, represented as load balancing services, bound to a virtual server of type ANY. The virtual server is configured with listen policies for identifying the traffic to be sent to the value added service.

To enable override lsn in a net profile by using the CLI

To enable override lsn while adding a net profile, at the command prompt, type

```
add netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable override lsn while adding a net profile, at the command prompt, type

```
set netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable override lsn in a net profile by using GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the **Override LSN** parameter while adding or modifying net profiles.

In the following sample configuration, net profile NETPROFILE-OVERRIDELSN-1 has override LSN option enabled and is bound to load balancing virtual server LBVS-1.

Sample configuration:

```
add netprofile NETPROFILE-OVERRIDELSN-1 -overrideLsn ENABLED

Done

set lb vserver LBVS-1 -netprofile NETPROFILE-OVERRIDELSN-1

Done
```



```
<!--NeedCopy-->
```

Clearing LSN Sessions

September 14, 2021

You can remove any unwanted or inefficient LSN sessions from the Citrix ADC appliance. The appliance immediately releases resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected LSN sessions from the Citrix ADC appliance.

To clear all LSN sessions by using the command line interface

At the command prompt, type:

```
flush lsn session

show lsn session
<!--NeedCopy-->
```

To clear selective LSN sessions by using the command line interface

At the command prompt, type:

```
flush lsn session [-clientname <string>] [-network <ip_addr> [-netmask
   <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
   port>]]

show lsn session
<!--NeedCopy-->
```

Example

Clear all LSN sessions existing on a Citrix ADC

```
flush lsn session

Done
<!--NeedCopy-->
```


Clear all LSN sessions related to LSN client entity LSN-CLIENT-1

```
flush lsn session -clientname LSN-CLIENT-1

Done
<!--NeedCopy-->
```

Clear all LSN sessions related to a subscriber network (192.0.2.0) of LSN client entity LSN-CLIENT-2 belonging to traffic domain 100

```
flush lsn session -clientname LSN-CLIENT-2 -network 192.0.2.0 -
   netmask 255.255.255.0 -td 100

Done
<!--NeedCopy-->
```

To clear all LSN sessions by using the configuration utility

Navigate to System > Large Scale NAT > Sessions, and click Flush Sessions.

Parameter Descriptions (of commands listed in the CLI procedure)

- flush lsn session
  - clientname

    Name of the LSN Client entity. Maximum Length: 127
  - network

    IP address or network address of subscriber(s).
  - netmask

    Subnet mask for the IP address specified by the network parameter.

    Default value: 255.255.255.255
  - td

    Traffic domain ID of the LSN client entity.

    Default value: 0

    Minimum value: 0

    Maximum value: 4094
  - natIP

    Mapped NAT IP address used in LSN sessions.

- natPort

  Mapped NAT port used in the LSN sessions.

Load Balancing SYSLOG Servers

September 14, 2021

The Citrix ADC appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the Citrix ADC appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

```
add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
    SYSLOGUDP)> <port>
<!--NeedCopy-->
```

Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```

Bind the service to the load balancing virtual server.

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

1. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
<!--NeedCopy-->
```

Add a SYSLOG policy by specifying the rule and action.


```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

Bind the SYSLOG policy to the system global for the policy to take effect.

```
bind system global <policyName>
<!--NeedCopy-->
```

Load balancing of SYSLOG servers using the configuration utility

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

   Navigate to Traffic Management > Services, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

   Navigate to Traffic Management > Virtual Servers, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

3. Bing the service to the load balancing virtual server to the service.

   Bing the service to the load balancing virtual server.

   Navigate to Traffic Management > Virtual Servers, select a virtual server and then selectAUDITLOGHASH in the Load Balancing Method.

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   Navigate to System > Auditing, click Servers and add a server by selecting LB Vserver option inServers.

5. Add a SYSLOG policy by specifying the rule and action.

   Navigate to System > Syslog, click Policies and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   Navigate to System > Syslog, select a SYSLOG policy and click Action, and then click Global Bindings and bind the policy to system global.

Example:

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. The Citrix ADC appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.


```
add service service1 192.0.2.10 SYSLOGUDP 514

add service service2 192.0.2.11 SYSLOGUDP 514

add service service3 192.0.2.11 SYSLOGUDP 514

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH

bind lb vserver lbvserver1 service1

bind lb vserver lbvserver1 service2

bind lb vserver lbvserver1 service3

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All

add syslogpolicy syspol1 ns_true sysaction1

bind system global syspol1
<!--NeedCopy-->
```

Limitations:

The Citrix ADC appliance does not support an external load balancing virtual server load balancing the SYSLOG messages among the log servers.

Port Control Protocol

September 14, 2021

Citrix ADC appliances now support Port Control Protocol (PCP) for large scale NAT (LSN). Many of an ISP's subscriber applications must be accessible from Internet (for example, Internet of Things (IOT) devices, such as an IP camera that provides surveillance over the Internet). One way to meet this requirement is to create static large scale NAT (LSN) maps. But for a very large number of subscribers, creating static LSN NAT maps is not a feasible solution.

Port Control Protocol (PCP) enables a subscriber to request specific LSN NAT mappings for itself and/or for other 3rd party devices. The large scale NAT device creates an LSN map and sends it to the subscriber. The subscriber sends the remote devices on the Internet the NAT IP address:NAT port at which they can connect to the subscriber.

Applications usually send frequent keep-alive messages to the large scale NAT device so that their LSN mappings do not time out. PCP helps reduce the frequency of such keep-alive messages by enabling

the applications to learn the timeout settings of the LSN mappings. This helps reduce bandwidth consumption on the ISP's access network and battery consumption on mobile devices.

PCP is a client-server model and runs over the UDP transport protocol. A Citrix ADC appliance implements the PCP server component and is compliant with RFC 6887.

Configuration Steps

Perform the following tasks for configuring PCP:

- (Optional) Create a PCP profile. A PCP profile includes settings for PCP related parameters (for example, to listen for mapping and peer PCP requests). A PCP profile can be bound to a PCP server. A PCP profile bound to a PCP server applies all its settings to the PCP server. A PCP profile can be bound to multiple PCP servers. By default, one PCP profile with default parameters settings is bound to all PCP servers. A PCP profile that you bind to a PCP server overrides the default PCP profile settings for that server. A default PCP profile has the following parameter settings:
  - Mapping: Enabled
  - Peer: Enabled
  - Minimum map life: 120 seconds
  - Maximum max life: 86400 seconds
  - Announce count: 10
  - Third Party: Disabled
- Create a PCP server and bind a PCP profile to it. Create a PCP server on the Citrix ADC appliance to listen for PCP related requests and messages from the subscribers. A Subnet IP (SNIP) address must be assigned to a PCP server to access it. By default, a PCP server listens on port 5351.
- Bind the PCP server to an LSN group of an LSN configuration. Bind the created PCP server to an LSN group of an LSN configuration by setting the PCP Server parameter to specify the created PCP server. The created PCP server can be accessed only by the subscribers of this LSN group.

> **Note**
>
> A PCP server for a large scale NAT configuration does not serve requests from subscribers that are identified from ACL rules.

To create a PCP profile by using the CLI

At the command prompt, type:

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]
```



```
2
3 show pcp profile <name>
4 <!--NeedCopy-->
```

To create a PCP server by using the CLI

At the command prompt, type:

```
1 add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
     string>]
2
3 show pcp server <name>
4 <!--NeedCopy-->
```

Sample Configuration for NAT44

In the following sample configuration, PCP server PCP-SERVER-9, with default PCP settings, is bound to LSN group LSN-GROUP-9. PCP-SERVER-9 serves PCP requests from subscribers in network 192.0.2.0/24.

Sample configuration:

```
1 add pcp server PCP-SERVER-9 192.0.3.9
2
3 Done
4
5 add lsn client LSN-CLIENT-9
6
7 Done
8
9 bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0
10
11 Done
12
13 add lsn pool LSN-POOL-9
14
15 Done
16
17 bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4
18
19 Done
20
21 add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
```



```
22
23 Done
24
25 bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
26
27 Done
28
29 bind lsn group  LSN-GROUP-9  -pcpServer PCP-SERVER-9
30
31 Done
32 <!--NeedCopy-->
```

LSN44 in a cluster setup

September 14, 2021

Large scale NAT44 configurations are supported on a Citrix ADC cluster setup.

A Citrix ADC cluster is a group of Citrix ADC appliances that are configured and managed as a single system. A Citrix ADC cluster provides scalability and availability. Each Citrix ADC appliance in a cluster setup acts as an independent LSN entity and is managed as a single system.

The LSN configuration in a cluster setup is same as in a standalone appliance except a specific pool of LSN IP addresses are owned by only one node at a time. In other words, an LSN IP pool entity is configured as a spotted entity in a particular node. All the nodes of a cluster setup can have a specific LSN IP pool entity. To make sure that the packets related to an LSN session are received on the same cluster node that performed the NAT operation, policy based backplane (PBS) steering is configured. PBS steers the received related packets of an LSN session to the same cluster node.

Sample configuration:

```
1  add lsn client LSN-CLIENT-1
2
3  Done
4
5  bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-1
10
11 Done
12
```



```
bind lsn pool LSN-POOL-1 -ownerNode 1 203.0.113.3

Done

bind lsn pool LSN-POOL-1 -ownerNode 2 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

add ns acl b1 ALLOW -srcIP = 192.0.2.0-192.0.2.255 -type DFD -dfdhash
   SIP


Done

apply ns acls -type DFD

Done
<!--NeedCopy-->
```

Dual-Stack Lite

September 14, 2021

Because of the shortage of IPv4 addresses, and the advantages of IPv6 over IPv4, many ISPs have started transitioning to IPv6 infrastructure. But during the transition, ISPs must continue to support IPv4 along with IPv6, because most of the public Internet still uses only IPv4, and many subscribers do not support IPv6.

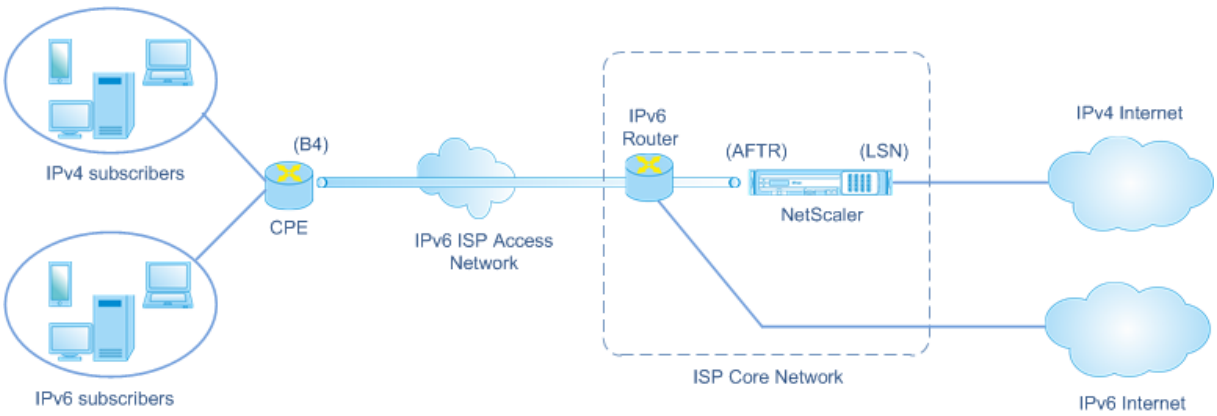
Dual Stack Lite (DS-Lite) is an IPv6 transition solution for ISPs with IPv6 infrastructure to connect their IPv4 subscribers to the Internet. DS-Lite uses IPv4-in-IPv6 tunneling to send a subscriber's IPv4 packet through a tunnel on the IPv6 access network to the ISP. The IPv6 packet is decapsulated to recover the subscriber's IPv4 packet and is then sent to the Internet after NAT address and port translation and other LSN related processing. The response packets traverse through the same path to the subscriber.

The Citrix ADC appliance implements the AFTR component of a DS-Lite deployment and is compliant with RFC 6333.

Architecture

The Dual-Stack Lite architecture for an ISP consists of the following components:

- **Basic Bridging Broadband (B4).** Basic Bridging broadband, or B4, is a device or component that resides in the subscriber premises. Typically, B4 is a component in the CPE devices in the subscriber premises. IPv4 subscribers are connected to the IPv6-only ISP access network through the CPE device containing the B4 component. The main function of the B4 is to initiate an IPv6 tunnel between B4 and an address family transition router (AFTR) in order to send or receive subscriber IPv4 request or response packets over the tunnel. B4 includes an IPv6 address known as the B4 tunnel endpoint address. B4 uses this address to source IPv6 packets to AFTR and receive packets from AFTR.
- **Address family transition router (AFTR).** AFTR is a device or component residing in the ISP's core network. AFTR terminates the IPv6 tunnel from the B4 device. In other words, the IPv6 tunnel is formed between B4 in the subscriber premise and AFTR in ISP core network. AFTR decapsulates IPv6 packets received from B4 to recover the subscribers' original IPv4 packets. AFTR sends the IPv4 packets to the LSN device or component. LSN routes the IPv4 packets to their destination after performing NAT address and port translation (NAT 44) and other LSN related processing. AFTR includes an IPv6 address known as the AFTR tunnel endpoint address. AFTR uses this address to source IPv6 packets to B4 and receive IPv6 packets from B4. The Citrix ADC appliance implements the AFTR component.
- **Softwire.** The IPv6 tunnel created between B4 and AFTR is called a softwire.



The DS-Lite architecture of an ISP using a Citrix ADC appliance consists of subscribers in private address spaces accessing the Internet through a Citrix ADC appliance deployed in ISP's core network. IPv4 subscribers are connected to a CPE device that includes the DS-Lite B4 functionality. The CPE device is connected to the ISP core network through ISP's IPv6-only access network. The Citrix ADC appliance contains the DS-Lite AFTR and LSN functionality.

IPv4 subscribers connected to the CPE device are assigned private IPv4 addresses either manually or through DHCP server running on the CPE device. On the CPE device, the AFTR tunnel endpoint address is specified manually or through DHCPv6. Configuration of CPE devices is vendor specific and therefore outside the scope of this documentation.

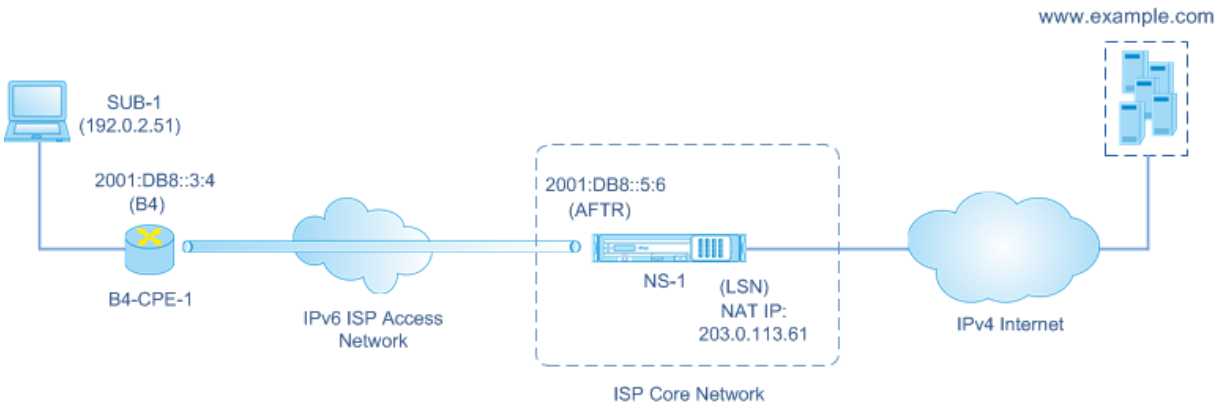
Upon receiving a request packet that is from an IPv4 subscriber and destined to a location on the Internet, the B4 component of the CPE device encapsulates the IPv4 packet in an IPv6 packet and sends it to the Citrix ADC appliance in the ISP core network. The Citrix ADC appliance's AFTR functionality decapsulates the IPv6 packet to recover the subscriber's original IPv4 packet. The LSN functionality of the Citrix ADC appliance translates the source IP address and port of the IPv4 packet to an NAT IP address and NAT port selected from the configured NAT pool, and then sends the packet to its destination on the Internet.

The appliance maintains a record of all active sessions that use the AFTR and LSN functionalities. These sessions are called DS-Lite sessions. The Citrix ADC appliance also maintains the mappings between B4 IPv6 address, subscriber IPv4 address and port, and NAT IPv4 address and port, for each DS-Lite session. These mappings are called DS-Lite LSN mappings. From DS-Lite session entries and DS-Lite LSN mapping entries, the Citrix ADC appliance recognizes a response packet (received from the Internet) as belonging to a particular DS-Lite session.

When the Citrix ADC appliance receives a response packet belonging to a particular DS-Lite session, the appliance's LSN functionality translates the destination IP address and port of the response packet from NAT IP address and port to the subscriber IP address and port, the AFTR functionality encapsulates the resulting packet in an IPv6 packet and sends it to the CPE device. The B4 functionality of the CPE device decapsulates the IPv6 packet to recover the IPv4 response packet, and then sends the IPv4 packet to the subscriber.

Example

Consider an example of a DS-Lite deployment consisting of Citrix ADC NS-1 in an ISP's core network, CPE device B4-CPE-1 in a subscriber premise, and a single IPv4 subscriber SUB-1. B4-CPE-1 supports the B4 functionality of DS-Lite feature.



The following table lists the settings used in this example.

Entity	Name	Details
IPv4 address of subscriber SUB-1		192.0.2.51
IPv6 address of softwire endpoint on the B4 device (B4-CPE-1)		2001:DB8::3:4
IPv6 address of the softwire endpoint on the AFTR device (NS-1)		2001:DB8::5:6

Settings on Citrix ADC appliance NS-1:

Entity	Name	Details
LSN client	LSN-DSLITE-CLIENT-1	Network6 (Identifying traffic from B4 devices) = 2001:DB8::3:0/100
LSN pool	LSN-DSLITE-POOL-1	LSN IPs (NAT IP) = 203.0.113.61 - 203.0.113.70
IPv6 Profile	LSN-DSLITE-PROFILE-1	Type = DS-LITE; IPv6 address (AFTR IPv6 address) = One of the Citrix ADC owned IPv6 address of type SNIP6 = 2001:DB8::5:6
LSN group	LSN-DSLITE-GROUP-1	LSN client = LSN-DSLITE-CLIENT-1; LSN pool = LSN-DSLITE-POOL-1;IPv6 profile = LSN-DSLITE-PROFILE-1

Following is the traffic flow in this example:

1. IPv4 subscriber SUB-1 sends a request to (`http://www.example.com/`). The IPv4 packet has:
    - Source IP address = 192.0.2.51

- Source port = 2552
- Destination IP address = 198.51.100.250
- Destination port = 80

2. Upon receiving the IPv4 request packet, B4-CPE-1 encapsulates it in the payload of an IPv6 packet and then sends the IPv6 packet to NS-1. The IPv6 packet has:
   - Source IP address = 2001:DB8::3:4
   - Destination IP address = 2001:DB8::5:6
3. When NS-1 receives the IPv6 packet, the AFTR module decapsulates the packet by removing the IPv6 headers. The resulting packet is SUB-1's original IPv4 request packet.
4. The LSN module of NS-1 translates the source IP address and port of the packet to an NAT IP address and NAT port selected from the configured NAT pool. The translated IPv4 packet has:
   - Source IP address = 203.0.113.61
   - Source port = 3002
   - Destination IP address = 198.51.100.250
   - Destination port = 80
5. The LSN module also creates an LSN mapping and session entry for this DS Lite session. The mapping includes the following information:
   - Source IP address of the IPv6 packet (B4-CPE-1's IPv6 address) = 2001:DB8::3:4
   - Source IP address of the IPv4 packet (SUB-1's IPv4 address) = 192.0.2.51
   - Source port of the IPv4 packet = 2552
   - NAT IP address = 203.0.113.61
   - NAT port = 3002
6. NS-1 sends the resulting IPv4 packet to its destination on the Internet.
7. The server for www.example.com processes the request packet and sends a response packet. The IPv4 response packet has:
   - Source IP address = 198.51.100.250
   - Source port = 80
   - Destination IP address = 203.0.113.61
   - Destination port = 3002
8. Upon receiving the IPv4 packet, NS-1 examines the LSN mapping and session entries and finds that the IPv4 response packet belongs to a DS Lite session. The LSN module of NS-1 translates the destination IP address and port. The IPv4 packet now has:
   - Source IP address = 198.51.100.250
   - Source port = 80
   - Destination IP address = 192.0.2.51

- Destination port = 2552

9. The AFTR module of NS-1 encapsulates the IPv4 packet in an IPv6 packet and then sends the IPv6 packet to B4-CPE-1. The IPv6 packet has:
   - Source IP address = 2001:DB8::5:6
   - Destination IP address = 2001:DB8::3:4
10. Upon receiving the packet, B4-CPE-1 decapsualtes the IPv6 packet by removing the IPv6 headers, and then sends the resulting IPv4 packet to CL-1.

Points to Consider before Configuring DS-Lite

September 14, 2021

Consider the following points before configuring DS-Lite on a Citrix ADC appliance:

1. You must understand the different components of DS-Lite, described in RFC 6333.
2. A DS-lite configuration on a Citrix ADC appliance uses the LSN commands sets. In a DS-Lite configuration, the LSN client entity specifies the IPv6 address or IPv6 network address or ACL6 rules for identifying the traffic from the B4 device. A DS-Lite configuration also includes an IPv6 profile, which specifies the IPv6 address AFTR component on a Citrix ADC appliance. For more information on Citrix ADC LSN feature, see Large Scale NAT.
3. For a DS-Lite configuration, the Citrix ADC appliance supports LSN for IPv4 packets that belong to one of the following protocols only. The Citrix ADC appliance drops IPv4 packets belonging to other protocols:
   - TCP
   - UDP
   - ICMP
4. The Citrix ADC appliance supports the following ALGs DS-Lite:
   - ICMP
   - FTP
   - TFTP
   - Session Initiation Protocol (SIP)
   - Real Time Streaming Protocol (RTSP)

Configuring DS-Lite

September 14, 2021

A DS-lite configuration on a Citrix ADC appliance uses the LSN commands sets. In a DS-Lite configuration, the LSN client entity specifies the IPv6 address or IPv6 network address or ACL6 rules for identifying the traffic from the B4 device. For more information on the Citrix ADC LSN feature, see Large Scale NAT. A DS-Lite configuration also includes an IPv6 profile, which specifies the IPv6 address (of type SNIP6) of the DS-Lite AFTR component on a Citrix ADC appliance.

Configuring DS-Lite on a Citrix ADC appliance consists of the following tasks:

- **Set the global LSN parameters**. Global parameters include the amount of Citrix ADC memory reserved for the LSN feature and synchronization of LSN sessions in a high availability setup.
- **Create an LSN client entity for identifying traffic from B4 CPE devices**. The LSN client entity refers to a set of DS-Lite B4 devices. The client entity includes IPv6 addresses or IPv6 network address or ACL6 rules for identifying the traffic from these B4 devices. An LSN client can be bound to only one LSN group. The command line interface has two commands for creating an LSN client entity and binding a subscriber to the LSN client entity. The configuration utility combines these two operations on a single screen.
- **Create an LSN pool and bind NAT IP addresses to it**. An LSN pool defines a pool of NAT IP addresses to be used by the Citrix ADC appliance to perform LSN. The command line interface has two commands for creating an LSN pool and binding NAT IP addresses to the LSN pool. The configuration utility combines these two operations on a single screen.
- **Create an LSN IP6 profile.** An LSN IP6 profile defines the IPv6 address of the DS-Lite AFTR component on the Citrix ADC appliance. The IPv6 address must be one of the Citrix ADC owned IPv6 address of type SNIP6.
- **(Optional) Create an LSN Transport Profile for a specified protocol**. An LSN transport profile defines various timeouts and limits, such as maximum LSN sessions and maximum ports usage that a subscriber can have for a given protocol. You bind an LSN transport profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. A profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called the default transport profile. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
- **(Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it**. An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination

ports, you bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The configuration utility combines these two operations on a single screen.

- **Create an LSN Group and bind LSN pools, LSN IPv6 profile, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group**. An LSN group is an entity consisting of an LSN client, an LSN IPv6 profile, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only one LSN IPv6 profile can be bound to an LSN group, and an LSN IPv6 profile bound to an LSN group cannot be bound to other LSN groups. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN group and binding LSN pools, LSN transport profiles, and LSN application profiles to the LSN group. The configuration utility combines these two operations in a single screen.

Configuration by Using the Command Line

To create an LSN client by using the command line interface:

At the command prompt, type:

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

To bind an IPv6 network or an ACL6 rule to an LSN client by using the command line interface:

At the command prompt, type:

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

```



```
3 show lsn client
4 <!--NeedCopy-->
```

To create an LSN pool by using the command line interface:

At the command prompt, type:

```
1 add lsn pool <poolname> [-nattype ( DYNAMIC )] [-portblockallocation (
     ENABLED | DISABLED )] [-portrealloctimeout <secs>] [-
     maxPortReallocTmq <positive_integer>]
2
3 show lsn pool
4 <!--NeedCopy-->
```

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

At the command prompt, type:

```
1 bind lsn pool <poolname> <lsnip>
2
3 show lsn pool
4 <!--NeedCopy-->
```

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

To configure an LSN IPv6 profile by using the command line interface:

At the command prompt, type:

```
1 add lsn ip6profile <name>  -type DS-Lite  -network6 < ipv6_addr|*s >
2
3 show lsn ip6profile
4 <!--NeedCopy-->
```

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

At the command prompt, type:

```
1 add lsn transportprofile <transportprofilename> <transportprotocol> [-
     sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
     positive_integer>] [-sessionquota <positive_integer>] [-
     portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
     ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]
2
3 show lsn transportprofile
4 <!--NeedCopy-->
```


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

At the command prompt, type:

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

show lsn appsprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

To create an LSN group by using the command line interface:

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC )]
    [-portblocksize <positive_integer>] [-logging (ENABLED | DISABLED )]
     [-sessionLogging ( ENABLED | DISABLED )][-sessionSync ( ENABLED |
    DISABLED )] [-snmptraplimit<positive_integer>] [-ftp ( ENABLED |
    DISABLED )] [-pptp ( ENABLED |DISABLED )] [-sipalg ( ENABLED |
    DISABLED )] [-rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```


Configuration by Using the Configuration Utility

To configure an LSN client and bind an IPv6 network address or an ACL6 rule by using the configuration utility:

Navigate to **System** > **Large Scale NAT** > **Clients**, and add a client and then bind an IPv6 network address or an ACL6 rule to the client.

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

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

To configure an LSN IPv6 profile by using the configuration utility:

Navigate to **System** > **Large Scale NAT** > **Profiles**, click the **IPv6** tab, and assign an IPv6 address for DS-Lite AFTR.

To configure an LSN transport profile by using the configuration utility:

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

To configure an LSN application profile by using the configuration utility:

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

To configure an LSN group and bind an LSN client, an LSN IPv6 profile, pools, transport profiles, and application profiles by using the configuration utility:

Navigate to **System** > **Large Scale NAT** > **Groups**, and add a group and then bind an LSN client, an LSN IPv6 profile, pools, transport profiles, and application profiles to the group.

```
1  > add lsn client LSN-DSLITE-CLIENT-1
2  Done
3  > bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
4  Done
5  > add lsn pool LSN-DSLITE-POOL-1
6  Done
7  > bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
8  Done
9  > add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
       DB8::5:6
10 Done
11 > add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
       portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1
12 Done
13 > add lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
```



```
14 Done
```

Logging and Monitoring DS-Lite

You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The Citrix ADC appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a Citrix ADC appliance whose NSIP address is 10.102.37.115.You can

log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The Citrix ADC appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a Citrix ADC appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry

DS-Lite session creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite session deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN mapping creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN mapping deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

Displaying Current DS-Lite Sessions

You can display the current DS-Lite sessions for detecting any unwanted or inefficient sessions on the Citrix ADC appliance. You can display all or some DS-Lite sessions, on the basis of selection parameters.

Configuration by Using the Command Line Interface

To display all DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

To display selected DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
   ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
   ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

Example:

The following sample ouput displays all DS-Lite sessions existing on a Citrix ADC appliance:

```
show lsn session  -nattype DS-Lite
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1.  2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
   3002 TCP OUT

2.  2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
   52862 TCP OUT

3.  2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
   48116 ICMP OUT

4.  2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
   48305 TCP OUT

Done
<!--NeedCopy-->
```

Configuration Using the Configuration Utility

To display all or selected DS-Lite sessions by using the configuration utility

1. **Navigate to System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. For displaying DS-Lite sessions on the basis of selection parameters, click **Search**.

Clearing DS-Lite Sessions

You can remove any unwanted or inefficient DS-Lite sessions from the Citrix ADC appliance. The appliance immediately releases the resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected DS-Lite sessions from the Citrix ADC appliance.

To clear all DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
flush lsn session -nattype DS-Lite
show lsn session -nattype DS-Lite
```

To clear selected DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
1 flush lsn session  - nattype DS-Lite [-clientname <string>] [-network <
     ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
     ip_addr> [-natPort <port>]]
2
3 show lsn session  - nattype DS-Lite
4 <!--NeedCopy-->
```

To clear all or selected DS-Lite sessions by using the configuration utility:

1. Navigate to **System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. Click **Flush Sessions**.

Configuring DS-Lite Static Maps

September 14, 2021

The Citrix ADC appliance supports manual creation of DS-Lite LSN mappings, which contain the mapping between the following information:

- Subscriber's IP address and port, and IPv6 address of B4 device or component
- NAT IP address and port

Static DS-Lite LSN mappings are useful in cases where you want to ensure that the connections initiated to a NAT IP address and port map to the subscriber IP address and port through the specified B4 device (for example, web servers located in the internal network).

Note: This feature is supported in release 11.0 build 64.x and later.

To create a DS-Lite static LSN mapping by using the command line

At the command prompt, type:

```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
    <positive_integer>] [-network6 <B4_ADDR>] [<natIP> [<natPort>]] [-
    destIP<ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

Parameter Descriptions

add lsn static

- name

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

- transportprotocol

Protocol for the DS-Lite LSN mapping entry.

- subscrIP

IPv4 address of a subscriber for the DS-Lite LSN mapping entry.

- subscrPort

Port of the subscriber for the DS-Lite LSN mapping entry.

- Network6

IPv6 address of the B4 device or component.

- td

ID of the traffic domain to which the B4 device belongs. The IPv6 address of the B4 device is specified in the network6 paramter. If you do not specify an ID, the B4 device is assumed to be a part of the default traffic domain.

- natIP

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

- natPort

NAT port for this DS-Lite LSN mapping entry.

- destIP

Destination IP address for the DS-Lite LSN mapping entry.

- dsttd

ID of the traffic domain through which the destination IP address for this DS-Lite LSN mapping entry is reachable from the Citrix ADC appliance. If you do not specify an ID, the destination IP address is assumed to be reachable through the default traffic domain, which has an ID of 0.

To create a DS-Lite static LSN mapping by using the configuration utility

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

Configuring Deterministic NAT Allocation for DS-Lite

September 14, 2021

Deterministic NAT allocation for DS-Lite LSN deployments is a type of NAT resource allocation in which the Citrix ADC appliance pre-allocates, from the LSN NAT IP pool and on the basis of the specified port block size, an LSN NAT IP address and a block of ports to each subscriber (subscriber behind B4 device).

Note: This feature is supported in release 11.0 build 64.x and later.

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

The Citrix ADC appliance logs the allocated NAT IP address and the port block for a subscriber. For a connection, a subscriber can be identified by just its mapped NAT IP address and port block. For this reason, the Citrix ADC appliance does not log the creation or deletion of an LSN session.

A DS-Lite subscriber can have only one deterministic port block. If the entire block of ports is being used, the Citrix ADC appliance drops any new connection from the subscriber.

Example: Deterministic DS-Lite

In this example, a deterministic DS-Lite configuration includes four subscribers with IP addresses 192.0.17.5, 192.0.17.6, 192.0.17.7, and 192.0.17.8. These ipv4 subscribers are behind a B4 device having the IPv6 address 2001:DB8::3:4. In this configuration, the port block size is set to 20480 and LSN NAT IP address pool has IP addresses in the range 203.0.113.41-203.0.113.42.

The Citrix ADC appliance sequentially pre-allocates, from the LSN NAT IP pool and on the basis of the set port block size, an LSN NAT IP address and a block of ports to each subscriber. It assigns the first block of ports (1024-21503) on the beginning NAT IP address (203.0.113.41) to the beginning subscriber IP address (192.0.17.5). The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT IP address is assigned to the subscriber, and so on. The Citrix ADC logs the NAT IP address and the block of ports allocated for each subscriber.

The Citrix ADC appliance does not log any LSN session created or deleted for these subscribers.

The following table lists the NAT IP address and blocks of ports allocated to each subscriber in this example:

Subscriber IP address	Allocated NAT IP address	Allocated Block of Ports	IPv6 address of B4
192.0.17.5	203.0.113.41	1024 - 21503	2001:DB8::3:4
192.0.17.6	203.0.113.41	21504 - 41983	2001:DB8::3:4
192.0.17.7	203.0.113.41	41984 - 62463	2001:DB8::3:4
192.0.17.8	203.0.113.42	1024 - 21503	2001:DB8::3:4

Configuration Steps

You need to configure deterministic NAT as part of the DS-Lite configuration. For instructions on configuring DS-Lite, see Configuring DS-Lite.

While configuring DS-Lite, make sure that you:

- Set the NAT Type parameter to Deterministic when adding the LSN pool and the LSN group.
- Set the desired port block size parameter when adding the LSN group, unless you can accept the default value.

Points to Consider before Configuring Deterministic DS-Lite

Consider the following points before configuring deterministic DS-Lite:

- The complete IP address of each subscriber must be specified in a separate add lsn client command, by setting the Network and Netmask parameters. (Set Netmask to 255.255.255.255.) Also the IPv4 address of the B4 device specified in Network6 parameter must be complete (/128 prefix). In other words, Network and Network6 parameter do not accept addresses other than /32 bit mask and /128 prefix, respectively.
- The Citrix ADC appliance drops connections from subscribers that are not specified in any deterministic DS-Lite configuration but are behind B4 devices specified in a deterministic DS-lite configuration.
- The Citrix ADC appliance recognizes subscribers having the same IPv4 address as different subscribers if they are behind different B4 devices. A combination of subscriber IPv4 address and B4 device defines a unique subscriber in the LSN client entity of a DS-Lite configuration.

Sample Deterministic DS-Lite Configuration:

The following configuration uses the settings listed in section Example: Deterministic DS-Lite.

```
1  add lsn client LSN-DSLITE-CLIENT-10
2
3  Done
4  bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.5 -netmask
       255.255.255.255 -network6 2001:DB8::3:4/128
5
6  Done
7  bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.6 -netmask
       255.255.255.255 -network6 2001:DB8::3:4/128
8
9  Done
10 bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.7 -netmask
       255.255.255.255 -network6 2001:DB8::3:4/128
11
12 Done
13 bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.8 -netmask
       255.255.255.255 -network6 2001:DB8::3:4/128
14
15 Done
16 add lsn pool LSN-DSLITE-POOL-10 -nattype DETERMINISTIC
17
18 Done
19 bind lsn pool LSN-DSLITE-POOL-10  203.0.113.41-203.0.113.42
20
21 Done
```



```
22 add lsn ip6profile LSN-DSLITE-PROFILE-10 -type DS-Lite -network6 2001:
      DB8::5:6
23
24 Done
25 add lsn group LSN-DSLITE-GROUP-10 -clientname LSN-DSLITE-CLIENT-10 -
      nattype DETERMINISTIC -portblocksize 20480 -ip6profile LSN-DSLITE-
      PROFILE-10
26
27 Done
28 bind lsn group LSN-DSLITE-GROUP-10 -poolname LSN-DSLITE-POOL-10
29
30 Done
31 <!--NeedCopy-->
```

Configuring Application Layer Gateways for DS-Lite

September 14, 2021

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

The Citrix ADC appliance supports ALG for the following protocols for DS-Lite:

- FTP
- ICMP
- TFTP
- SIP
- RTSP

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

September 14, 2021

You can enable or disable ALG for the FTP protocol for a DS-Lite configuration by enabling or disabling the FTP ALG option of the LSN group of the configuration.

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

ALG for the TFTP protocol is disabled by default. TFTP ALG is enabled automatically for a DS-Lite configuration when you bind a UDP LSN application profile, with endpoint-independent-mapping,

endpoint-independent filtering, and destination port as 69 (well-known port for TFTP), to the LSN group.

Application Layer Gateway for SIP Protocol

September 14, 2021

Using DS-Lite with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. SIP ALG for DS-Lite is compliant with RFC 3261, RFC 3581, RFC 4566, and RFC 4475.

> **Note**
>
> SIP ALG is supported in a Citrix ADC standalone appliance, in a Citrix ADC high availability setup, as well as in a Citrix ADC cluster setup.

Limitations of SIP ALG

SIP ALG for DS-Lite has the following limitations:

- Only SDP payload is supported.
- The following are not supported:
  - Multicast IP addresses
  - Encrypted SDP
  - SIP TLS
  - FQDN translation
  - SIP layer authentication
  - Admin partitions
  - Multipart body
  - Line folding

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuring DS-Lite. While configuring LSN, make sure that you:

- Set the following parameters while adding an LSN application profile:
  - IP Pooling = PAIRED

  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT

- Create a SIP ALG profile and make sure that you define either the source port range or destination port range. Bind the SIP ALG profile to the LSN group
- Enable SIP ALG in the LSN group

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string>[-sipalg ( ENABLED |
    DISABLED )]

show lsn group<groupname>
<!--NeedCopy-->
```

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn sipalgprofile<sipalgprofilename>[-dataSessionIdleTimeout<
    positive_integer>][-sipSessionTimeout<positive_integer>][-
    registrationTimeout<positive_integer>][-sipsrcportrange<port[-port
    ]>][-sipdstportrange<port[-port]>][-openRegisterPinhole ( ENABLED |
    DISABLED )][-openContactPinhole ( ENABLED | DISABLED )][-
    openViaPinhole ( ENABLED | DISABLED )][-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP )[-
    openRoutePinhole ( ENABLED | DISABLED )][-rport ( ENABLED | DISABLED
     )]

show lsn sipalgprofile<sipalgprofilename>
<!--NeedCopy-->
```

Sample Configuration

The following sample DS-Lite configuration, SIP ALG is enabled for TCP traffic from B4 devices in the network 2001:DB8::3:0/96.

```
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/96
Done
```



```
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-1 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn sipalgprofile SIPALGPROFILE-1  -sipdstportrange 5060 -sipTransportProtocol TCP
Done
add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1 -sipalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -appsprofilename LSN-DSLITE-APPS-PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -sipalgprofilename SIPALGPROFILE-1
Done
<!--NeedCopy-->
```

Application Layer Gateway for RTSP Protocol

September 14, 2021

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. Citrix ADC functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to Citrix ADC owned public IP address called LSN IP address.
- Inbound response—LSN IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

> **Note**
>
> RTSP ALG is supported in a Citrix ADC standalone appliance, in a Citrix ADC high availability setup, as well as in a Citrix ADC cluster setup.

Limitations of RTSP ALG

The RTSP ALG does not support the following:

- Multicast RTSP sessions
- RTSP session over UDP
- Admin partitions
- RTSP Authentication
- HTTP tunneling

Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuring DS-Lite. While configuring LSN, make sure that you:

- Set the following parameters while adding an LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Enable RTSP ALG in the LSN group
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```


To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

Sample RTSP ALG Configuration

The following sample DS-Lite configuration, RTSP ALG is enabled for TCP traffic from B4 devices in the network 2001:DB8::4:0/96.

Sample RTSP ALG Configuration:

```
add lsn client LSN-DSLITE-CLIENT-5
Done
bind lsn client LSN-DSLITE-CLIENT-5 -network6 2001:DB8::4:0/96
Done
add lsn pool LSN-DSLITE-POOL-5
Done
bind lsn pool LSN-DSLITE-POOL-5 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-5 -type DS-Lite -network6 2001:
    DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-5 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-5 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-DSLITE-GROUP-5 -clientname LSN-DSLITE-CLIENT-5 -
    portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-5 -rtspalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-5 -poolname LSN-DSLITE-POOL-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -appsprofilename LSN-DSLITE-APPS-
    PROFILE-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -rtspalgprofilename RTSPALGPROFILE-5
Done
```



```
<!--NeedCopy-->
```

Logging and Monitoring DS-Lite

September 14, 2021

You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The Citrix ADC appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a Citrix ADC appliance whose NSIP address is 10.102.37.115.You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The Citrix ADC appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a Citrix ADC appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry
DS-Lite session creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite session deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN mapping creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN mapping deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

Displaying Current DS-Lite Sessions

You can display the current DS-Lite sessions for detecting any unwanted or inefficient sessions on the Citrix ADC appliance. You can display all or some DS-Lite sessions, on the basis of selection parameters.

To display all DS-Lite sessions by using the command line interface

At the command prompt, type:

```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

To display selected DS-Lite sessions by using the command line interface

At the command prompt, type:

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

The following sample ouput displays all DS-Lite sessions existing on a Citrix ADC appliance:

> show lsn session –nattype DS-Lite

```
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1. 2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
    3002 TCP OUT

2. 2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
    52862 TCP OUT

3. 2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4. 2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT
Done
<!--NeedCopy-->
```

Configuration Using the Configuration Utility

To display all or selected DS-Lite sessions by using the configuration utility

1. **Navigate to System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. For displaying DS-Lite sessions on the basis of selection parameters, click **Search**.

Clearing DS-Lite Sessions

You can remove any unwanted or inefficient DS-Lite sessions from the Citrix ADC appliance. The appliance immediately releases the resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected DS-Lite sessions from the Citrix ADC appliance.

To clear all DS-Lite sessions by using the command line interface

At the command prompt, type:

```
flush lsn session -nattype DS-Lite

show lsn session -nattype DS-Lite
<!--NeedCopy-->
```

To clear selected DS-Lite sessions by using the command line interface

At the command prompt, type:

```
flush lsn session -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session -nattype DS-Lite
<!--NeedCopy-->
```

To clear all or selected DS-Lite sessions by using the configuration utility

1. Navigate to **System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. Click **Flush Sessions**.

Logging HTTP Header Information

The Citrix ADC appliance can log request header information of an HTTP connection that is using the DS-Lite functionality. The following header information of an HTTP request packet can be logged:

- URL that the HTTP request is destined to
- HTTP Method specified in the HTTP request
- HTTP version used in the HTTP request
- IPv4 address of the subscriber that sent the HTTP request

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set of subscribers. For example, an ISP can use this feature to find out the most popular website among a set of subscribers.

Configuration Steps

Perform the following tasks for configuring the Citrix ADC appliance to log HTTP header information:

- **Create an HTTP header log profile**. An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging.
- **Bind the HTTP header to an LSN group of a DS-Lite LSN configuration**. Bind the HTTP header log profile to an LSN group of an LSN configuration by setting the HTTP header log profile name parameter to the name of the created HTTP header log profile. The Citrix ADC appliance then logs HTTP header information of any HTTP requests related to the LSN group. An HTTP header log profile can be bound to multiple LSN groups, but an LSN group can have only one HTTP header log profile.

To create an HTTP header log profile by using the command line interface

At the command prompt, type:

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

To bind an HTTP header log profile to an LSN group by using the command line interface

At the command prompt, type:

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

Sample Configuration

In the following DS-Lite LSN configuration, HTTP header log profile HTTP-Header-LOG-1 is bound to LSN group LSN-DSLITE-GROUP-1. The log profile has all the HTTP attributes (URL, HTTP method,

HTTP version, and HOST IP address) enabled for logging, so that all these attributes are logged for any HTTP requests from B4 devices (in the network 2001:DB8:5001::/96).

Sample Configuration:

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1

Done

add lsn client LSN-DSLITE-CLIENT-1

Done

bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100

Done

add lsn pool LSN-DSLITE-POOL-1

Done

bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70

Done

add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
   DB8::5:6

Done

add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
   portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -httphdrlogprofilename HTTP-HEADER-
   LOG-1

Done
<!--NeedCopy-->
```


IPFIX Logging

The Citrix ADC appliance supports sending information about LSN events in Internet Protocol Flow Information Export (IPFIX) format to the configured set of IPFIX collector(s). The appliance uses the existing AppFlow feature to send LSN events in IPFIX format to the IPFIX collectors.

IPFIX based logging is available for the following DS_Lite related events:

- Creation or deletion of an LSN session.
- Creation or deletion of an LSN mapping entry.
- Allocation or de-allocation of port blocks in the context of deterministic NAT.
- Allocation or de-allocation of port blocks in the context of dynamic NAT.
- Whenever subscriber session quota is exceeded.

Points to Consider before you Configure IPFIX logging

Before you start configuring IPSec ALG, consider the following points:

- You must configure the AppFlow feature and IPFIX collector(s) on the Citrix ADC appliance. For instructions, see Configuring the AppFlow feature.

Configuration Steps

Perform the following tasks for logging LSN information in IPFIX format:

- **Enable LSN logging in the AppFlow configuration**. Enable the LSN logging parameter as part of AppFlow configuration.
- **Create an LSN log profile**. An LSN log profile includes the IPFIX parameter that enables or disables the log information in IPFIX format.
- **Bind the LSN log profile to an LSN group of an LSN configuration**. Bind the LSN log profile to one or multiple LSN group(s). Events related to the bound LSN group will be logged in IPFIX format.

To enable LSN logging in the AppFlow configuration by using the CLI

At the command prompt, type:

```
set appflow param -lsnLogging (ENABLED |DISABLED )

show appflow param
<!--NeedCopy-->
```


To create an LSN log profile by using the CLIAt the command prompt, type

At the command prompt, type:

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind the LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

To create an LSN log profile by using the GUI

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **Log** tab, and then add a log profile.

To bind the LSN log profile to an LSN group of an LSN configuration by using the GUI

1. Navigate to **System** > **Large Scale NAT** > **LSN Group,** open the **LSN** group.
2. In **Advanced Settings,** click **+ Log Profile** to bind the created Log profile to the LSN group.

Port Control Protocol for DS-Lite

September 14, 2021

Citrix ADC appliances now support Port Control Protocol (PCP) for large scale NAT (LSN). Many of an ISP's subscriber applications must be accessible from Internet (for example, Internet of Things (IOT) devices, such as an IP camera that provides surveillance over the Internet). One way to meet this requirement is to create static large scale NAT (LSN) maps. But for a very large number of subscribers, creating static LSN NAT maps is not a feasible solution.

Port Control Protocol (PCP) enables a subscriber to request specific LSN NAT mappings for itself and/or for other 3rd party devices. The large scale NAT device creates an LSN map and sends it to the subscriber. The subscriber sends the remote devices on the Internet the NAT IP address:NAT port at which they can connect to the subscriber.

Applications usually send frequent keep-alive messages to the large scale NAT device so that their LSN mappings do not time out. PCP helps reduce the frequency of such keep-alive messages by enabling the applications to learn the timeout settings of the LSN mappings. This helps reduce bandwidth consumption on the ISP's access network and battery consumption on mobile devices.

PCP is a client-server model and runs over the UDP transport protocol. A Citrix ADC appliance implements the PCP server component and is compliant with RFC 6887.

Configuration Steps

Perform the following tasks for configuring PCP:

- (Optional) Create a PCP profile. A PCP profile includes settings for PCP related parameters (for example, to listen for mapping and peer PCP requests). A PCP profile can be bound to a PCP server. A PCP profile bound to a PCP server applies all its settings to the PCP server. A PCP profile can be bound to multiple PCP servers. By default, one PCP profile with default parameters settings is bound to all PCP servers. A PCP profile that you bind to a PCP server overrides the default PCP profile settings for that server. A default PCP profile has the following parameter settings:
  - Mapping: Enabled
  - Peer: Enabled
  - Minimum map life: 120 seconds
  - Maximum max life: 86400 seconds
  - Announce count: 10
  - Third Party: Disabled
- Create a PCP server and bind a PCP profile to it. Create a PCP server on the Citrix ADC appliance to listen for PCP related requests and messages from the subscribers. A Subnet IP (SNIP) address must be assigned to a PCP server to access it. By default, a PCP server listens on port 5351.
- Bind the PCP server to an LSN group of an LSN configuration. Bind the created PCP server to an LSN group of an LSN configuration by setting the PCP Server parameter to specify the created PCP server. The created PCP server can be accessed only by the subscribers of this LSN group. Note: A PCP server for a large scale NAT configuration does not serve requests from subscribers that are identified from ACL rules.

To create a PCP profile by using the CLI

At the command prompt, type:

```
1 add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]
```



```
2
3 show pcp profile <name>
4 <!--NeedCopy-->
```

To create a PCP server by using the CLI

At the command prompt, type:

```
1 add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
     string>]
2
3 show pcp server <name>
4 <!--NeedCopy-->
```

Sample Configuration for DS-LITE

In the following sample configuration, PCP server PCP-SERVER-1, with PCP settings from PCP-DSLITE-PROFILE-1, is bound to LSN group LSN-DSLITE-GROUP-1. PCP-SERVER-9 serves PCP requests from IPv4 subscribers behind B4 devices from network 2001:DB8::3:0/100.

Sample configuration:

```
1 add pcp profile PCP-DSLITE-PROFILE-1 -minMapLife 300
2 Done
3 add pcp server PCP-DSLITE-SERVER-1 192.0.3.10  -pcpProfile PCP-DSLITE-
     PROFILE-1
4 Done
5 add lsn client LSN-DSLITE-CLIENT-1
6 Done
7 bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
8 Done
9 add lsn pool LSN-DSLITE-POOL-1
10 Done
11 bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
12 Done
13 add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
     DB8::5:6
14 Done
15 add lsn group LSN-DSLITE-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
     ip6profile LSN-NAT64-PROFILE-1
16 Done
17 bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-NAT64-POOL-1
18 Done
```



```
bind lsn group LSN-DSLITE-GROUP-1 -poolname  PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

Large Scale NAT64

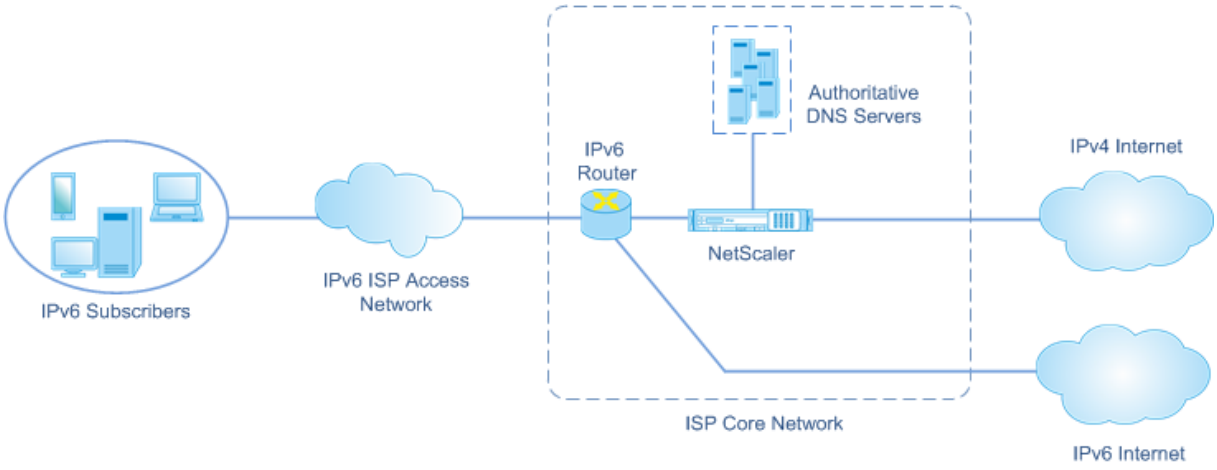
September 14, 2021

Because of the imminent exhaustion of IPv4 addresses, ISPs have started transitioning to IPv6 infrastructure. But during the transition, ISPs must continue to support IPv4 along with IPv6, because most of the public Internet still uses IPv4. Large scale NAT64 is an IPv6 transition solution for ISPs with IPv6 infrastructure to connect their IPv6-only subscribers to the IPv4 Internet. DNS64 is a solution for enabling discovery of IPv4-only domains by IPv6-only clients. DNS64 is used with large scale NAT64 to enable seamless communication between IPv6-only clients and IPv4-only servers.

A Citrix ADC appliance implements large scale NAT64 and DNS64 and is compliant with RFCs 6145, 6146, 6147, 6052, 3022, 2373, 2765, and 2464.

Architecture

The NAT64 architecture of an ISP using a Citrix ADC appliance consists of IPv6 subscribers accessing the IPv4 Internet through a Citrix ADC appliance deployed in the ISP's core network. IPv6 subscribers are connected to the ISP core network through the ISP's IPv6-only access network.



The large scale NAT64 functionality of a Citrix ADC appliance enables communication between IPv6 clients and IPv4 servers through IPv6-to-IPv4 packet translation, and vice versa, while maintaining session information on the Citrix ADC appliance.Citrix ADC DNS64 functionality represents IPv4-only domains to IPv6-subscribers by synthesizing DNS AAAA records for IPv4-only domains and sending them to the subscribers.

Large scale NAT64 has two main components: NAT64 prefix and NAT IPv4 pool. DNS64 has one main component, DNS64 prefix, which has the same value as NAT64 prefix.

Upon receiving an AAAA request from an IPv6-only subscriber for a domain name that is hosted on an IPv4-only web server on the Internet, the Citrix ADC DNS64 functionality synthesizes an AAAA record for the domain name and sends it to the subscriber. The AAAA record is synthesized by concatenating the DNS64 prefix (which is set to the NAT64 prefix) and the actual IPv4 address of the domain name.

The subscriber now has an IPv6 destination address that corresponds to the desired domain name. The subscriber sends the request to the synthesized IPv6 address. Upon receiving the IPv6 request, the large scale Citrix ADC NAT64 functionality translates the IPv6 request packet to an IPv4 request packet. Large scale NAT64 sets the IPv4 request's destination address to the IPv4 address, which is extracted from the IPv6 request's destination address by stripping the NAT64 prefix from the IPv6 address. The destination port is retained from the IPv6 request. Large Scale NAT64 also sets the source IP address:source port of the IPv4 packet to the NAT IP address:NAT port selected from the configured NAT pool.

The appliance maintains a record of all active sessions that use the large scale NAT64 functionality. These sessions are called large scale NAT64 sessions. The appliance also maintains the mappings between subscriber IPv6 address and port, and NAT IPv4 address and port, for each large scale NAT64 session. These mappings are called large scale NAT64 mappings. From large scale NAT64 session entries and large scale NAT64 mapping entries, the Citrix ADC appliance recognizes a response packet (received from the Internet) as belonging to a particular NAT64 session.

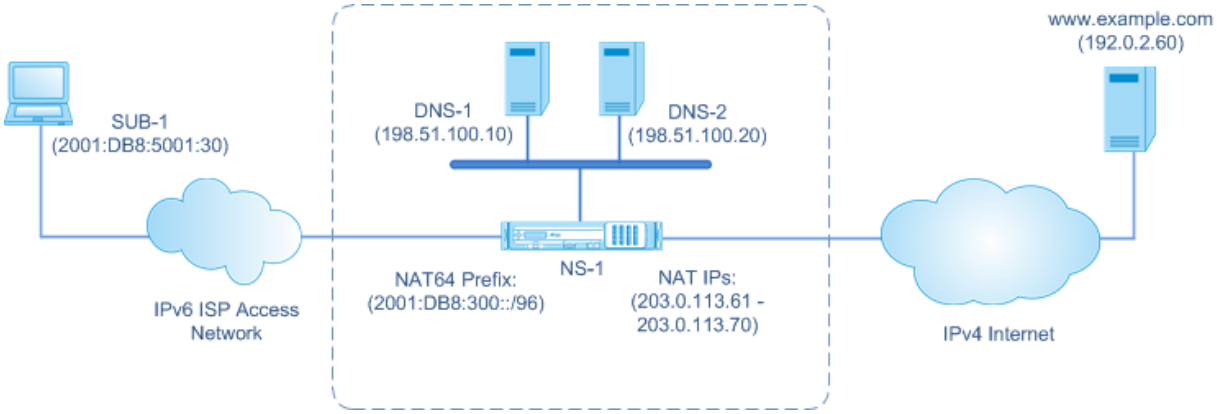
When the appliance receives an IPv4 response packet belonging to a particular NAT64 session, it uses the information stored in the NAT64 session to translate the IPv4 packet into an IPv6 packet, and then sends the IPv6 response packet to the subscriber.

Example: Traffic Flow of NAT64 and DNS64 Deployment

Consider an example of a large scale NAT64 and DNS64 deployment consisting of Citrix ADC appliance NS-1 and two local DNS servers, DNS-1 and DNS-2, in an ISP's core network, and IPv6 subscriber SUB-1. SUB-1 is connected to NS-1 through the ISP's IPv6 access network. NS-1 includes large scale NAT64 and DNS64 configurations for enabling the communication between IPv6 subscriber SUB-1 and IPv4 hosts (internal and external).

Large scale NAT64 configuration includes a NAT64 prefix (2001:DB8:300::/96) and NAT IPv4 pool for translation of IPv6 requests to IPv4 requests and IPv4 responses to IPv6 responses.

DNS64 configuration includes a DNS load balancing virtual server LBVS-DNS64-1 (2001:DB8:9999::99) and a DNS64 prefix (2001:DB8:300::/96). LBVS-DNS64-1 represents local DNS server DNS-1 and DNS-2 to ISP's subscribers. The DNS64 prefix, which has the same value as the NAT64 prefix, is used for synthesizing DNS AAAA records from DNS A records received from DNS servers DNS-1 and DNS-2. NS-1 responds with a synthesized AAAA record to SUB-1 for a DNS request to resolve an IPv4 host.



DNS64 Traffic Flow

Traffic flows between IPv6 subscriber SUB-1 and site `www.example.com`, which resides on an IPv4-only web server on the Internet, as follows:

1. IPv6 subscriber SUB-1 sends a DNS AAAA request for `www.example.com` to its designated DNS server (2001:DB8:9999::99).
2. DNS load balancing virtual server LBVS-DNS64-1 (2001:DB8:9999::99) on Citrix ADC appliance NS1 receives the AAAA request. LBVS-DNS64-1's load balancing algorithm selects DNS server DNS-1 and forwards the AAAA request to it.
3. DNS-1 returns an empty record or an error message, because there is no AAAA record available for `www.example.com`.
4. Because the DNS64 option is enabled on LBVS-DNS64-1 and the AAAA request from CL1 matches the condition specified in DNS64-Policy-1, NS1 sends a DNS A request to DNS-1 for the IPv4 address of `www.example.com`.
5. DNS-1 responds with the A record of 192.0.2.60 for `www.example.com`.
6. DNS64 module on NS1 synthesizes an AAAA record for `www.example.com` by concatenating the DNS64 Prefix (2001:DB8:300::/96) associated with LBVS-DNS64-1, and IPv4 address (192.0.2.60) for `www.example.com` = 2001:DB8:300::192.0.2.60
7. NS1 sends the synthesized AAAA record to IPv6 client CL1. NS1 also caches the A record into its memory. NS1 uses the cached A record to synthesize AAAA records for subsequent AAAA requests.

NAT64 Traffic Flow

1. IPv6 subscriber SUB-1 sends a request to 2001:DB8:5001:30 `www.example.com`. The IPv6 packet has:
   - Source IP address = 2001:DB8:5001:30
   - Source port = 2552

- Destination IP address = 2001:DB8:300::192.0.2.60
- Destination port = 80

2. IPv6 subscriber SUB-1 sends a request to 2001:DB8:5001:30 www.example.com. The IPv6 packet has:
   - Source IP address = 2001:DB8:5001:30
   - Source port = 2552
   - Destination IP address = 2001:DB8:300::192.0.2.60
   - Destination port = 80
3. When NS-1 receives the IPv6 packet, the large scale NAT64 module creates a translated IPv4 request packet with:
   - Source IP address = One of the IPv4 addresses available in the configured NAT pool (203.0.113.61)
   - Source port = One of ports available with the allocated NAT IPv4 address (3002)
   - Destination IP address = IPv4 address extracted from the IPv6 request's destination address by stripping the NAT64 prefix (2001:DB8:300::/96) from the IPv6 address (192.0.2.60)
   - Destination port = IPv6 request's destination port (80)
4. The large scale NAT64 module also creates mapping and session entries for this large scale NAT64 flow. The session and mapping entries include the following information:
   - Source IP address of the IPv6 packet = 2001:DB8:5001:30
   - Source port of the IPv6 packet = 2552
   - NAT IP address = 203.0.113.61
   - NAT port = 3002
   - NS-1 sends the resulting IPv4 packet to its destination on the Internet.
5. Upon receiving the request packet, the server for www.example.com processes the packet and sends a response packet to NS-1. The IPv4 response packet has:
   - Source IP address = 192.0.2.60
   - Source port = 80
   - Destination IP address = 203.0.113.61
   - Destination port = 3002
6. Upon receiving the IPv4 response packet, NS-1 examines the large scale NAT64 mapping and session entries and finds that the IPv4 response packet belongs to a large scale NAT64 session. The large scale NAT64 module creates a translated IPv6 response packet:
   - Source IP address = 2001:DB8:300::192.0.2.60
   - Source port = 80
   - Destination IP address = 2001:DB8:5001:30
   - Destination port = 2552

7. NS-1 sends the translated IPv6 response to client SUB-1.

Large Scale NAT64 features Supported on Citrix ADC appliances

Large scale NAT64 on a Citrix ADC appliance supports the standard LSN feature set. For more information on these LSN features, see LSN Architecture.

Following are some of the large scale NAT64 features supported on Citrix ADC appliances:

- ALGs. Support of application Layer Gateway (ALG) for SIP, RTSP, FTP, ICMP, and TFTP protocols.
- Deterministic/Fixed NAT. Support for pre-allocation of blocks of ports to subscribers to minimize logging.
- Mapping. Support of Endpoint-independent mapping (EIM), Address-dependent mapping (ADM), and Address-Port dependent mapping (APDM).
- Filtering. Support of Endpoint-Independent Filtering (EIF), Address-Dependent Filtering (ADF), and Address-Port-Dependent Filtering (APDF).
- Quotas. Configurable limits on number of ports, sessions per subscriber, and sessions per LSN group.
- Static Mapping. Support for manually defining a large scale NAT64 mapping.
- Hairpin Flow. Support for communication between subscribers or internal hosts using NAT IP addresses.
- 464XLAT connections. Support for communication between IPv4-only applications on IPv6 subscriber hosts and IPv4 hosts on the Internet through IPv6 network.
- Variable length NAT64 and DNS64 prefixes. The Citrix ADC appliance supports defining NAT64 and DNS64 prefixes of lengths of 32, 40, 48, 56, 64, and 96.
- Multiple NAT64 and DNS64 prefix. The Citrix ADC appliance supports multiple NAT64 and DNS64 prefixes.
- LSN Clients. Support for specifying or identifying subscribers for large scale NAT64 by using IPv6 prefixes and extended ACL6 rules.
- Logging. Support for logging NAT64 sessions for law enforcement. In addition, the following are also supported for logging.
  - **Reliable SYSLOG**. Support for sending SYSLOG messages over TCP to external log servers for a more reliable transport mechanism.
  - **Load balancing of log servers**. Support for load balancing of external log servers for preventing storage of redundant log messages.
  - **Minimal Logging**. Deterministic LSN configurations or Dynamic LSN configurations with port block significantly reduce the large scale NAT64 log volume.
  - **Logging MSISDN information**. Support for including subscribers' MSISDN information in large scale NAT64 logs to identify and track subscriber activity over the Internet.

Points to Consider for Configuring Large Scale NAT64

September 14, 2021

Before you start configuring large scale NAT64 and DNS64, consider these points:

1. Make sure you understand the different components of large scale NAT64, described in RFCs.
2. The Citrix ADC appliance supports only the following ALGs for large Scale NAT64:
   - FTP
   - TFTP
   - ICMP
   - SIP
   - RTSP
3. In a high availability setup of two Citrix ADC appliances, large NAT64 session synchronization (connection mirroring) is not supported.

Configuring DNS64

September 14, 2021

Creating the required entities for stateful NAT64 configuration on the Citrix ADC appliance involves the following procedures:

- Add DNS services. DNS services are logical representations of DNS servers for which the Citrix ADC appliance acts as a DNS proxy server. For more information on setting optional parameters of a service, see Load Balancing.
- Add DNS64 action and DNS64 policy and then bind the DNS64 action to the DNS64 policy. A DNS64 policy specifies conditions to be matched against traffic for DNS64 processing according to the settings in the associated DNS64 action. The DNS64 action specifies the mandatory DNS64 prefix and the optional exclude-rule and mapped-rule settings.
- Create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it. The DNS load balancing virtual server acts as a DNS proxy server for DNS servers represented by the bound DNS services. Traffic arriving at the virtual server is matched against the bound DNS64 policy for DNS64 processing. For more information on setting optional parameters of a load balancing virtual server, see Load Balancing.

> **Note**
>
> The command line interface has separate commands for these two tasks, but the GUI combines them in a single dialog box.

- Enable caching of DNS records. Enable the global parameter for the Citrix ADC appliance to cache DNS records, which are obtained through DNS proxy operations. For more information on enabling caching of DNS records, see Enabling Caching of DNS Records.

To create a service of type DNS by using the command line interface

At the command prompt, type:

```
add service <name> <IP> <serviceType> <port> …
<!--NeedCopy-->
```

To create a DNS64 action by using the command line interface

At the command prompt, type:

```
add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <
    expression>] [-excludeRule <expression>]
<!--NeedCopy-->
```

To create a DNS64 policy by using the command line interface

At the command prompt, type:

```
add dns policy64 <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

To create a DNS load balancing virtual server by using the command line interface

At the command prompt, type:

```
add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED
    ) [-bypassAAAA ( YES | NO)] …
<!--NeedCopy-->
```

To bind the DNS services and the DNS64 policy to the DNS load balancing virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName> ...

```



```
bind lb vserver <name> -policyName <string> -priority <positive_integer
    > ...
<!--NeedCopy-->
```

Sample configuration:

```
add service SVC-DNS-1 203.0.113.50 DNS 53
Done
add service SVC-DNS-2 203.0.113.60 DNS 53
Done
add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
Done
add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:
    DB8:5001::/64)" -action DNS64-Action-1
Done
add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-1
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-2
Done
bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
Done
<!--NeedCopy-->
```

Configuring Large Scaler NAT64

September 14, 2021

A large scale NAT64 configuration on a Citrix ADC appliance uses the LSN commands sets. In a large scale NAT64 configuration, the LSN client entity specifies the IPv6 address or IPv6 network address, or ACL6 rules, for identifying IPv6 subscribers. A NAT64 configuration also includes an IPv6 profile, which specifies a NAT64 prefix.

Configuring NAT64 on a Citrix ADC appliance consists of the following tasks:

- Set the global LSN parameters. Global parameters include the amount of Citrix ADC memory reserved for the LSN feature and synchronization of LSN sessions in a high availability setup.
- Create an LSN client entity for identifying traffic from IPv6 subscribers. The LSN client entity refers to a set of IPv6 subscribers. The client entity includes IPv6 addresses or IPv6 network prefixes, or ACL6 rules, for identifying the traffic from these subscribers. An LSN client can be bound to only one LSN group. The command line interface has two commands for creating an

LSN client entity and binding a subscriber to the LSN client entity. The GUI combines these two operations on a single screen.

- Create an LSN pool and bind NAT IP addresses to it. An LSN pool defines a pool of NAT IP addresses to be used by the Citrix ADC appliance to perform large scale NAT64. The command line interface has two commands for creating an LSN pool and binding NAT IP addresses to the LSN pool. The GUI combines these two operations on a single screen.
- Create an LSN IP6 profile. An LSN IP6 profile defines the NAT64 prefix for a large scale NAT64 configuration.
- (Optional) Create an LSN Transport Profile for a specified protocol. An LSN transport profile defines various timeouts and limits, such as maximum large scale NAT64 sessions and maximum ports usage that a subscriber can have for a given protocol. You bind an LSN transport profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. A profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called the default transport profile. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
- (Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it. An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination ports, you bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The GUI combines these two operations on a single screen.
- Create an LSN Group and bind LSN pools, LSN IPv6 profile, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group. An LSN group is an entity consisting of an LSN client, an LSN IPv6 profile, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only one LSN IPv6 profile can be bound to an LSN group, and an LSN IPv6 profile bound to an LSN group cannot be bound to other LSN groups. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN

group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN group and binding LSN pools, LSN transport profiles, and LSN application profiles to the LSN group. The GUI combines these two operations in a single screen.

Configuration Using the Command Line

You can create different configurations using the command line interface. Follow the steps given below.

To create an LSN client by using the command line interface

At the command prompt, type:

```
add lsn client <clientname
<!--NeedCopy-->
```

To bind an IPv6 network or an ACL6 rule to an LSN client by using the command line interface

At the command prompt, type:

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
   string>)

show lsn client
<!--NeedCopy-->
```

To create an LAN pool by using the command line interface

At the command prompt, type:

```
add lsn pool <poolname>

show lsn pool <poolname>
<!--NeedCopy-->
```

To bind NAT IP addresses to an LSN pool by using the command line interface

At the command prompt, type:


```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

> **Note**
>
> For removing NAT IP (LSN IP addresses) addresses from an LSN pool, use the unbind lsn pool command.

To configure an LSN IPv6 profile by using the command line interface

At the command prompt, type:

```
add lsn ip6profile <name>  -type NAT64 -natprefix <ipv6_addr|*>

show lsn ip6profile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn transportprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn appsprofile
<!--NeedCopy-->
```


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

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

To create an LSN group by using the command line interface

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging(
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync ( ENABLED | DISABLED )] [-snmptraplimit<positive_integer
    >] [-ftp ( ENABLED | DISABLED )] [-sipalg ( ENABLED | DISABLED )] [-
    rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

Sample Large Scale NAT64 Configurations

Here are some sample configurations of large scale NAT64:

Simple large scale NAT64 configuration with default settings:

```
add lsn client LSN-NAT64-CLIENT-1
```



```
2
3 bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
4
5 add lsn pool LSN-NAT64-POOL-1
6
7 bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
8
9 add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
10
11 add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
12
13 bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
14
15 <!--NeedCopy-->
```

Simple large scale NAT64 configuration with an extended ACL6 rule for identifying subscribers:

```
1 add ns acl6 LSN-NAT64-ACL-2 ALLOW  - srcIPv6 = 2001:DB8:5002::20 - 2001:
    DB8:5002::200
2
3 apply acl6s
4
5 add lsn client LSN-NAT64-CLIENT-2
6
7 bind lsn client LSN-NAT64-CLIENT-2  - acl6name LSN-NAT64-ACL-2
8
9 add lsn pool LSN-NAT64-POOL-2
10
11 bind lsn pool LSN-NAT64-POOL-2 203.0.113.5-203.0.113.10
12
13 add lsn ip6profile LSN-NAT64-PROFILE-2 -type NAT64 -natprefix 2001:DB8
    :302::/96
14
15 add lsn group LSN-NAT64-GROUP-2 -clientname LSN-NAT64-CLIENT-2 -
    ip6profile LSN-NAT64-PROFILE-2
16
17 bind lsn group LSN-NAT64-GROUP-2 -poolname LSN-NAT64-POOL-2
18
19 <!--NeedCopy-->
```

Large scale NAT64 configuration with deterministic NAT resource allocation:

```
1 add lsn client LSN-NAT64-CLIENT-7
```



```
2
3 bind lsn client LSN-NAT64-CLIENT-7 -network6 2001:DB8:1002::7/128
4
5 add lsn pool LSN-NAT64-POOL-7 -nattype DETERMINISTIC
6
7 bind lsn pool LSN-NAT64-POOL-7 203.0.113.24-203.0.113.27
8
9 add lsn ip6profile LSN-NAT64-PROFILE-7 -type NAT64 -natprefix 2001:DB8
     :307::/96
10
11 add lsn group LSN-NAT64-GROUP-7 -clientname LSN-NAT64-CLIENT-7 -
     ip6profile LSN-NAT64-PROFILE-7 -nattype DETERMINISTIC -portblocksize
      256
12
13 bind lsn group LSN-NAT64-GROUP-7 -poolname LSN-POOL-7
14
15 <!--NeedCopy-->
```

Configuring Application Layer Gateways for Large Scale NAT64

September 14, 2021

For some Application layer protocols, the IP addresses and protocol port numbers are also communicated in the packet payload. Application Layer Gateway for a protocol parses the packet's payload and does necessary changes to ensure that the protocol continues to work over large scale NAT64.

The Citrix ADC appliance supports ALG for the following protocols for large scale NAT64:

- FTP
- ICMP
- TFTP
- SIP
- RTSP

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

September 14, 2021

You can enable or disable ALG for the FTP protocol for an large scale NAT64 configuration by enabling or disabling the FTP ALG option of the LSN group of the configuration.

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

ALG for the TFTP protocol is disabled by default. TFTP ALG is enabled automatically for an large scale NAT64 configuration when you bind a UDP LSN application profile, with endpoint-independent-mapping, endpoint-independent filtering, and destination port as 69 (well-known port for TFTP), to the LSN group.

Application Layer Gateway for SIP Protocol

September 14, 2021

Using Large Scale NAT64 with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. SIP ALG for large scale NAT64 is compliant with RFC 3261, RFC 3581, RFC 4566, and RFC 4475.

> **Note**
>
> SIP ALG is supported in a Citrix ADC standalone appliance, in a Citrix ADC high availability setup, as well as in a Citrix ADC cluster setup.

Limitations of SIP ALG

SIP ALG for large scale NAT64 has the following limitations:

- Only SDP payload is supported.
- The following are not supported:
  - Multicast IP addresses
  - Encrypted SDP
  - SIP TLS
  - FQDN translation
  - SIP layer authentication
  - Traffic Domains
  - Admin partitions
  - Multipart body
  - Line folding

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Large Scale NAT64. While configuring LSN, make sure that you:

- Set the following parameters while adding an LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Create a SIP ALG profile and make sure that you define either the source port range or destination port range. Bind the SIP ALG profile to the LSN group.
- Enable SIP ALG in the LSN group.

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-sipalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn sipalgprofile <sipalgprofilename>[-dataSessionIdleTimeout <
    positive_integer>][-sipSessionTimeout <positive_integer>] [-
    registrationTimeout <positive_integer>] [-sipsrcportrange <port[-
    port]>] [-sipdstportrange <port[-port]>] [-openRegisterPinhole (
    ENABLED | DISABLED )] [-openContactPinhole ( ENABLED | DISABLED )]
    [-openViaPinhole ( ENABLED | DISABLED )] [-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP ) [-
    openRoutePinhole ( ENABLED | DISABLED )] [-rport ( ENABLED |
    DISABLED )]

show lsn sipalgprofile <sipalgprofilename
<!--NeedCopy-->
```


Sample Configuration

The following sample large scale NAT64 configuration, SIP ALG is enabled for TCP traffic from subscriber devices in the network 2001:DB8:1003::/96.

```
add lsn client LSN-NAT64-CLIENT-9

Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96

Done
add lsn pool LSN-NAT64-POOL-9

Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90

Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
   :309::/96

Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
   mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done
add lsn sipalgprofile SIPALGPROFILE-9  -sipdstportrange 5060  -
   sipTransportProtocol TCP

Done
add lsn group LSN-NAT64-GROUP-9 -clientnameLSN-NAT64-CLIENT-9 -
   ip6profile LSN-NAT64-PROFILE-7 -sipalg ENABLED

Done
bind lsn group LSN-NAT64-GROUP-9 -poolnameLSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
   PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9 -sipalgprofilename SIPALGPROFILE-9
Done
<!--NeedCopy-->
```


Application Layer Gateway for RTSP Protocol

September 14, 2021

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. Citrix ADC functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to Citrix ADC owned public IP address called LSN IP address.
- Inbound response—LSN IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

> **Note**
>
> RTSP ALG is supported in a Citrix ADC standalone appliance, in a Citrix ADC high availability setup, as well as in a Citrix ADC cluster setup.

Limitations of RTSP ALG

The RTSP ALG does not support the following:

- Multicast RTSP sessions
- RTSP session over UDP
- Admin partitions
- RTSP Authentication
- HTTP tunneling

Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuring Large Scale NAT64. While configuring, make sure that you:

- Set the following parameters while adding an LSN application profile:

  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Enable RTSP ALG in the LSN group
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

Sample RTSP ALG Configuration

The following sample large scale NAT64 configuration, RTSP ALG is enabled for TCP traffic from subscriber devices in the network 2001:DB8:1002::/96.

```
add lsn client LSN-NAT64-CLIENT-9
Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96
Done
add lsn pool LSN-NAT64-POOL-9
Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90
Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
    :309::/96
```



```
Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-9 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-NAT64-GROUP-9 -clientname LSN-NAT64-CLIENT-9 -
    ip6profile LSN-NAT64-PROFILE-7 -rtspalg ENABLED
Done
bind lsn group LSN-NAT64-GROUP-9 -poolname LSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
    PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9  -rtspalgprofilename RTSPALGPROFILE-9
Done
<!--NeedCopy-->
```

Configuring Static Large Scale NAT64 Maps

September 14, 2021

The Citrix ADC appliance supports manual creation of NAT64 mappings, which contain the mapping between the following information:

- Subscriber's IP address and port
- NAT IP address and port

Static Large Scale NAT64 mappings are useful in cases where you want to ensure that the IPv4 connections initiated to a NAT IP address:port are IPv6 translated and mapped to the subscriber IP address:port (for example, web servers located in the internal network).

To create a Large Scale NAT64 mapping by using the command line

At the command prompt, type:

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

show lsn static
<!--NeedCopy-->
```


Wildcard Port Static Large Scale NAT64 Maps

A static large scale NAT64 mapping entry is usually a one-to-one mapping between a subscriber IPv6 address:port and a NAT IPv4 address:port. A one-to-one static large scale NAT64 mapping entry exposes only one port of the subscriber IP address to the Internet.

Some situations might require exposing all ports (64K - limited to the maximum number of ports of a NAT IPv4 address) of a subscriber IP address to the Internet (for example, a server hosted on an internal network and running a different service on each port). To make these internal services accessible through the Internet, you have to expose all the ports of the server to the Internet.

One way to meet this requirement is to add 64 thousand one-to-one static mapping entries, one mapping entry for each port. Creating those entries is very cumbersome and a big task. Also, this large number of configuration entries might lead to performance issues in the Citrix ADC appliance.

A simpler method is to use wildcard ports in a static mapping entry. You just need to create one static mapping entry with NAT-port and subscriber-port parameters set to the wildcard character (*), and the protocol parameter set to ALL, to expose all the ports of a subscriber IP address for all protocols to the Internet.

For a subscriber's inbound or outbound connections matching a wildcard static mapping entry, the subscriber's port does not change after the NAT operation. When a subscriber-initiated connection to the Internet matches a wildcard static mapping entry, the Citrix ADC appliance assigns a NAT port that has the same number as the subscriber port from which the connection is initiated. Similarly, an Internet host gets connected to a subscriber's port by connecting to the NAT port that has the same number as the subscriber's port.

To configure the Citrix ADC appliance to provide access to all ports of a subscriber IPv6 address, create a wildcard static map with the following mandatory parameter settings:

- Protocol=ALL
- Subscriber port = *
- NAT port = *

In a wildcard static map, unlike in a one-to-one static map, setting the NAT IP parameter is mandatory. Also, the NAT IP address assigned to a wildcard static map cannot be used for any other subscribers.

To create a wildcard static map by using the command line interface

At the command prompt, type:

```
1 add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
     positive_integer>] [-destIP <ip_addr>
2
3 show lsn static
4 <!--NeedCopy-->
```


In the following sample configuration of a wildcard static map, all ports of a subscriber whose IP address is 2001:DB8:5001::3 are made accessible through NAT IP 203.0.11.33.

```
1 add lsn static NAT64-WILDCARD-STATIC-1 ALL 2001:DB8:5001::3 *
     203.0.113.33 *
2 Done
3 <!--NeedCopy-->
```

Logging and Monitoring Large Scale NAT64

September 14, 2021

You can log large scale NAT64 information to diagnose and troubleshoot problems, and to meet legal requirements. You can monitor the performance of the large scale NAT64 deployment by using statistical counters and displaying the related current sessions.

Logging Large Scale NAT64

Logging large scale NAT64 information is required for ISPs to meet legal requirements and identify the source of traffic at any given time.

A log message for a large scale NAT64 mapping entry consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp.
- Entry type (MAPPING).
- Whether the mapping entry was created or deleted.
- Subscriber's IP address, port, and traffic domain ID.
- NAT IP address and port.
- Protocol name.
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for endpoint-independent mapping.
  - Only the destination IP address is logged for address-dependent mapping. The port is not logged.
  - Destination IP address and port are logged for address-port-dependent mapping.

A log message for a large scale NAT64 session consists of the following information:

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced

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

The following table displays sample large scale NAT64 log entries of each type stored on the configured log servers. The log entries show that a subscriber whose IPv6 address is 2001:db8:5001::9 was connected to destination IP:port 23.0.0.1:80 through NAT IP:port 203.0.113.63:45195 on April 7, 2016, from 14:07:57 GMT to 14:10:59 GMT.

Log Entry Type	Sample Log Entry
Session Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
Mapping Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
Session Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
Mapping Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

Configuration Steps

You can configure logging of large scale NAT64 information for a large scale NAT64 configuration by setting the LSN groups's logging and session logging parameters. These are group level parameters and are disabled by default. The Citrix ADC appliance logs large scale NAT64 sessions for an LSN group only when both logging and session logging parameters are enabled.

The following table displays the logging behavior for an LSN group for various settings of logging and session logging parameters.

Logging	Session Logging	Logging Behavior
Enabled	Enabled	Logs LSN mapping entries as well as LSN sessions
Enabled	Disabled	Logs LSN mapping entries but not LSN sessions
Disabled	Enabled	Logs neither mapping entries nor LSN sessions

To log large scale NAT64 information by using the CLI

To set the logging and session logging parameters while adding an LSN group, at the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-logging (ENABLED|
    DISABLED)] [-sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

To set the logging and session logging parameters for an existing LSN group, at the command prompt, type:

```
set lsn group <groupname> [-logging (ENABLED|DISABLED)] [-
    sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```


Sample Configuration

In this example of large scale NAT64 configuration, logging and session logging paramters are enabled for LSN group LSN-NAT64-GROUP-1.

The Citrix ADC appliance logs large scale NAT64 session and mapping information for connections from subscribers (in the network 2001:DB8:5001::/96).

Sample configuration:

```
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1  -logging ENABLED -sessionLogging
    ENABLED
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
<!--NeedCopy-->
```

Logging MSISDN Information for Large Scale NAT64

A Mobile Station Integrated Subscriber Directory Number (MSISDN) is a telephone number uniquely identifying a subscriber across multiple mobile networks. The MSISDN is associated with a country code and a national destination code identifying the subscriber's operator.

You can configure a Citrix ADC appliance to include MSISDNs in large scale NAT64 LSN log entries for subscribers in mobile networks. The presence of MSISDNs in the LSN logs facilitates faster and accurate back tracing of a mobile subscriber who has violated a policy or law, or whose information is required by lawful interception agencies.

The following sample LSN log entries include MSISDN information for a connection from a mobile subscriber in an LSN configuration. The log entries show that a mobile subscriber whose MSISDN is E164:5556543210 and IPv6 address is 2001:db8:5001::9 was connected to destination IP:port 23.0.0.1:80 through the NAT IP:port 203.0.113.63:45195 on April 7, 2016, from 14:07:57 GMT to 14:10:59 GMT.

Log Entry Type	Sample Log Entry
Session Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
Mapping Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
Session Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
Mapping Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

Configuration Steps

Perform the following tasks for including MSISDN information in LSN logs:

- **Create an LSN log profile**. An LSN log profile includes the log subscriber ID parameter, which specifies whether to or not to include the MSISDN information in the LSN logs of an LSN configuration.
- Bind the LSN log profile to an LSN group of an LSN configuration.Bind the created LSN log profile to an LSN group of an LSN configuration by setting the log profile name parameter to the created LSN log profile name. MSISDN information is included in all LSN logs related to mobile subscribers of this LSN group.

To create an LSN log profile by using the CLI

At the command prompt, type:

```
add lsn logprofile <logprofilename> -logSubscriberID ( ENABLED |
   DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind an LSN log profile to an LSN group of an NAT64 LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

Sample Configuration

In this example of NAT64 LSN configuration, the LSN log profile LOG-PROFILE-MSISDN-1 has the log subscriber ID parameter enabled. LOG-PROFILE-MSISDN-1 is bound to LSN group LSN-NAT64-GROUP-1. MSISDN information is included in the LSN session and LSN mapping logs for connections from mobile subscribers (in network 2001:DB8:5001::/96).

```
add lsn logprofile  LOG-PROFILE-MSISDN-1  -logSubscriberID ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
```



```
bind lsn group LSN-NAT64-GROUP-1 -logprofilename  LOG-PROFILE-MSISDN-1
Done
<!--NeedCopy-->
```

Compact Logging for Large Scale NAT

Logging LSN information is one of the important functions needed by ISPs to meet legal requirements and be able to identify the source of traffic at any given time. This eventually results in a huge volume of log data, requiring the ISPs to make large investments to maintain the logging infrastructure.

Compact logging is a technique for reducing the log size by using a notational change involving short codes for event and protocol names. For example, C for client, SC for session created, and T for TCP. Compact logging results in an average of 40 percent reduction in log size.

Configuration Steps

Perform the following tasks for logging LSN information in compact format:

1. Create an LSN log profile. An LSN log profile includes the Log Compact parameter, which specifies whether to or not to log information in compact format for an LSN configuration.
2. Bind the LSN log profile to an LSN group of an LSN configuration. Bind the created LSN log profile to an LSN group of an LSN configuration by setting the Log Profile Name parameter to the created LSN log profile name. All sessions and mappings for this LSN group are logged in compact format.

To create an LSN log profile by using the CLI

At the command prompt, type:

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

To bind an LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```


Sample Configuration for NAT64:

```
add lsn logprofile  LOG-PROFILE-COMPACT-1 -logCompact ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - logProfileName LOG-PROFILE-COMPACT-1
Done
<!--NeedCopy-->
```

Logging HTTP Header Information

The Citrix ADC appliance can log request header information of an HTTP connection that is using the Citrix ADC large scale NAT64 functionality. The following header information of an HTTP request packet can be logged:

- URL that the HTTP request is destined to
- HTTP Method specified in the HTTP request
- HTTP version used in the HTTP request
- IPv6 address of the subscriber that sent the HTTP request

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set of subscribers. For example, an ISP can use this feature to find out the most popular website among a set of subscribers.

Configuration Steps

Perform the following tasks for configuring the Citrix ADC appliance to log HTTP header information:

- Create an HTTP header log profile. An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging.
- Bind the HTTP header to an LSN group of a large scale NAT64 configuration. Bind the HTTP header log profile to an LSN group of an LSN configuration by setting the HTTP header log profile name parameter to the name of the created HTTP header log profile. The Citrix ADC appliance then logs HTTP header information of any HTTP requests related to the LSN group. An HTTP header log profile can be bound to multiple LSN groups, but an LSN group can have only one HTTP header log profile.

To create an HTTP header log profile by using the the command line interface

At the command prompt, type:

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

To bind an HTTP header log profile to an LSN group by using the the command line interface

At the command prompt, type:

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

Sample Configuration

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
```



```
11 add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
      :300::/96
12 Done
13 add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
      ip6profile LSN-NAT64-PROFILE-1
14 Done
15 bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
16 Done
17 bind lsn group LSN-NAT64-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG
      -1
18 Done
19 <!--NeedCopy-->
```

Displaying Current Large Scale NAT64 Sessions

You can display the current large scale NAT64 sessions in order to detect any unwanted or inefficient sessions on the Citrix ADC appliance. You can display all or some large scale NAT64 sessions on the basis of selection parameters.

> **Note**
>
> When more than a million large scale NAT64 sessions exist on the Citrix ADC appliance, Citrix recommends using the selection parameters to display selected large scale NAT64 sessions instead of displaying all of them.

To display all large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
1 show lsn session  - nattype NAT64
2 <!--NeedCopy-->
```

To display selective large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
1 show lsn session  - nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
      <string>] [-natIP <ip_addr> [-natPort <port>]]
2 <!--NeedCopy-->
```


Displaying Large Scale NAT64 Statistics

You can display statistics related to large scale NAT64 module, and evaluate its performance or troubleshoot problems. You can display a summary of statistics of all large scale NAT64 configurations or of a particular large scale NAT64 configuration. The statistical counters reflect events since the Citrix ADC appliance was last restarted. All these counters are reset to 0 when the Citrix ADC appliance is restarted.

To display total statistics of large scale NAT64 by using the command line interface

At the command prompt, type:

```
stat lsn nat64
<!--NeedCopy-->
```

To display statistics for a specified large scale NAT64 configuration by using the command line interface

At the command prompt, type:

```
stat lsn group <groupname>
<!--NeedCopy-->
```

Clearing Large Scale NAT64 Sessions

You can remove any unwanted or inefficient large scale NAT64 sessions from the Citrix ADC appliance. The appliance immediately releases resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected large scale NAT64 sessions from the Citrix ADC appliance.

To clear all large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
flush lsn session -nattype NAT64

show lsn session -nattype NAT64
<!--NeedCopy-->
```


To clear selective large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
flush lsn session -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname <string>] [-natIP <ip_addr> [-natPort <port>]]

show lsn session -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

Sample configuration:

Clear all large scale NAT64 sessions existing on a Citrix ADC appliance

```
flush lsn session -nattype NAT64
Done
<!--NeedCopy-->
```

Clear all large scale NAT64 sessions related to client entity LSN-NAT64-CLIENT-1

```
flush lsn session -nattype NAT64 -clientname LSN-NAT64-CLIENT-1
Done
<!--NeedCopy-->
```

Clear all large scale NAT64 sessions related to a subscriber network (2001:DB8:5001::/96) of LSN client entity LSN-NAT64-CLIENT-2

```
flush lsn session -nattype NAT64 -network6 2001:DB8:5001::/96 -clientname LSN-NAT64-CLIENT-2
Done
<!--NeedCopy-->
```

IPFIX Logging

The Citrix ADC appliance supports sending information about LSN events in Internet Protocol Flow Information Export (IPFIX) format to the configured set of IPFIX collector(s). The appliance uses the existing AppFlow feature to send LSN events in IPFIX format to the IPFIX collectors.

IPFIX based logging is available for the following NAT64 related events:

- Creation or deletion of an LSN session.
- Creation or deletion of an LSN mapping entry.
- Allocation or de-allocation of port blocks in the context of deterministic NAT.
- Allocation or de-allocation of port blocks in the context of dynamic NAT.
- Whenever subscriber session quota is exceeded.

Points to Consider before you Configure IPFIX logging

Before you start configuring IPSec ALG, consider the following points:

- You must configure the AppFlow feature and IPFIX collector(s) on the Citrix ADC appliance. For instructions, see Configuring the AppFlow feature.

Configuration Steps

Perform the following tasks for logging LSN information in IPFIX format:

- **Enable LSN logging in the AppFlow configuration**. Enable the LSN logging parameter as part of AppFlow configuration.
- **Create an LSN log profile**. An LSN log profile includes the IPFIX parameter that enables or disables the log information in IPFIX format.
- **Bind the LSN log profile to an LSN group of an LSN configuration**. Bind the LSN log profile to one or multiple LSN group(s). Events related to the bound LSN group will be logged in IPFIX format.

To enable LSN logging in the AppFlow configuration by using the CLI

At the command prompt, type:

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

To create an LSN log profile by using the CLIAt the command prompt, type

At the command prompt, type:

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind the LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

```



```
show lsn group
<!--NeedCopy-->
```

To create an LSN log profile by using the GUI

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **Log** tab, and then add a log profile.

To bind the LSN log profile to an LSN group of an LSN configuration by using the GUI

1. Navigate to **System** > **Large Scale NAT** > **LSN Group,** open the **LSN** group.
2. In **Advanced Settings,** click **+ Log Profile** to bind the created Log profile to the LSN group.

Port Control Protocol for Large Scale NAT64

September 14, 2021

Citrix ADC appliances now support Port Control Protocol (PCP) for large scale NAT (LSN). Many of an ISP's subscriber applications must be accessible from Internet (for example, Internet of Things (IOT) devices, such as an IP camera that provides surveillance over the Internet). One way to meet this requirement is to create static large scale NAT (LSN) maps. But for a very large number of subscribers, creating static LSN NAT maps is not a feasible solution.

Port Control Protocol (PCP) enables a subscriber to request specific LSN NAT mappings for itself and/or for other 3rd party devices. The large scale NAT device creates an LSN map and sends it to the subscriber. The subscriber sends the remote devices on the Internet the NAT IP address:NAT port at which they can connect to the subscriber.

Applications usually send frequent keep-alive messages to the large scale NAT device so that their LSN mappings do not time out. PCP helps reduce the frequency of such keep-alive messages by enabling the applications to learn the timeout settings of the LSN mappings. This helps reduce bandwidth consumption on the ISP's access network and battery consumption on mobile devices.

PCP is a client-server model and runs over the UDP transport protocol. A Citrix ADC appliance implements the PCP server component and is compliant with RFC 6887.

Configuration Steps

Perform the following tasks for configuring PCP:

- **(Optional) Create a PCP profile**. A PCP profile includes settings for PCP related parameters (for example, to listen for mapping and peer PCP requests). A PCP profile can be bound to a

PCP server. A PCP profile bound to a PCP server applies all its settings to the PCP server. A PCP profile can be bound to multiple PCP servers. By default, one PCP profile with default parameters settings is bound to all PCP servers. A PCP profile that you bind to a PCP server overrides the default PCP profile settings for that server. A default PCP profile has the following parameter settings:

- Mapping: Enabled
- Peer: Enabled
- Minimum map life: 120 seconds
- Maximum max life: 86400 seconds
- Announce count: 10
- Third Party: Disabled

- **Create a PCP server and bind a PCP profile to it**. Create a PCP server on the Citrix ADC appliance to listen for PCP related requests and messages from the subscribers. A Subnet IP (SNIP) or (SNIP6) address must be assigned to a PCP server to access it. By default, a PCP server listens on port 5351.
- **Bind the PCP server to an LSN group of an LSN configuration**. Bind the created PCP server to an LSN group of an LSN configuration by setting the PCP Server parameter to specify the created PCP server. The created PCP server can be accessed only by the subscribers of this LSN group.

> **Note**
>
> A PCP server for a large scale NAT configuration does not serve requests from subscribers that are identified from ACL rules.

To create a PCP profile by using the CLI

At the command prompt, type:

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

To create a PCP server by using the CLI

At the command prompt, type:

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]
```



```
2
3 show pcp server <name>
4 <!--NeedCopy-->
```

Sample Configuration for NAT64

In the following sample configuration, PCP server PCP-SERVER-1, with PCP settings from PCP-PROFILE-1,is bound to LSN group LSN-NAT64-GROUP-1. PCP-SERVER-1 serves PCP requests from IPv6 subscribers in network 2001:DB8:5001::/96.

Sample configuration:

```
1 add pcp profile PCP-PROFILE-1 -minMapLife 400
2 Done
3 add pcp server PCP-SERVER-1 2001:DB8:6001::90  -pcpProfile PCP-PROFILE
     -1
4 Done
5 add lsn client LSN-NAT64-CLIENT-1
6 Done
7 bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
8 Done
9 add lsn pool LSN-NAT64-POOL-1
10 Done
11 bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
12 Done
13 add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
     :300::/96
14 Done
15 add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
     ip6profile LSN-NAT64-PROFILE-1
16 Done
17 bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
18 Done
19 bind lsn group LSN-NAT64-GROUP-1  - pcpServer PCP-NAT64-SERVER-1
20 Done
21 <!--NeedCopy-->
```

LSN64 in a cluster setup

September 14, 2021

Large scale NAT64 configurations are supported on a Citrix ADC cluster setup.

A Citrix ADC cluster is a group of Citrix ADC appliances that are configured and managed as a single system. A Citrix ADC cluster provides scalability and availability. Each Citrix ADC appliance in a cluster setup acts as an independent LSN entity and is managed as a single system.

The LSN configuration in a cluster setup is same as in a standalone appliance except for a specific pool of LSN IP addresses are owned by only one node at a time. In other words, an LSN IP pool entity is configured as a spotted entity in a particular node. All the nodes of a cluster setup can have a specific LSN IP pool entity. To make sure that the packets related to an LSN session are received on the same cluster node that performed the NAT operation, policy based backplane (PBS) steering is configured. PBS steers the received related packets of an LSN session to the same cluster node.

Sample configuration:

```
add lsn client LSN-NAT64-CLIENT-1

Done

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

Done

add lsn pool LSN-NAT64-POOL-1

Done

bind lsn pool LSN-NAT64-POOL-1 -ownerNode 1  203.0.113.61 -
    203.0.113.70

Done

bind lsn pool LSN-NAT64-POOL-1 -ownerNode 2  203.0.113.101 -
    203.0.113.110

Done

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96

Done

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1

Done

```



```
29 bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
30
31 Done
32
33 add ns acl6 NAT64-DFD ALLOW -srcIPv6 = 2001:DB8:5001:: -type DFD -
      dfdhash SIP -dfdprefix 64
34
35 Done
36
37 apply ns acls6 -type DFD
38
39 Done
40 <!--NeedCopy-->
```

Mapping Address and Port using Translation

September 14, 2021

Mapping Address and Port using Translation (MAP-T) is an IPv6 transition solution for ISPs with IPv6 infrastructure to connect their IPv4 subscribers to the IPv4 internet_.MAP-T is built on stateless IPv4 and IPv6 address translation technologies. MAP-T is a mechanism that performs double translation (IPv4 to IPv6 and vice versa) on customer edge (CE) devices and border routers (in ISP core network).

In a MAP-T deployment, the CE device implements a combination of stateful NAPT44 translation and stateless NAT46 translation. The CE device obtains NAT-IP and the port-block to be used for translation through DHCPv6 or any other method.

When an IPv4 packet from a subscriber device arrives at the CE device, the CE device performs NAPT44 and stores the NAPT44 binding information. After NAT44 translation, the packet is subjected to NAT46 translation and then forwarded to the border router (BR) device located in the ISP's core network. The BR device receives the IPv6 packets from the CE device, extracts and validates the NAT-IP and port-block embedded in the IPv6 header, and forwards the IPv4 packet to the IPv4 Internet. When the BR receives the IPv4 packet from the Internet, it translates the IPv4 packet to an IPv6 packet and send the IPv6 packet to the CE device.

MAP-T is stateless on a BR device, so it does not require the BR device to perform NAT on the traffic. Instead, NAT functionality is delegated to the CE devices. This delegation and stateless functionality in BR devices allows the BR deployment to scale in proportion to the volume of traffic.

The Citrix ADC appliance implements the BR functionality of a MAP-T solution as described by RFC 7599.

Configuring MAP-T

Configuring MAP-T on a Citrix ADC appliance consists of the following tasks:

- Add a default mapping rule
- Add a basic mapping rule
- Bind an IPv4 NAT address range of CE devices to a basic mapping rule
- Add a map domain and bind a basic mapping rule and default mapping rule to the domain

To add a default mapping rule by using the CLI

At the command prompt, type:

```
add MapDmr <name> -BRIpv6Prefix ( <ipv6_addr> | <*> )

show MapDmr <name>
<!--NeedCopy-->
```

To add a basic mapping rule by using the CLI

At the command prompt, type:

```
add MapBmr <name> -RuleIpv6Prefix <ipv6_addr> | <*>  [-psidoffset <
    positive_integer>] [-EAbitLength <positive_integer>]  [-psidlength <
    positive_integer>]

show MapBmr <name>
<!--NeedCopy-->
```

To bind IPv4 NAT address range of CE devices to a basic mapping rule by using the CLI

At the command prompt, type:

```
bind MapBmr <name> (-network <ip_addr>  [-netmask <netmask>])

show MapBmr <name>
<!--NeedCopy-->
```

To add a map domain by using the CLI

At the command prompt, type:


```
add MapDomain <name> -MapDmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```

To bind a basic mapping rule to a map domain by using the CLI

At the command prompt, type:

```
bind MapDomain <name> -MapBmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```

Sample configuration

```
add mapdmr DMR-1 -BRIpv6Prefix 2002:db8::/64

Done

add mapbmr BMR-1 -ruleIpv6Prefix 2002:db8:89ab::/48 -eAbitLength 16 -
    psidlength 8 -psidoffset 6

Done

bind mapbmr BMR-1 -network 192.0.1.0 -netmask 255.255.255.0

Done

add MapDomain MAP-DOMAIN-1 -mapdmrname DMR-1

Done

bind MapDomain MAP-DOMAIN-1 -mapbmrname BMR-1
Done
<!--NeedCopy-->
```

Telco subscriber management

September 14, 2021

The number of subscribers in a telco network is increasing at an unprecedented rate, and managing them is becoming a challenge for service providers. Newer, faster, and smarter devices are placing high demand on the network and the subscriber management systems. It is no longer feasible to provide each subscriber the same standard of service, and the need for traffic processing on a per-subscriber basis is imperative.

The Citrix ADC appliance provides the intelligence to profile subscribers based on their information stored in the Policy and Charging Rules Function (PCRF). When a mobile subscriber connects to the Internet, the packet gateway associates an IP address with the subscriber and forwards the data packet to the appliance. The appliance receives the subscriber information dynamically, or you can configure static subscribers. This information enables the appliance to apply its rich traffic management capabilities, such as content switching, integrated caching, rewrite, and responder, on a per-subscriber basis to manage the traffic.

Before you configure the Citrix ADC appliance to manage subscribers, you must allocate memory to the module that stores subscriber sessions. For dynamic subscribers, you must configure an interface through which the appliance receives session information. Static subscribers must be assigned IDs, and you can associate them with policies.

You can also do the following:

- Subscriber policy enforcement and management.
- Configure the appliance to uniquely identify a subscriber by using only the IPv6 prefix instead of the complete IPv6 address.
- Use policies to optimize TCP traffic for both dynamic and static subscribers. These policies associate different TCP profiles with different types of users.
- Manage idle sessions on a Citrix ADC appliance.
- Enable logging to a log server.
- Remove LSN sessions for deleted subscriber sessions.

Allocating memory for the subscriber session store module

Each subscriber session entry consumes 1 KB of memory. Storing 500,000 subscriber sessions at any point in time requires 500 MB of memory. This value must be added to the minimum memory requirement, which is shown as part of the output of the “show extendedmemoryparam” command. In the following example, the output is for a Citrix ADC VPX instance with 3 packet engines and 8 GB memory.

To store 500,000 subscriber sessions on this appliance, the configured memory must be 2058+500 MB (500,000 x 1 KB = 500 MB.)

> **Note**

> The configured memory must be in multiples of 2 MB and must not exceed the maximum memory usage limit. The appliance must be restarted for the changes to take effect.

Example

```
show extendedmemoryparam
       Extended Memory Global Configuration. This memory is utilized by
           LSN and Subscriber Session Store Modules:
        Active Memory Usage: 0 MBytes
        Configured Memory Limit: 0 MBytes
        Minimum Memory Required: 2058 MBytes
        Maximum Memory Usage Limit: 2606 MBytes
    Done
    set extendedmemoryparam -memLimit 2558
    Done
    show extendedmemoryparam
            Extended Memory Global Configuration. This memory is
               utilized by LSN and Subscriber Session Store Modules:

            Active Memory Usage: 2558 MBytes
            Configured Memory Limit: 2558 MBytes
            Minimum Memory Required: 2058 MBytes
            Maximum Memory Usage Limit: 2606 MBytes
    Done
<!--NeedCopy-->
```

Configure an interface for dynamic subscribers

The Citrix ADC appliance dynamically receives the subscriber information through any of the following types of interface:

- Gx Interface
- RADIUS Interface
- RADIUS and Gx Interface

> **Note**
>
> - Starting with NetScaler release 12.0 build 57.19, Gx interface is supported for a cluster deployment. For more information see Gx interface in a cluster topology.
> - In an HA setup, the subscriber sessions are continually synchronized on the secondary node. In the event of a failover, the subscriber information is still available on the secondary node.

Gx interface

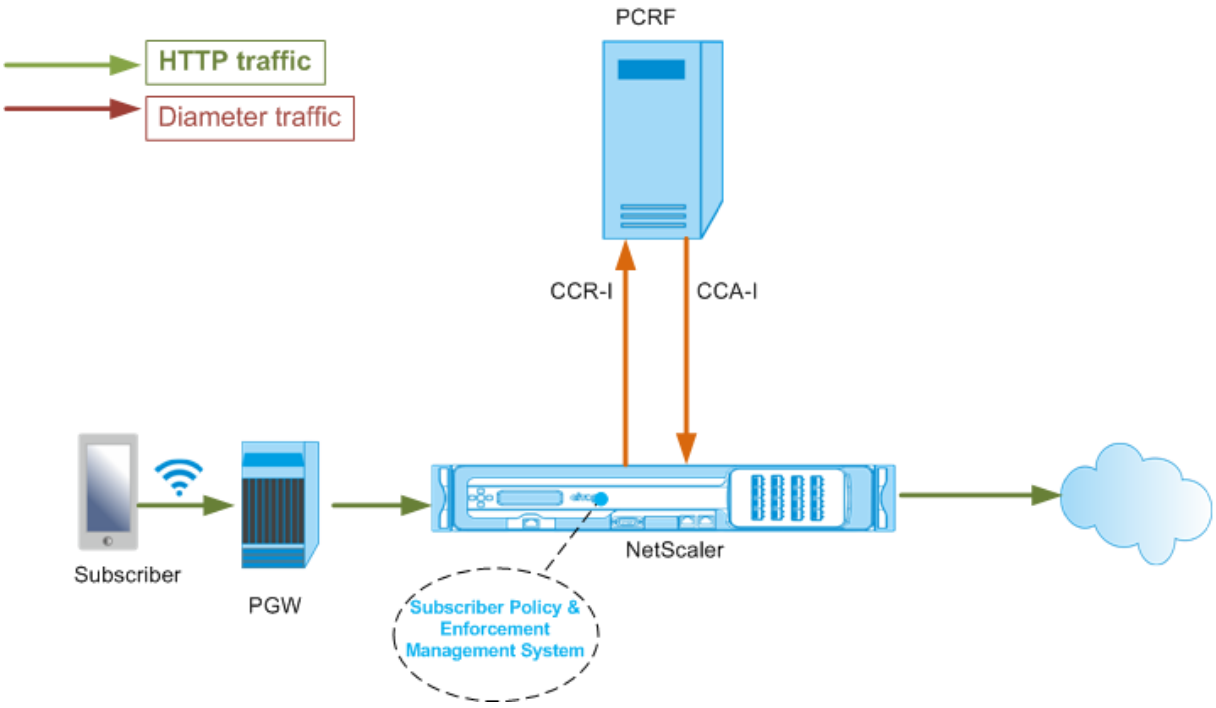
A Gx interface (as specified in 3GPP 29.212) is a standard interface based on the Diameter protocol that allows exchange of policy control and charging rules between a PCRF and a Policy and Charging Enforcement Function (PCEF) entity in a Telco network.

When an IP-CAN session is established, the packet gateway forwards the subscriber ID, such as the MSISDN, and Framed-IP address information about the subscriber to the PCRF as a Diameter message. When the data packet arrives at the appliance from packet gateway (PGW), the appliance uses the subscriber IP address to query the PCRF to get the subscriber information. This is also known as secondary PCEF functionality.

The Policy and Charging Control (PCC) rules received by the appliance over the Gx interface are stored on the appliance during the subscriber session, that is, until the PCRF sends a Re-Auth-Request (RAR) message with a Session-Release-Cause AVP or the subscriber session is terminated from the CLI or the configuration utility. If there are any updates to an existing subscriber, the PCRF sends the updates in an RAR message. A subscriber session is initiated when a subscriber logs on to the network, and terminated when the subscriber logs off.

> **Note**: If the PCRF server is down, the Citrix ADC appliance creates negative sessions for the pending or incoming Gx subscriber requests. When the PCRF server is back up again, the Citrix ADC appliance prevents a storm of requests by waiting for the negative sessions to expire before performing the specific subscriber requests.

The following illustration shows the high-level traffic flow. It assumes that the data plane traffic is HTTP. The appliance sends a Credit Control Request (CCR) over a Gx interface to the PCRF server and, in the credit control answer (CCA), receives the PCC rules and, optionally, other information, such as the Radio Access Technology (RAT) type, that applies to the particular subscriber. PCC rules include one or more policy (rule) names and other parameters. The appliance uses this information to retrieve the predefined rules stored on the appliance, and to direct the flow of traffic. It also stores this information in the subscriber policy and enforcement management system during the subscriber session. After a subscriber session is terminated, the appliance discards all the information about the subscriber.



The following example shows the commands for configuring a Gx interface. The commands are in boldface.

To set up a Gx interface, perform the following tasks

Add a DIAMETER service for each Gx interface. For example:

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

Add a non-addressable DIAMETER load balancing virtual server and bind the services created in step 1 to this virtual server. For more than one service, specify a persistenceType and the persistAVPno so that specific sessions are handled by the same PCRF server. For example:

```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

Configure Citrix ADC diameter identity and realm. Identity and realm are used as Origin-Host and Origin-Realm AVPs in diameter messages sent by the Gx client. For example:


```
set ns diameter  - identity netscaler.com  - realm com
<!--NeedCopy-->
```

Configure the Gx interface to use the virtual server created in step 2 as the PCRF virtual server. Specify the PCRF realm to use as Destination-Realm AVP in diameters messages sent by the Gx client. For example:

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com
<!--NeedCopy-->
```

Set the subscriber interface type to GxOnly. For example:

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

To see the Gx interface configuration and status, type:

```
show subscriber gxinterface
<!--NeedCopy-->
```

Example

```
show subscriber gxinterface
    Gx Interface parameters:
        PCRF Vserver: vdiam (DOWN)
        Gx Client Identity...:  netscaler1.com
        Gx Client Realm ..........:  com
        PCRF Realm: epc.mnc030.mcc234.3gppnetwork.org
        Hold Packets On Subscriber Absence: YES
        CCR Request Timeout: 4 Seconds
        CCR Request Retry Attempts: 1
        Gx HealthCheck enabled: NO
        Gx HealthCheck TTL : 30 Seconds
        CER Request Timeout: 10 Seconds
        RevalidationTimeout: 30 Seconds
        NegativeTTL: 60 Seconds
        NegativeTTL Limited Success: NO
        Purge SDB on Gx Failure: YES
        ServicePath AVP code: 262099    ServicePath AVP VendorID: 3845
        PCRF Connection State: PCRF is not ready
    Done

<!--NeedCopy-->
```


ARGUMENTS

vServer

Name of the load balancing or content switching virtual server to which the Gx connections are established. The service type of the virtual server must be DIAMETER or SSL_DIAMETER. This parameter is mutually exclusive with the service parameter. Therefore, you cannot set both service and the virtual server in the Gx interface.

Service

Name of DIAMETER or SSL_DIAMETER service corresponding to PCRF to which the Gx connection is established. This parameter is mutually exclusive with the vserver parameter. Therefore, you cannot set both service and the virtual server in the Gx Interface.

pcrfRealm

The realm of PCRF to which the message is to be routed. This is the realm used in Destination-Realm AVP by Citrix ADC Gx client (as a Diameter node).

holdOnSubscriberAbsence

Set to Yes to hold packets until the subscriber session information is fetched from the PCRF server. If set to No, the default subscriber profile is applied until the subscriber session information is fetched from the PCRF server. If a default subscriber profile is not configured, an UNDEF is raised for expressions that use subscriber attributes.

requestTimeout

Time, in seconds, within which the Gx CCR request must complete. If the request does not complete within this time, the request is retransmitted for the number of times specified in the requestRetryAttempts parameter. If request is not complete even after retransmitting, then the default subscriber profile is applied to this subscriber. If a default subscriber profile is not configured, an UNDEF is raised for expressions that use subscriber attributes. Zero disables the timeout. Default value: 10

requestRetryAttempts

Specify the number of times a request must be retransmitted if the request does not complete within the value specified in the requestTimeout parameter. Default value: 3.

healthCheck

Set to Yes to enable inline health check of Gx peer. When enabled, Citrix ADC sends DWR packets to the PCRF server. When the Gx session is idle, the healthCheck timer expires and DWR packets are initiated to check if the PCRF server is active. Default value: No.
Note: This parameter is supported in Citrix ADC 12.1 build 51.xx and later.

healthCheckTTL

Time in seconds defined for watchdog supervision. After the health check TTL time expires, DWR is sent to check the status of the PCRF server. Any CCR, CCA, RAR, or RAA message resets the timer.
Minimum value: 6 seconds. Default value: 30 seconds.
Note: This parameter is supported in Citrix ADC 12.1 build 51.xx and later.

cerRequestTimeout

Time in seconds defined for retransmission of the capabilities exchange request. Citrix ADC initiates a new CER message if it does not receive a CEA from the PCRF within this configured time.
If no response is received from the PCRF server, the appliance tries to send the CER message 5 times. If there is no response even after 5 CER messages, the appliance closes the TCP connection and reports a failure. If the timeout value is set to 0, the application health check feature is disabled.
Minimum value: 0 seconds. Default value: 0 seconds.
Note: This parameter is supported in Citrix ADC 12.1 build 51.xx and later.

revalidationTimeout

Time, in seconds, after which the Gx CCR-U request is sent after any PCRF activity on a session. Any RAR or CCA message resets the timer. Zero value disables the idle timeout.

negativeTTL

Time, in seconds, after which the Gx CCR-I request is resent for sessions that have not been resolved by PCRF because the server is down or there is no response or a failed response is received. Instead of polling the PCRF server constantly, a negative-TTL makes the appliance hold on to an unresolved session. For negative sessions, the appliance inherits the attributes from the default subscriber profile, if one is configured and from the RADIUS accounting message, if one is received. Zero value disables the negative sessions. The appliance does not install negative sessions even if a subscriber session could not be fetched. Default value: 600

negativeTTLLimitedSuccess

Set to Yes to create negative session for partial success response code (2002). If set to No, regular session is created. Default value: No.

This parameter is supported in Citrix ADC 12.1 build 49.xx and later.

purgeSDBonGxFailure

Set to Yes to flush subscriber database when the Gx interface fails. Gx interface failure includes both DWR monitoring (if enabled) and network healthCheck (if enabled). When set to Yes, all subscriber sessions are cleared.
Default value: No.
Note: This parameter is supported in Citrix ADC 12.1 build 51.xx and later.

servicePathAVP

The AVP code in which PCRF sends the service path applicable to a subscriber.

servicePathVendorid

The vendor id of the AVP in which PCRF sends the service path applicable to a subscriber.

To configure Gx interface by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.
3. In Interface Type, select **GxOnly**.
4. Specify the values for the all required parameters.
5. Click **OK**.

Detect transport failures over established Gx connections

Note: This feature is supported in Citrix ADC 12.1 build 51.xx and later.

A Citrix ADC appliance can be configured to detect transport failures over established Gx connections by using device watchdog request (DWR) and device watchdog answer (DWA) messages.

After a Gx session is established, a predefined timer is triggered to detect if a session is idle. A DWR message is sent after the idle time timer expires. The idle time timer is reset each time the Citrix ADC appliance receives a message over an established Gx session. The peer's availability is confirmed based on the DWA message after a DWR message is sent.

- If the DWA is received, a peer's availability is confirmed and the watchdog timer is reset.

- If the DWA is not received and the watchdog timer expires twice consecutively, the session is considered as down and peer unavailable. The appliance closes the session and tries to establish a new session with the Gx peer.

When the watchdog timer expires twice without a response, the Citrix ADC appliance considers the Gx connection as faulty and initiates a connection closure. Once the connection is closed, no other watchdog request is sent towards the Gx peer. Citrix ADC appliance uses the next available Gx session for any PCRF requests.

To detect transport failures over established Gx connections by using the CLI

At the command prompt, type:

```
set subscriber gxInterface [-vServer <string>] [-service <string>] [-
    healthCheck ( YES | NO )] [-healthCheckTTL<positive_integer>][-
    cerRequestTimeout <positive_integer>] [-purgeSDBonGxFailure ( YES |
    NO )]
<!--NeedCopy-->
```

Example:

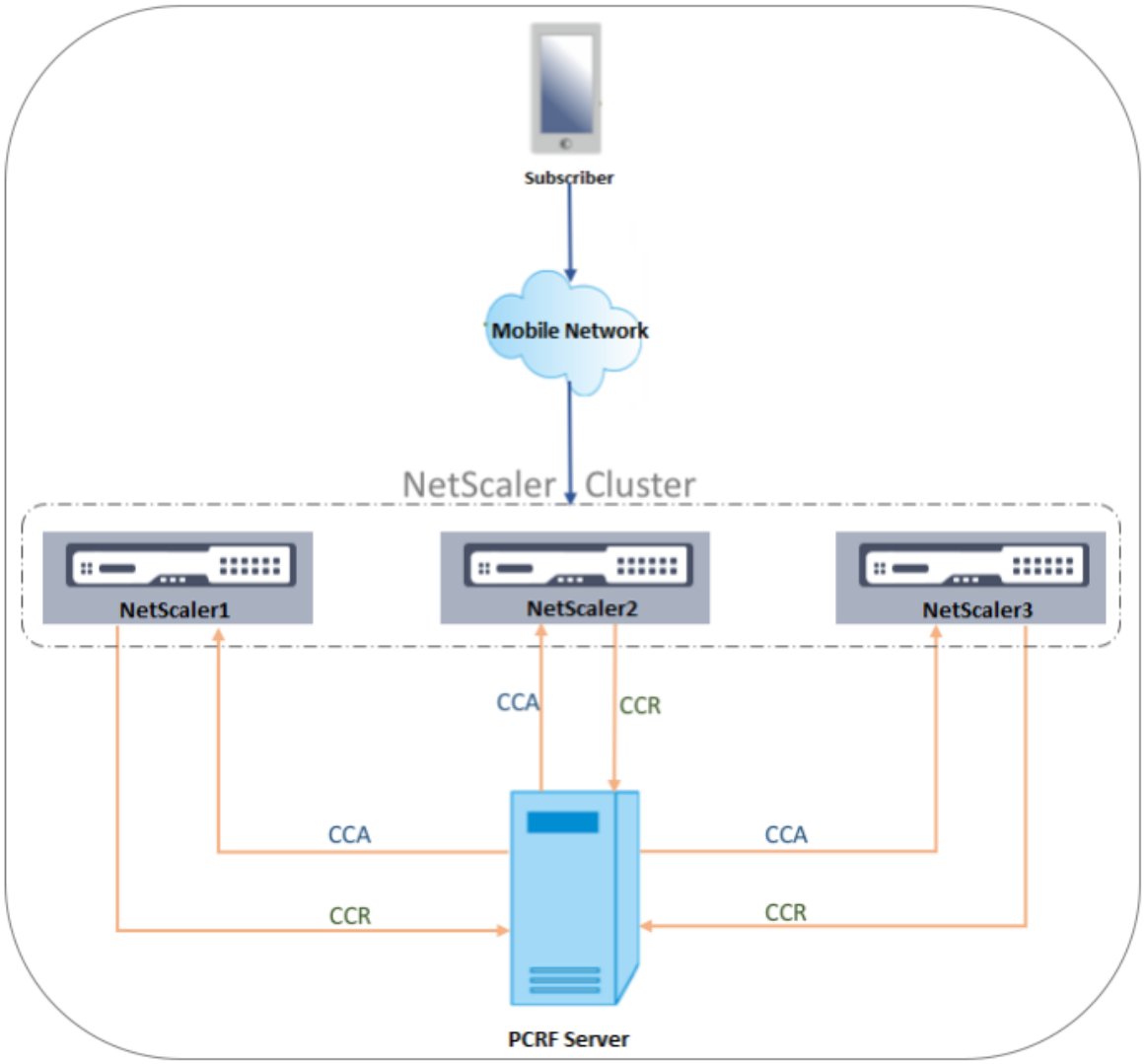
```
set subscriber gxInterface set subscriber gxInterface -vServer vdiam -
    healthCheck YES -healthCheckTTL 31 -cerRequestTimeout 15
    purgeSDBonGxFailure  YES
<!--NeedCopy-->
```

To detect transport failures over established Gx connections by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.
3. In **Interface Type**, select **GxOnly**.
4. Specify the values for all required parameters.
5. Select **Health Check** and specify values for **Health Check TTL** and **CER Request Timeout**.
6. Click **OK**.

Gx interface in a cluster topology

The Citrix ADC appliance supports Gx interface in a cluster topology.



The Citrix ADC nodes in the cluster communicate with an external PCRF server through the Gx interface. When a node receives client traffic, the appliance performs the following:

- Sends a CCR-I request to the PCRF server to fetch subscriber information.
- The PCRF server responds with a CCR-A.
- The Citrix ADC node then stores the received subscriber information in its subscriber store and applies the rules to the client traffic.

Each node maintains an independent subscriber store and subscriber sessions are not synchronized to other nodes.

As per the Diameter Base Protocol RFC 6733, each peer must be configured with a unique diameter identity to communicate with other peers over the diameter protocol. Hence, in a cluster deployment, configuration of diameter identity is spotted. The diameter parameters (identity, realm, server close propagation) for each node can be configured individually by using the GUI or the CLI.

When a node is added to a cluster, it assumes the default diameter parameters (identity=netscaler.com, realm=com, serverClosePropogation=NO). After the nodes are added, the diameter parameters for each node must be configured.

To configure the diameter parameters by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Change Diameter Parameters**.
3. In the Diameter Parameters page, select the Citrix ADC node for which you want to configure the diameter parameters and then click **Configure**.
4. In the Configure Diameter Parameters page, configure the diameter Identity, diameter Realm, and server Close Propagation for the selected node.
5. Click **OK**.

To configure the diameter parameters by using the CLI

At the command prompt, type:

```
set ns diameter [-identity <string>] [-ownerNode <positive_integer>]
<!--NeedCopy-->
```

ARGUMENTS

Identity

Diameter Identity is used to identify a Diameter node uniquely. Before setting up diameter configuration, the Citrix ADC appliance (as a Diameter node) must be assigned a unique diameter identity.

For example, set ns diameter -identity netscaler.com -ownerNode 1. So, whenever Citrix ADC system needs to use identity in diameter messages, it uses ‘netscaler.com’ as Origin-Host AVP as defined in RFC3588.

Maximum Length: 255

OwnerNode

OwnerNode represents the ID of the cluster node for which the diameter ID is set. OwnerNode can be configured only through CLIP.

Minimum value: 0

Maximum value: 31

Example:

set ns diameter -identity netscaler1.com -ownerNode 1

Note:

The OwnerNode option is also added to the show ns diameter command.

Example:

```
show diameter -ownerNode <0-31>
<!--NeedCopy-->
```

When the show ns diameter command is executed, it displays the diameter parameters for a given node.

To configure a Gx interface for cluster deployment

To set up a Gx interface, perform the following tasks:

Add a DIAMETER service for each Gx interface.

Example:

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868
add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

Add a DIAMETER load balancing virtual server and bind the services created in step 1 to this virtual server.

Example:

```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

Configure Citrix ADC diameter identity and realm on all the cluster nodes. Identity and realm are used as Origin-Host and Origin-Realm AVPs in diameter messages sent by the Gx client.

Example:

```
set ns diameter -identity node0.netscaler.com -realm netscaler.com -
    ownerNode 0
```



```
2
3 set ns diameter -identity node1.netscaler.com -realm netscaler.com -
     ownerNode 1
4 <!--NeedCopy-->
```

Configure the Gx interface to use the virtual server created in step 2 as the PCRF virtual server and set the PCRF realm as well.

Example:

```
1 set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com
2
3 Set the subscriber interface type to GxOnly.
4 <!--NeedCopy-->
```

Example:

```
1 set subscriber param -interfaceType GxOnly
2 <!--NeedCopy-->
```

To see the Gx interface configuration and status, type:

```
1 show subscriber gxinterface
2 <!--NeedCopy-->
```

RADIUS interface

With a RADIUS interface, the packet gateway forwards the subscriber information in a RADIUS Accounting Start message to the appliance through the RADIUS interface when an IP-CAN session is established. A service of type RADIUSListener processes RADIUS Accounting messages. Add a shared secret for the RADIUS client. If a shared secret is not configured, the RADIUS message is silently dropped. The following example shows the commands for configuring a RADIUS interface. The commands are in boldface.

To set up a RADIUS interface, perform the following tasks:

Create a RADIUS listener service at the SNIP address where the RADIUS messages are received. For example:

```
1 add service srad1 192.0.0.206 RADIUSLISTENER 1813
2 <!--NeedCopy-->
```

Configure the subscriber RADIUS interface to use this service. For example:


```
set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

Set the subscriber interface type to RadiusOnly. For example:

```
set subscriber param -interfaceType RadiusOnly
<!--NeedCopy-->
```

Add a RADIUS client specifying a subnet and shared secret. For example:

```
add radius client 192.0.2.0/24 -radkey client123
<!--NeedCopy-->
```

A subnet of 0.0.0.0/0 implies that it is the default shared secret for all clients. To see the RADIUS interface configuration and status, type:

```
show subscriber radiusInterface
<!--NeedCopy-->
```

RADIUS Interface parameters:

Radius Listener Service: srad1(UP)

Done

Example:

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

ARGUMENTS

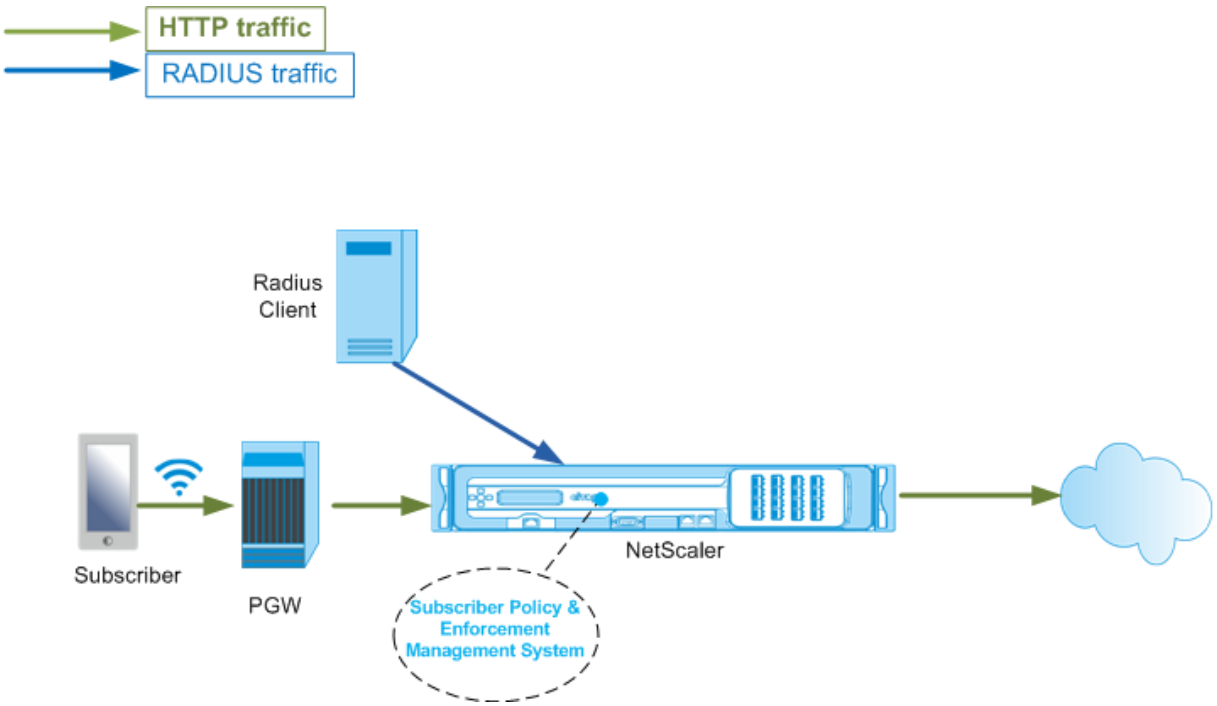
ListeningService

Name of the RADIUS listening service that processes the RADIUS accounting requests.

svrState

The state of the RADIUS listening service.

The following illustration shows the high-level traffic flow.



To configure RadiusOnly interface by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.
3. In Interface Type, select **RadiusOnly**.
4. Specify the values for the all required parameters.
5. Click **OK**.

RADIUS and Gx interface

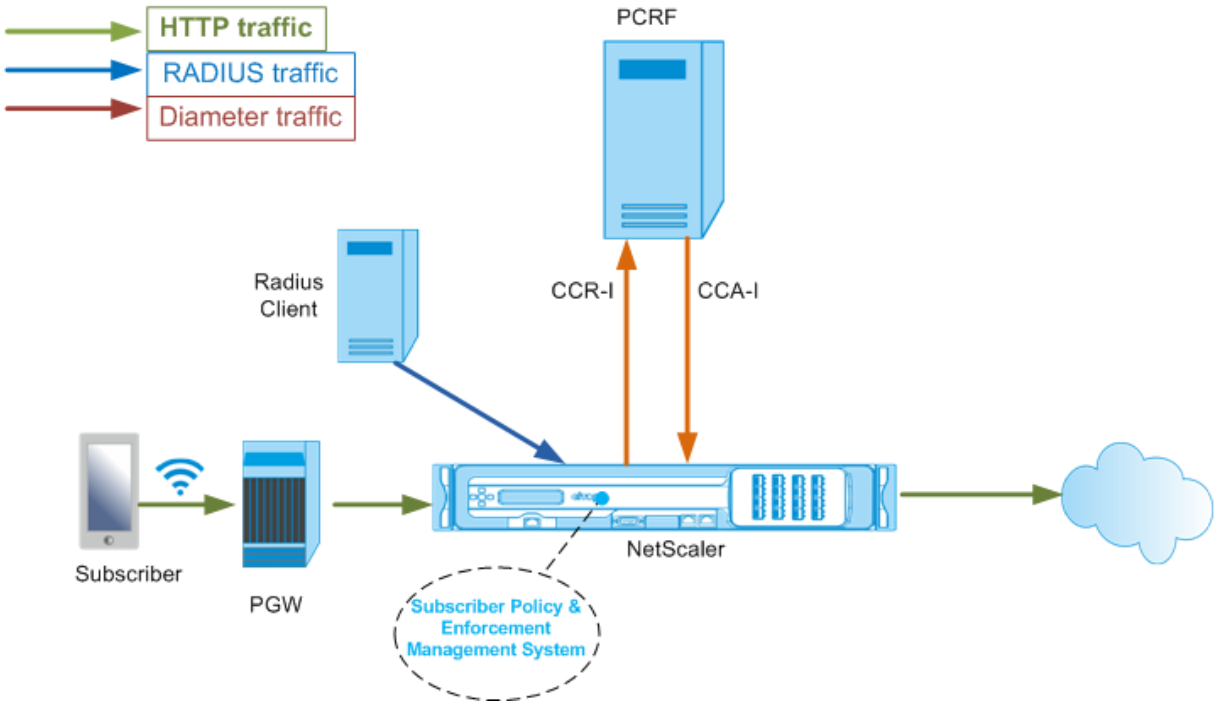
With a RADIUS and Gx interface, when an IP-CAN session is established, the packet gateway forwards the subscriber ID, such as the MSISDN, and Framed-IP address information about the subscriber to the appliance through the RADIUS interface. The appliance uses this subscriber ID to query the PCRF on the Gx interface to get the subscriber information. This is known as primary PCEF functionality. The following example shows the commands for configuring a RADIUS and Gx interface.

```
set subscriber param -interfaceType RadiusandGx
add service pcrf-svc 203.0.113.1 DIAMETER 3868
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263
bind lb vserver vdiam pcrf-svc
set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -
    holdOnSubscriberAbsence YES -revalidationTimeout 60 -negativeTTL 120
```



```
6 add service srad1 192.0.0.206 RADIUSLISTENER 1813    set subscriber
     radiusInterface -listeningService srad1
7 <!--NeedCopy-->
```

The following illustration shows the high-level traffic flow.



To configure RadiusAndGx interface by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.
3. In Interface Type, select **RadiusAndGx**.
4. Specify the values for the all required parameters.
5. Click **OK**.

Configure static subscribers

You can configure the subscribers manually on the Citrix ADC appliance by using the command line or the configuration utility. You create static subscribers by assigning a unique subscriber ID and optionally associating a policy with each subscriber. The following examples show the commands for configuring a static subscriber.

In the following examples, **subscriptionIdvalue** specifies the international telephone number, and **subscriptionIdType** (E164 in this example) specifies the general format for international telephone numbers.


```
1     add subscriber profile 203.0.113.6 -subscriberRules policy1 policy2
          -subscriptionIdType E164 -subscriptionIdvalue 98767543211
2     add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1
         policy3 -subscriptionIdtype E164 -subscriptionIdvalue
         98767543212
3     add subscriber profile 203.0.24.2 10 -subscriberRules policy2
         policy3 -subscriptionIdtype E164 -subscriptionIdvalue
         98767543213
4 <!--NeedCopy-->
```

To view the configured subscriber profiles, type:

> show subscriber profile

```
1     > show subscriber profile
2
3     1)  Subscriber IP: 203.0.24.2 VLAN:10
4         Profile Attributes:
5                 Active Rules: policy2, policy3
6                 Subscriber Id Type: E164
7                 Subscriber Id Value: 98767543213
8     2)  Subscriber IP: 2002::/64
9         Profile Attributes:
10                Active Rules: policy1, policy3
11                Subscriber Id Type: E164
12                Subscriber Id Value: 98767543212
13    3)  Subscriber IP: 203.0.113.6
14        Profile Attributes:
15                Active Rules: policy1, policy2
16                Subscriber Id Type: E164
17                Subscriber Id Value: 98767543211
18
19     Done
20 <!--NeedCopy-->
```

Default subscriber profile

A default subscriber profile is used if the subscriber IP address is not found in the subscriber session store on the appliance. In the following example, a default subscriber profile is added with the subscriber rule policy1.

```
1     > add subscriber profile * -subscriberRules policy1
2 <!--NeedCopy-->
```


View and clear subscriber sessions

Use the following command to display all the static and dynamic subscriber sessions.

> show subscriber sessions

```
    > show subscriber sessions
    1)  Subscriber IP: 2002::/64
        Session Attributes:
                Active Rules: policy1, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543212
    2)  Subscriber IP: *
        Session Attributes:
                Active Rules: policy1
    3)  Subscriber IP: 203.0.24.2 VLAN:10
        Session Attributes:
                Active Rules: policy2, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543213
    4)  Subscriber IP: 203.0.113.6
        Session Attributes:
                Active Rules: policy1, policy2
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543211
    5)  Subscriber IP: 192.168.0.11
        Session Attributes:
                Idle TTL remaining: 361 Seconds
                Active Rules: policy1
                Subscriber Id Type: E164
                Subscriber Id Value: 1234567811
                Service Path: policy1
                AVP(44): 34 44 32 42 42 38 41 43  2D 30 30 30 30 30 30
                   31 31
                AVP(257): 00 01 C0 A8 0A 02
                PCRF-Host: host.pcrf.com
                AVP(280): 74 65 73 74 2E 63 6F 6D

    Done
<!--NeedCopy-->
```

Use the following command to clear a single session or the complete session store. If you do not specify an IP address, the complete subscriber session store is cleared.

```
clear subscriber sessions <ip>
<!--NeedCopy-->
```


Subscriber policy enforcement & management system

The Citrix ADC appliance uses the subscriber's IP address as the key to the subscriber policy enforcement and management system.

You can add subscriber expressions to read the subscriber information available in the Subscriber Policy Enforcement & Management System. These expressions can be used with policy rules and actions that are configured for Citrix ADC features, such as integrated caching, rewrite, responder, and content switching.

The following commands are an example of adding a subscriber-based responder action and policy. The policy evaluates to true if the subscriber rule value is"pol1."

```
1     add responder action error_msg respondwith '"HTTP/1.1 403 OKrnrn" +
          " You are  not authorized to access Internet"'
2     add responder policy no_internet_access "SUBSCRIBER.RULE_ACTIVE("
         pol1")" error_msg
3 <!--NeedCopy-->
```

The following example shows the commands to add a subscriber-based rewrite action and policy. The action inserts an HTTP header "X-Nokia-MSISDN" by using the value of AVP(45) in the subscriber session.

```
1     > add rewrite action AddHDR-act insert_http_header X-Nokia-MSISDN "
         SUBSCRIBER.AVP(45).VALUE"
2     > add rewrite policy AddHDR-pol "HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.
         URL).EQUALS_ANY("patset-test")" AddHDR-act
3 <!--NeedCopy-->
```

In the following example, two policies are configured on the appliance. When the appliance checks the subscriber information and the subscriber rule is cache_enable, it performs caching. If the subscriber rule is cache_disable, the appliance does not perform caching.

```
1     > add cache policy nocachepol -rule "SUBSCRIBER.RULE_ACTIVE("
         cache_disable")" - action NOCACHE
2     > add cache policy cachepol -rule "SUBSCRIBER.RULE_ACTIVE("
         cache_enable")" - action CACHE -storeInGroup cg1
3 <!--NeedCopy-->
```

For a complete list of expressions starting with "SUBSCRIBER." see the Policy Configuration Guide.

> **Important**
>
> Citrix ADC software release 12.1 supports IPANDVLAN key lookup method when the subscriber interface is set to GxOnly. For details, see IP address and VLAN ID key lookup method.

IPv6 prefix based subscriber sessions

A telco user is identified by the IPv6 prefix rather than the complete IPv6 address. The Citrix ADC appliance now uses the prefix instead of the complete IPv6 address (/128) to identify a subscriber in the database (subscriber store). For communicating with the PCRF server (for example, in a CCR-I message), the appliance now uses the framed-IPv6-Prefix AVP instead of the complete IPv6 address. The default prefix length is /64, but you can configure the appliance to use a different value.

To configure the IPv6 prefix by using the command line

```
set subscriber param [-ipv6PrefixLookupList <positive_integer> ...]
```

The first example command below sets a single prefix and the second example command sets multiple prefixes.

```
1     set subscriber param -ipv6PrefixLookupList 64
2     set subscriber param -ipv6PrefixLookupList 64 72 96
3 <!--NeedCopy-->
```

To configure the IPv6 prefix by using the configuration utility

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. In the details pane, under **Settings**, click **Configure Subscriber Parameters** and in **IPv6 Prefix Lookup List**, specify one or more prefixes.

IP address and VLAN ID key lookup method

The Citrix ADC appliance uses the subscriber's IP address as the key lookup method to the subscriber policy enforcement and management system. This method is not effective if the IP addresses are overlapping. In such cases, you can use the VLAN ID as an additional subscriber lookup type. IPANDVLAN key lookup method is supported only when the subscriber interface is set to GxOnly. When IPANDVLAN is configured as the lookup method, the Citrix ADC appliance performs the following:

- Includes the originating VLAN ID in the Gx query for IPv4 subscribers.
- Includes the Gx VLAN AVP in all Gx responses. However, if there is a VLAN ID mismatch, the appliance ignores the responses.

For example, if the appliance sends a CCR-I with GxSessionId-a:IPv4-b:VLAN-c and the response contains GxSessionId-a:IPv4-b:VLAN-d, the response is dropped and a default subscriber entry is created.

> **Note**
>
> - Interface type RadiusAndGx and RadiusOnly cannot be configured together with key type

IPANDVLAN.

- If the traffic is from an IPv6 address, the Citrix ADC appliance uses the IP lookup method.

To configure IP or IPANDVLAN as the key lookup method by using the CLI

At the command prompt, type:

```
set subscriber param [-keytype ( IP | IPANDVLAN )] [-interfaceType <
   interfaceType>]
<!--NeedCopy-->
```

Example:

```
set subscriber param -keytype IPANDVLAN -interfaceType GxOnly

set subscriber param -keytype IP -interfaceType GxOnly
<!--NeedCopy-->
```

Note

Changing the keytype parameter from IP to IPANDVLAN and conversely clears all subscriber data.

VLAN parameter

The VLAN parameter is also added for the following commands.

```
add subscriber profile <ip>@ [-vlan]

set subscriber profile <ip>@ [-vlan] [-subscriptionIdType  <
   subscriptionIdType>]

show subscriber profile [<ip>@] [-vlan]

rm subscriber profile <ip>@ [-vlan <positive_integer>]
<!--NeedCopy-->
```

Arguments

ip

Represents the subscriber IP address. This is a mandatory argument and cannot be changed after the subscriber profile is added.

vlan

Represents the VLAN number on which the subscriber is located. The VLAN number cannot be changed after the subscriber profile is added.

Minimum value: 1

Maximum value: 4096

```
add subscriber profile 192.0.2.23 10

set subscriber profile 192.0.2.23 10 -subscriptionIdtype E164

show subscriber profile 192.0.2.23 10

rm subscriber profile 192.0.2.23 10

<!--NeedCopy-->
```

To configure IP or IPANDVLAN as the key lookup method by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters.**
2. Click **Configure Subscriber Parameters.**
3. In **Key Type**, select **IP** or **IPANDVLAN** as per your requirement.
4. Complete the configuration and click **OK**.

Idle session management of subscriber sessions in a Telco network

Subscriber session cleanup on a Citrix ADC appliance is based on control plane events, such as a RADIUS Accounting Stop message, a Diameter RAR (session release) message, or a “clear subscriber session” command. In some deployments, the messages from a RADIUS client or a PCRF server might not reach the appliance. Also, during heavy traffic, the messages might be lost. A subscriber session that is idle for a long time continues to consume memory and IP resources on the Citrix ADC appliance. The idle session management feature provides configurable timers to identify idle sessions, and cleans up these sessions based on the specified action.

A session is considered idle if no traffic from this subscriber is received on the data plane or the control plane. You can specify an update, terminate (inform PCRF and then delete the session), or delete (without informing PCRF) action. The action is taken only after the session is idle for the time specified in the idle timeout parameter.

To configure the idle session timeout and the associated action by using the command line


```
set subscriber param [-idleTTL <positive_integer>] [-idleAction <
    idleAction>]
<!--NeedCopy-->
```

Examples:

```
set subscriber param -idleTTL 3600 -idleAction ccrTerminate

set subscriber param -idleTTL 3600 -idleAction ccrUpdate

set subscriber param -idleTTL 3600 -idleAction delete
<!--NeedCopy-->
```

To disable the idle session timeout, set the idle timeout to zero.

set subscriber param –idleTTL 0

To configure the idle session timeout and the associated action by using the configuration utility

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. In the details pane, under **Settings**, click **Configure Subscriber Parameters** and specify an **Idle Time** and **Idle Action**.

Subscriber session event logging

If you enable subscriber logging, you can track the RADIUS and Gx control plane messages specific to a subscriber, and use the historical data to analyze subscriber activities. Some of the key attributes are MSISDN and time stamp. The following attributes are also logged:

- Session Event (Install, Update, Delete, Error)
- Gx Message Type (CCR-I, CCR-U, CCR-T, RAR)
- Radius Message Type (Start, Stop)
- Subscriber IP
- SubscriberID Type (MSISDN(E164), IMSI)
- SubscriberID value

By using these logs, you can track users by IP address and, if available, MSISDN.

You can enable subscriber session logging to a local or remote syslog or nslog server. The following example shows how to enable subscriber logging to a remote syslog server.


```
1      > add syslogAction sysact1 192.0.2.0 -loglevel EMERGENCY ALERT
          CRITICAL ERROR WARNING NOTICE INFORMATIONAL -subscriberlog
          enabled
2  <!--NeedCopy-->
```

From these logs, you can learn about any activity related to a user, such as the time when a session was updated, deleted, or created (installed). Additionally, error messages are also logged.

Examples:

1. The following log entries are examples of RADIUSandGx session creation, session update, and session deletion.

   > 09/30/2015:16:29:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 147 0 : Session Install, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:16:30:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 148 0 : Session Update, GX MsgType: CCR-U, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:17:27:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 185 0 : Session Delete, GX MsgType: CCR-T, RADIUS MsgType: Stop, IP: 100.10.1.1, ID: E164 - 30000000001

2. The following log entries are examples of failure messages, such as when a subscriber is not found on the PCRF server and when the appliance cannot connect to the PCRF server.

   > 09/30/2015:16:44:15 GMT Error 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 169 0 : Failure Reason: PCRF failure response, GX MsgType: CCR-I, IP: 100.10.1.1
   >
   > Sep 30 13:03:01 09/30/2015:16:49:08 GMT 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 176 0 : Failure Reason: Unable to connect to PCRF, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001#000#000#000#000#000#000#000#000#000#000#000#000#00

Subscriber aware LSN session termination

In earlier releases, if a subscriber session is deleted when a RADIUS Accounting STOP or a PCRF-RAR message is received, or as a result of any other event, such as TTL expiry or flush, the corresponding LSN sessions of the subscriber are removed only after the configured LSN timeout period. LSN sessions that are kept open until this timeout expires continue to consume resources on the appliance.

From release 11.1, a new parameter (subscrSessionRemoval) is added. If this parameter is enabled, and the subscriber information is deleted from the subscriber database, LSN sessions corresponding to that subscriber are also removed. If this parameter is disabled, the subscriber sessions are timed out as specified by the LSN timeout settings.

To configure subscriber aware LSN session termination by using the CLI

At the command prompt, type:

```
set lsn parameter -subscrSessionRemoval (          DISABLED )
ENABLED
```

```
1      > set lsn parameter -subscrSessionRemoval ENABLED
2      Done
3      > sh lsn parameter
4                   LSN Global Configuration:
5
6                   Active Memory Usage: 0 MBytes
7                   Configured Memory Limit: 0 MBytes
8                   Maximum Memory Usage Limit: 912 MBytes
9                   Session synchronization: ENABLED
10                  Subscriber aware session removal: ENABLED
11 <!--NeedCopy-->
```

To configure subscriber aware LSN session termination by using the GUI

1. Navigate to **System** > **Large Scale NAT**.
2. In **Getting started**, click **Set LSN Parameter**.
3. Set the **Subscriber Aware Session Removal parameter**.

Troubleshooting

If your deployment is not working as expected, use the following commands to troubleshoot:

- show subscriber gxinterface
  This command's output can include the following error messages (shown here with suggested responses):
  - Gx Interface Not Configured-Use set subscriber param command to configure the correct interface type.
  - PCRF not configured-Configure a Diameter vServer or Service on GxInterface-Use the set subscriber gx interface command to assign a Diameter virtual server or service to this interface.
  - PCRF is not ready-Check corresponding vserver/service for more details-Use the show LB vserver or show service command to check the state of the service.
  - Citrix ADC is waiting for CEA from PCRF-Capability negotiation between the PCRF and Citrix ADC might be failing. This could be an intermittent state. If it persists, check the DIAMETER

settings on your PCRF server.
  - Memory is not configured to store subscriber sessions. Please use ‘set extendedmemoryparam -memlimit <>’-Use the set extendedmemoryparam command to configure extended memory.
- show subscriber radiusinterface
  If “Not Configured” is the output of this command, use the set subscriber radiusinterface command to specify a RADIUSListener service.

If subscriber logging is enabled, you can get more detailed information from the log files.

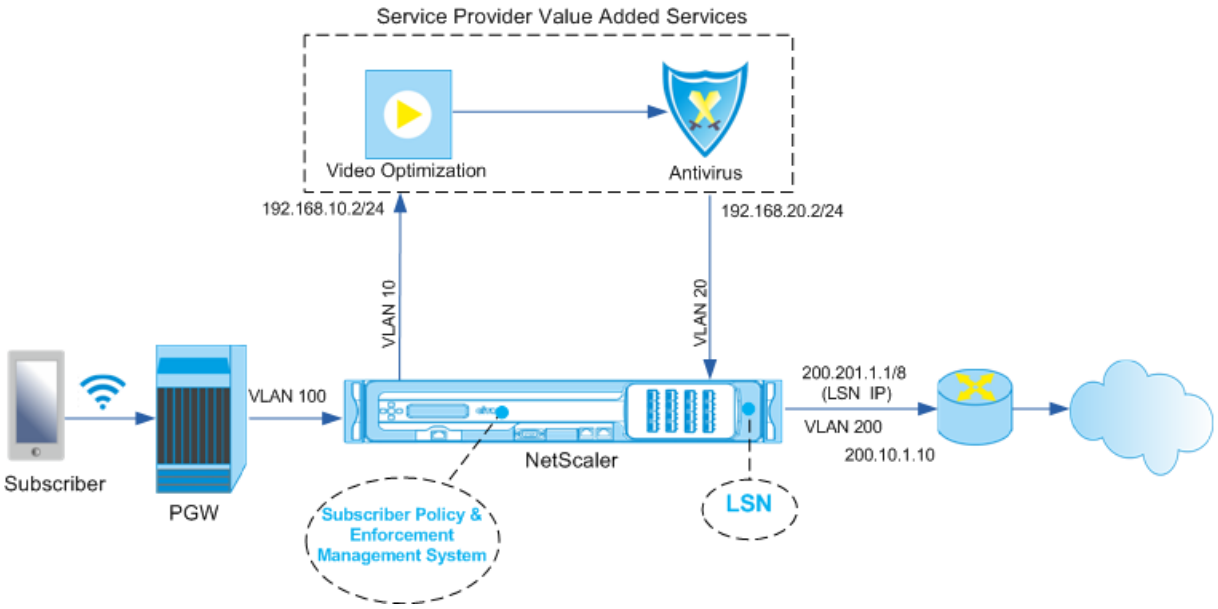
Subscriber aware traffic steering

September 14, 2021

Traffic steering directs subscriber traffic from one point to another. When a subscriber connects to the network, the packet gateway associates an IP address with the subscriber and forwards the data packet to the Citrix ADC appliance. The appliance communicates with the PCRF server over the Gx interface to get the policy information. Depending on the policy information, the appliance performs one of the following actions:

- Forward the data packet to another set of services (as shown in the following illustration).
- Drop the packet.
- Perform only Large Scale NAT (LSN), if LSN is configured on the appliance.

The values shown in the following figure are configured in the CLI procedure that follows the figure. A content switching virtual server on the Citrix ADC appliance directs requests to the value added services or skips them, depending on the defined rule, and then sends the packet out to the Internet after performing LSN.



To configure traffic steering for the above deployment by using the CLI

Add the appliance's subnet IP (SNIP) addresses.

Example:

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 100.100.100.1 255.0.0.0 -type snip

add ns ip 200.200.200.1 255.0.0.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
<!--NeedCopy-->
```

Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet IP addresses.

Example:

```
add vlan 10

add vlan 20

add vlan 100
```



```
6
7  add vlan 200
8
9  bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0
10
11 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0
12
13 bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0
14
15 bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.1.1.1 255.0.0.0
16 <!--NeedCopy-->
```

Specify the VLAN on which the subscriber traffic arrives on the appliance. Specify the service path AVP that tells the appliance where to look for the service path name within the subscriber session. For primary PCEF functionality, specify the interfaceType as RadiusAndGx.

Example:

```
1 set ns param -servicePathIngressVLAN 100
2
3 set subscriber gxinterface -servicepathAVP 1001 1005 -
     servicepathVendorid 10415
4
5 set subscriber param -interfaceType RadiusAndGx
6 <!--NeedCopy-->
```

Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Then, specify the PCRF realm and subscriber Gx interface parameters. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface.

Example:

```
1  add service sd1 10.102.232.200 DIAMETER 3868
2
3  add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
      persistAVPno 263
4
5  bind lb vserver vdiam sd1
6
7  set ns diameter -identity netscaler.sc1.net -realm pcrf1.net
8
9  set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
      holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120
10
11 add service srad1 10.102.232.236 RADIUSListener 1813
```



```
12
13 set subscriber radiusInterface -listeningService srad1
14 <!--NeedCopy-->
```

Add service functions to associate a VAS with an ingress VLAN. Add a service path to define the chain, that is, specify the VAS that the packet must be sent to and the order in which it must go to that VAS. The service path name is usually sent by the PCRF. However, the service path of the default subscriber profile (*) applies if any of the following is true:

- PCRF does not have the subscriber information.
- The subscriber information does not include this AVP.
- The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

The service path AVP that contains this name must already be configured as part of the global configuration. Bind the service function to the service path. The service index specifies the order in which the VAS is added to the chain. The highest number (255) indicates the beginning of the chain.

Example:

```
1 add ns servicefunction SF1 -ingressVLAN 20
2
3 add ns servicepath pol1
4
5 bind ns servicepath pol1 -servicefunction SF1 -index 255
6
7 add subscriber profile * -subscriberrules default_path
8 <!--NeedCopy-->
```

Add the LSN configuration. That is, define the NAT pool and identify the clients for which the appliance must perform LSN.

```
1 add lsn pool pool1
2
3 bind lsn pool pool1 200.201.1.1
4
5 add lsn client client1
6
7 bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0
8
9 add lsn group group1 -clientname client1
10
11 bind lsn group group1 -poolname pool1
12 <!--NeedCopy-->
```


The appliance performs LSN by default. To override LSN, you must create a net profile with the overrideLsn parameter enabled, and bind this profile to all the load balancing virtual servers that are configured for value added services (VASs).

Example:

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

Configure the VAS on the appliance. This includes creating the services and virtual servers and then binding the services to the virtual servers.

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vint sint
<!--NeedCopy-->
```

Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The traffic arrives at the CS virtual server and is then redirected to the appropriate load balancing virtual server. Define expressions that associate a virtual server with a service function.

Example:

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule  SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
    SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

bind cs vserver cs1 -policyName cspol1 -priority 110

```



```
bind cs vserver cs1 -lbvserver vint
<!--NeedCopy-->
```

To configure traffic steering on the appliance by using the GUI

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining** > **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:
   - Interface Type: Specify **RadiusAndGx**.
   - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
   - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management** > **Service Chaining** > **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System** > **Network** > **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System** > **Network** > **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System** > **Network** > **Settings** > **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

To configure service chaining on the appliance by using the GUI

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining** > **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:

- Interface Type: Specify **RadiusAndGx**.
- Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
- Specify the RADIUS interface parameters.

5. Navigate to **Traffic Management** > **Service Chaining** > **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System** > **Network** > **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System** > **Network** > **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System** > **Network** > **Settings** > **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

Subscriber aware service chaining

September 14, 2021

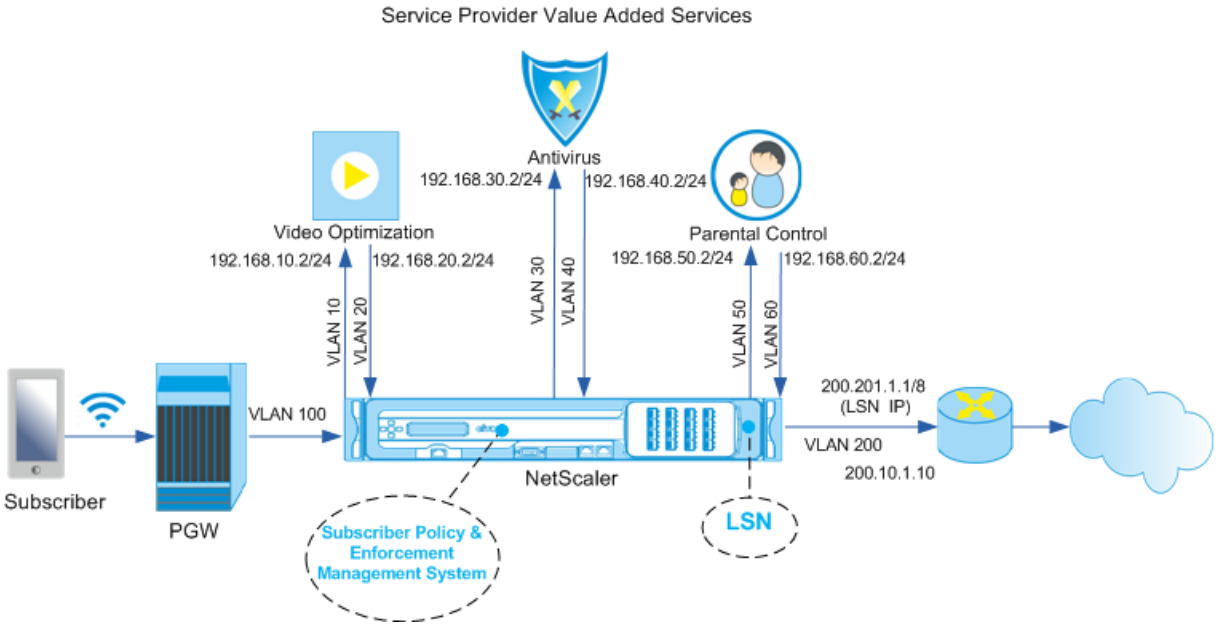
With the huge increase in the data traffic passing through telco networks, it is no longer feasible for service providers to steer all the traffic through all the value added services (VAS). A service provider should be able to optimize usage of VAS and intelligently steer traffic to improve the user experience. For example, video optimization is not required for traffic that does not include a video. Moreover, if a subscriber is connected to a 4G network, content can be streamed in high definition (HD), and video optimization might not be needed. However, video optimization improves the experience for a user in a 3G network. Similarly, caching provides a faster and better user experience and can be enabled depending on the subscriber plan. Another example of VAS is parental control. If parents provide a mobile handset to a minor child, they would like some kind of control over the websites that their child visits.

To do the above and more, service providers must be able to provide value-added services on a per-subscriber basis. In other words, entities in the service provider network must be capable of extracting the subscriber information and intelligently steering the packet on the basis of this information.

Service chaining determines the set of services through which the traffic from a subscriber must pass before going to the Internet. Instead of sending all the traffic to all the services, the Citrix ADC intelligently routes all requests from a subscriber to a specific set of services on the basis of the policy

defined for that subscriber.

The following figure shows the entities involved in service chaining. The values shown are configured in the procedure that follows the figure. A content switching virtual server on the Citrix ADC appliance directs requests to the value added services or skips them, depending on the defined rule, and then sends the packet out to the Internet after performing LSN.



To configure service chaining for the above deployment by using the CLI

Add the appliance's subnet IP (SNIP) addresses.

Example:

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.30.1 255.255.255.0 -type snip

add ns ip 192.168.40.1 255.255.255.0 -type snip

add ns ip 192.168.50.1 255.255.255.0 -type snip

add ns ip 192.168.60.1 255.255.255.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
```



```
<!--NeedCopy-->
```

Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet IP addresses. Add an ingress and an egress VLAN for each VAS.

Example:

```
add vlan 10

add vlan 20

add vlan 30

add vlan 40

add vlan 50

add vlan 60

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.30.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.40.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.50.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.60.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.201.1.1 255.0.0.0
<!--NeedCopy-->
```

Specify the VLAN on which the subscriber traffic arrives on the appliance. Specify the service path AVP that tells the appliance where to look for the service path name within the subscriber session. For primary PCEF functionality, specify the interfaceType as RadiusAndGx.

Example:


```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
    servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Then, specify the PCRF realm and subscriber Gx interface parameters. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface.

Example:

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

Add service functions to associate a VAS with an ingress VLAN. Add a service path to define the chain, that is, specify the VAS that the packet must be sent to and the order in which it must go to that VAS. The service path name is usually sent by the PCRF. However, the service path of the default subscriber profile (*) applies if any of the following is true:

- PCRF does not have the subscriber information.
- The subscriber information does not include this AVP.
- The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

The service path AVP that contains this name must be configured as part of the global configuration earlier. Bind the service function to the service path. The service index specifies the order in which the VAS is added to the chain. The highest number (255) indicates the beginning of the chain.

Example:

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicefunction SF2 -ingressVLAN 40

add ns servicefunction SF3 -ingressVLAN 60

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

bind ns servicepath pol1 -servicefunction SF2 -index 254

bind ns servicepath pol1 -servicefunction SF3 -index 253

add ns servicepath pol2

bind ns servicepath pol2 -servicefunction SF2 -index 255

add ns servicepath pol3

bind ns servicepath pol3 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

Add the LSN configuration. That is, define the NAT pool and identify the clients for which the appliance must perform LSN.

Example:

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
<!--NeedCopy-->
```

The appliance performs LSN by default. To override LSN, you must create a net profile with over-

rideLsn parameter enabled and bind this profile to all the load balancing virtual servers that are configured for value added services (VASs).

Example:

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

Configure the VAS on the appliance. This includes creating the services and virtual servers and then binding the services to the virtual servers.

Example:

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service vas2 192.168.30.2 ANY 80 -usip YES

add service vas3 192.168.50.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vs2 ANY -m MAC -l2Conn ON

add lb vserver vs3 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vs2 vas2

bind lb vserver vs3 vas3

bind lb vserver vint sint
<!--NeedCopy-->
```

Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The traffic arrives at the CS virtual server and is then redirected to the appropriate load balancing virtual server. Define expressions that associate a virtual server with a service function.

Example:

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csact2 -targetLBVserver vs2

add cs action csact3 -targetLBVserver vs3

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
   SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

add cs policy cspol2 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF2") &&
   SYS.VSERVER("vs2").STATE.EQ(UP)" -action csact2

add cs policy cspol3 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF3") &&
   SYS.VSERVER("vs3").STATE.EQ(UP)" -action csact3

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -policyName cspol2 -priority 120

bind cs vserver cs1 -policyName cspol3 -priority 130

bind cs vserver cs1 -lbvserver vint
<!--NeedCopy-->
```

To configure service chaining on the appliance by using the GUI

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to **System** > **Network** > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining** > **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:
   - Interface Type: Specify **RadiusAndGx**.
   - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
   - Specify the RADIUS interface parameters.

5. Navigate to **Traffic Management** > **Service Chaining** > **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System** > **Network** > **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System** > **Network** > **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System** > **Network** > **Settings** > **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

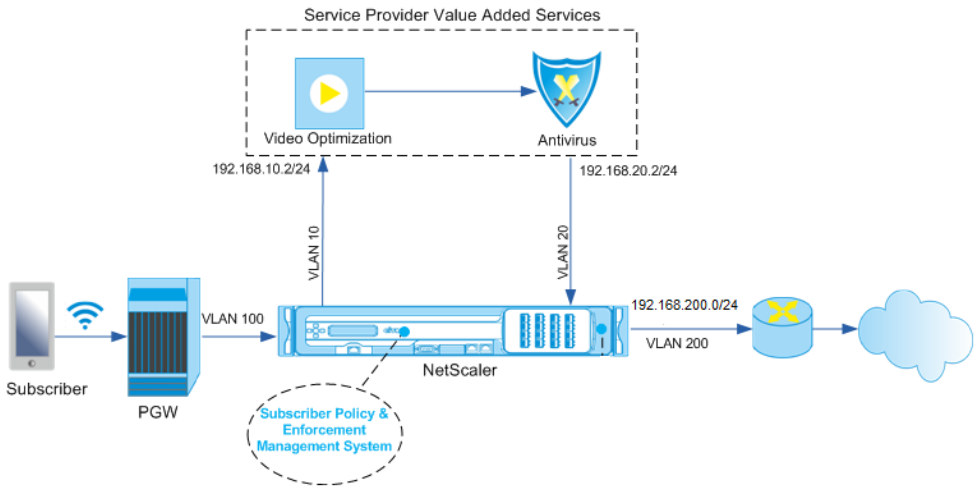
Subscriber aware traffic steering with TCP optimization

September 14, 2021

Traffic steering directs subscriber traffic from one point to another. When a subscriber connects to the network, the packet gateway associates an IP address with the subscriber and forwards the data packet to the Citrix ADC appliance. The appliance communicates with the PCRF server over the Gx interface to get the subscriber policy information. Depending on the policy information, the appliance performs one of the following actions:

- Forward the data packet to another set of services (as shown in the following illustration).
- Perform only TCP optimization.

The values shown in the following figure are configured in the CLI procedure that follows the figure. A content switching virtual server on the Citrix ADC appliance directs requests to the value added services or skips them and performs TCP optimization, depending on the defined rule, and then sends the packet out to the Internet.



> **Note**
>
> Support for the configuration shown below was introduced in release 11.1 build 50.10.

To configure traffic steering for the above deployment by using the CLI:

1. Add the appliance's subnet IP (SNIP) addresses.

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.100.1 255.255.255.0 -type snip

add ns ip 192.168.200.1 255.255.255.0 -type snip

add ns ip 10.102.232.236 255.255.255.0  - type snip
<!--NeedCopy-->
```

2. Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet IP addresses.

```
add vlan 10

add vlan 20

add vlan 100

add vlan 200

add vlan 102
```



```
10
11 bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1
      255.255.255.0
12
13 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1
      255.255.255.0
14
15 bind vlan 100 -ifnum 1/2 -tagged -IPAddress 192.168.100.1
      255.255.255.0
16
17 bind vlan 200 -ifnum 1/2 -tagged -IPAddress 192.168.200.1
      255.255.255.0
18
19 bind vlan 102  - ifnum 1/1  - tagged  - IPAddress 10.102.232.236
      255.255.255.0
20 <!--NeedCopy-->
```

3. Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Specify the PCRF realm and values for the subscriber Gx interface parameters. Also specify the service path AVP that indicates where the appliance can find the service path name within the subscriber session. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface, and specify the interface type as “RadiusAndGx”.

```
1  add service sd1 10.102.232.200 DIAMETER 3868
2
3  add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER
      -persistAVPno 263
4
5  bind lb vserver vdiam sd1
6
7  set ns diameter -identity netscaler.sc1.net -realm pcrf1.net
8
9  set extendedmemoryparam -memLimit 2558
10
11 set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net
12
13 set subscriber gxinterface -servicepathAVP 1001 1005 -
      servicepathVendorid 10415
14
15 add service srad1 10.102.232.236 RADIUSListener 1813
16
17 set subscriber radiusInterface -listeningService srad1
18
19 set subscriber param -interfaceType RadiusAndGx
```



```
20 <!--NeedCopy-->
```

4. Specify a default subscriber profile (*) to be applied if any of the following is true:

   - PCRF does not have the subscriber information.
   - The subscriber information does not include the service path AVP.
   - The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

```
1 add subscriber profile * -subscriberrules default_path
2 <!--NeedCopy-->
```

5. Create TCP profiles for the VAS and TCP optimization path, respectively. Traffic steered to VAS will not undergo any TCP optimization before or after leaving the VAS. Therefore, the TCP mode of the VAS profile should be set to TRANSPARENT while the TCP mode of the TCPOpt profile should be set to ENDPOINT.

   add ns tcpProfile VAS –tcpMode TRANSPARENT

   add ns tcpProfile TCPOpt -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

6. Configure load balancing for the VAS servers. Create a non-addressable virtual server of type TCP. Create TCP services with the IP addresses of the VAS servers, and bind the services to the virtual server. The virtual server and services will use the transparent TCP profile created for the VAS path:

```
1  add service vas1 192.168.10.2 TCP * -usip YES -useproxyport NO -
       TCPB NO -tcpProfileName VAS
2
3  add service vas2 192.168.10.3 TCP * -usip YES -useproxyport NO -
       TCPB NO -tcpProfileName VAS
4
5  add lb vserver vs1 TCP -m MAC -l2Conn ON  - tcpProfileName VAS
6
7  bind lb vserver vs1 vas1
8
9  bind lb vserver vs1 vas2
10 <!--NeedCopy-->
```

7. Add a load balancing virtual server to capture VAS egress traffic. This vserver will monitor the VAS egress VLAN and will use the transparent TCP profile:


```
add lb vserver vsint TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(20)"
    -Listenpriority 30 -l2Conn ON -tcpProfileName VAS
<!--NeedCopy-->
```

8. Add a TCP optimization virtual server that listens for any traffic in the wireless-side VLAN and uses the endpoint TCP profile created for the TCP optimization path:

```
add lb vserver vs-TcpOpt TCP * * -Listenpolicy "client.vlan.id.eq
    (100)" -Listenpriority 20 -l2Conn ON -tcpProfileName TCPOpt
<!--NeedCopy-->
```

9. Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The CS virtual server receives the traffic and redirects it to the appropriate load balancing virtual server according to defined CS policies. Create a CS TCP virtual server that listens for any traffic in the wireless-side VLAN with highest priority and uses the endpoint TCP profile. Create a CS policy that evaluates to TRUE when “vas” is the subscriber rule, and specify a CS action that steers traffic to VAS. Make the TCP optimization virtual server the default LB vserver. Any subscriber traffic with a rule other than “vas” will go through the default LB vserver.

```
add cs vserver cs1 TCP * * -Listenpolicy "client.vlan.id.eq(100)"
    -Listenpriority 10 -l2Conn ON -tcpProfileName TCPOpt

add cs action csact1 -targetLBVserver vs1

add cs policy cspol1 -rule SUBSCRIBER.RULE_ACTIVE(\"vas\") && SYS
    .VSERVER(\"vs1\").STATE.EQ(UP)" -action csact1

bind cs vserver cs1 -policyName cspol1

bind cs vserver cs1 -lbvserver vs-TcpOpt
<!--NeedCopy-->
```

10. Add static or policy based routes to the internet. Dynamic routing is also supported in this configuration. The following example uses policy based routes:

```
add ns pbr pbr-vlan100-to-vlan200 ALLOW -nextHop 192.168.200.10 -
    vlan 100 -priority 10

add ns pbr pbr-vlan20-to-vlan200 ALLOW -nextHop 192.168.200.10 -
    vlan 20 -priority 11

apply ns pbrs
<!--NeedCopy-->
```


Note

The CS policies can contain IP addresses and port numbers in addition to the subscriber expressions—for example, SUB-SCRIBER.RULE_ACTIVE(”vas”) && && (CLIENT.TCP.DSTPORT.EQ(80)	CLIENT.TCP.DSTPORT.EQ(443). They can also contain HTTP based expressions—for example, HTTP.REQ.HOSTNAME.DOMAIN.EQ(”somedor In this case, replace TCP entities (vserver, service, etc.) with HTTP. The TCP profile configuration remains the same.

-
- Add IPv6 configuration (addresses, routes, PBRs) to support IPv6 subscribers. Happy Eyeballs client applications will work smoothly for both VAS and TCP optimization paths.
- Add VLANs, IP addresses, PBRs and LB virtual servers in front of VAS (vs1, vs2, etc.) to support multiple subscriber flows. Modify the listen policies of CS vserver “cs1” and LB vserver “vsint” to include the additional VLANs.

Policy based TCP profile selection

September 14, 2021

You can configure the Citrix ADC appliance to perform TCP optimization based on subscriber attributes. For example, the appliance can select different TCP profiles at run time, based on the network to which the user equipment (UE) is connected. As a result, you can improve a mobile user’s experience by setting some parameters in the TCP profiles and then using policies to select the appropriate profile.

Create separate TCP profiles for subscribers connecting through a 4G network and for users connecting through any other network. Define a policy rule that is selected on the basis of a subscriber parameter, such as RAT-type. In the following examples, if RAT-Type is EUTRAN, a TCP profile that supports a faster connection is selected (Example 1). For all other RAT-Type values, a different TCP profile is selected (Example 2).

Note

> The RAT-Type AVP (AVP code 1032) is of type Enumerated and is used to identify the radio access technology that is serving the UE.
>
> The value “1004” indicates that the RAT is EUTRAN. (RFC 29.212).

Example1:

```
add ns tcpProfile tcp2 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 1000000 -flavor BIC
     - dynamicReceiveBuffering DISABLED -sendBuffsize 1000000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 500 -maxburst 15

add appqoe action appact2 -priority HIGH -tcpprofile tcp2

add appqoe policy apppol2 -rule "SUBSCRIBER.AVP(1032).VALUE.
    GET_UNSIGNED32(0, BIG_ENDIAN).EQ(1004)" -action appact2

bind cs vserver <name>  -policyname apppol2 -priority 20 -type request
<!--NeedCopy-->
```

Example2:

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 150000 -flavor BIC
    - dynamicReceiveBuffering DISABLED -sendBuffsize 150000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 200 -maxburst 15

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "SUBSCRIBER.AVP(1032).VALUE.
    GET_UNSIGNED32(0, BIG_ENDIAN).NE(1004)" -action appact1

bind cs vserver <name>  -policyname apppol1 -priority 10 -type request
<!--NeedCopy-->
```

Load Balance Control-Plane Traffic that is based on Diameter, SIP, and SMPP Protocols

September 14, 2021

With the increase in control-plane traffic, the servers can become a bottleneck because the traffic is not optimally distributed among the servers. Therefore, messages must be load balanced. The Citrix ADC appliance supports Diameter, SIP, and SMPP load balancing.

SIP

Citrix ADC enables you to load balance SIP messages over UDP or over TCP (including TLS) to a group of proxy servers. Citrix ADC also provides Call-ID based persistence and Call-ID hash load balancing method using which you direct packets for a particular SIP session to the same load balanced SIP server.
The Citrix ADC default expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions are intended to be used in policies for SIP protocol that operates on a request/response basis. These expressions can be used in content switching, rate limiting, responder, and rewrite policies.

For more information, see Load Balancing a Group of SIP Servers.

SMPP

Millions of short messages are exchanged daily between individuals and value-added service providers, such as banks, advertisers, and directory services, by using the short message peer to peer (SMPP) protocol. Often, message delivery is delayed because servers are overloaded and traffic is not optimally distributed among the servers.
The Citrix ADC appliance provides optimal distribution of messages across your servers, preventing poor performance and outages. The Citrix ADC appliance:

- Load balances messages originating from the server and from the client
- Monitors the health of the message centers
- Provides content switching support for the message centers
- Handles concatenated messages

Limitation: Message IDs, from the message center, longer than 59 bytes are not supported. If the message ID length returned by the message center is more than 59 bytes, ancillary operations fail and the Citrix ADC appliance responds with an error message.

For more information, see SMPP Load Balancing

Diameter

Diameter is a base protocol with more than 50 protocols (also called applications) built over it. Therefore, the diameter traffic generated in a Telco network is high. To optimally maintain this diameter traffic, the Citrix ADC appliance performs load balancing, content switching, and acts as a relay agent. Additionally, the appliance offers rewrite and responder functionality. The appliance supports rate limiting of Diameter messages.

For more information, see Configuring Diameter Load Balancing.

Provide DNS Infrastructure/Traffic Services, such as, Load Balancing, Caching, and Logging for Telecom Service Providers

September 14, 2021

Telecom service providers can configure the Citrix ADC appliance to function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by default on the Citrix ADC appliance. This enables the Citrix ADC appliance to provide quick responses for repeated translations and hence enhances the customer experience and also saves the bandwidth. The caches responses from DNS name servers. When the appliance receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the Citrix ADC appliance returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the Citrix ADC appliance. The Citrix ADC appliance then returns the address to the client.

For requests for a domain that has been cached earlier, the Citrix ADC appliance serves the Address record of the domain from the cache without querying the configured DNS server and hence saves the bandwidth.

From 11.0 release onwards, Citrix ADC also logs the DNS requests that it receives and also the responses that it sends to the client. Telecom service providers can use this log to:

- Audit the DNS responses to the client
- Audit DNS clients
- Detect and prevent DNS attacks
- Troubleshooting

For more information, see Domain Name System.

Provide Subscriber Load Distribution Using GSLB Across Core-Networks of a Telecom Service Provider

September 14, 2021

Scalability, high availability and performance are critical to service provider deployments. While many service providers deploy there infrastructure at a single location or multiple location, these deployments are subject to a number of inherent limitations, such as:

- If the site loses connectivity to all or part of the public Internet, it will be inaccessible to users and customers, which can have significant impact on the business.

- Users accessing the site from geographically distant locations may experience large and highly variable delays, which are exacerbated by the large number of round trips that HTTP requires to transfer content.

Citrix ADC appliance's Global Server Load Balancing (GSLB) overcomes these problems by distributing traffic among sites deployed in multiple geographic locations. By serving content from many different points in the Internet, GSLB alleviates the impact of network bandwidth bottlenecks and provides robustness in case of network failures at a particular site. Users can be automatically directed to the nearest or least loaded site at the time of the request, minimizing the likelihood of long download delays and/or service disruptions.

You can use Citrix ADC appliance's global server load balancing for:

- Disaster recovery or high availability by configuring an Active-standby data center setup that consists of an active and a standby data center. When a failover occurs as a result of a disaster event, the standby data center becomes operational.
- High availability and speed by configuring an active-active data center setup that consists of multiple active data centers. Client requests are load balanced across active data centers.
- Directing client requests to the data center that is closest in geographical distance or network distance by configuring a proximity setup.
- Full-DNS resolutions, GSLB processes DNS queries of the A, AAAA and CNAME types, and the DNS function option can process DNS queries of all other types, such as MX and PTR. Also, if the recursive resolution is enabled, the appliance will forward DNS queries for domain names that are not configured on the Citrix ADC appliance.

For more information, see Global Server Load Balancing.

Bandwidth Utilization Using Cache Redirection Functionality

September 14, 2021

The volume of web traffic on the Internet is enormous and a large percentage of that traffic is redundant. Multiple clients ask web servers for the same content repeatedly leading to inefficient use of bandwidth. To relieve the origin web server of processing each request, Internet Service Providers (ISPs) can use the cache redirection feature of Citrix ADC appliance and serve the content from a cache server instead of from the origin server. The Citrix ADC appliance analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers. Cache redirection feature of Citrix ADC is policy-based, and by default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

For more information, see Cache Redirection.

Citrix ADC TCP Optimization

September 14, 2021

The Citrix ADC appliance provides advanced TCP tuning and optimization techniques and capabilities that are well suited to modern 3.5 and 4G networks, improving user experience and perceived download speeds significantly.

This section focuses on detailed instructions relevant to:

- Choosing and inserting an appropriate Citrix ADC T1000 Series model in a mobile network for TCP optimization
- Full configuration instructions related to not only TCP optimization but also for appropriate Layer-2 and Layer-3 configuration of the T1 device

The section includes the following topics:

- Getting Started
- Managment Network
- Licensing
- High Availability
- Gi-LAN Integration
- TCP Optimization Configuration
- Optimizing TCP Performance Using TCP NILE
- Analytics and Reporting
- Real-time Statistics
- SNMP
- Technical Recipes
- Troubleshooting Guidelines
- Frequently Asked Questions

Getting Started

September 14, 2021

Hardware

Citrix provides a wide breath of Citrix ADC models that might be loosely based on two factors:

- Capacity, currently ranging from hundreds of Mbps for the low-end VPX appliance to 160Gbps for the high-end 25000 MPX series appliance
- Telco grade, with the availability of the T1000 series for Telco datacenters.

Your Citrix Sales or Support representative can help you select the appropriate hardware for your demo, trial, or production needs.

The remainder of this section uses a Citrix ADC T1200 as a reference hardware. Note that putting aside superficial differences related to number and notation of available interfaces (see * in note) or well-documented limitations of Citrix ADC VPX (see * in note) the instructions should apply mostly verbatim regardless of the Citrix ADC model selected.

> **Note**
>
> * For instance a the T1010 model only has 12x1GbE typically marked as 1/1-1/12 rather than the 10/x notation used in this document.
>
> ** A Citrix ADC VPX instance typically doesn't support LACP aggregation; it might also not support VLAN tagging.

Initial Setup

Through Serial Console

After a serial cable is connected, you can log on to the Citrix ADC appliance with the following credentials:

- Username: nsroot
- Password: nsroot

Once logged in, configure the basic details of the Citrix ADC appliance as shown in the screen capture below.

Example:

```
set ns config  - IPAddress <ip_addr> -netmask <netmask>

saveconfig

reboot -warm
<!--NeedCopy-->
```

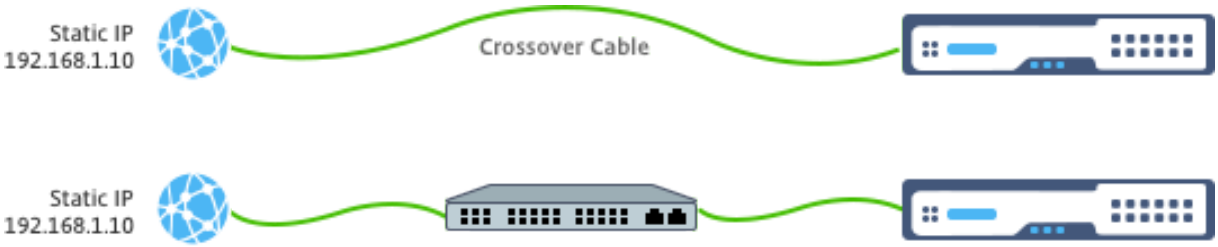
After you restart the appliance, you might use SSH for further configuration of the T1100 nodes.

Through LOM

Lights out Management (LOM) port on the front panel of the Citrix ADC appliance allows operator to remotely monitor and manage the appliance independently of the operating system. Operator can change the IP address, power cycle, and perform a code dump by connecting to the Citrix ADC appliance through the LOM port.

Default IP Address of LOM port is 192.168.1.3

Figure. Intial Configuration of LOM Module



Set a static IP on your laptop and plug it directly into the LOM interface with a crossover cable or into a switch in the same broadcast domain as the LOM interface.

For initial configuration, type the port's default address: http://192.168.1.3 in a web browser and change the LOM port's default IP address.

Refer to Configuration Guides for further details.

Software

The Citrix ADC TCP optimization for mobile networks is constantly evolving. The capabilities and tunings outlined in this document require a Citrix ADC Telco build. Here is an example showing the Citrix ADC Telco build.

Example:

```
1 show ver
2
3 NetScaler NS11.0: Build 64.957.nc, Date: Aug 26 2016, 02:00:23
4 <!--NeedCopy-->
```

If the T1000 has not shipped with the appropriate build revision, contact the Citrix ADC Customer Support.

> **Important**
>
> Both the appliances should have the same software image.

SSH Client

A Citrix ADC appliance can be configured by using either the CLI or the HTML5 GUI. However, this section provides only CLI-based instructions.

While the CLI is accessible through the Citrix ADC serial console, an SSH client is normally recommended to allow for remote Citrix ADC configuration.

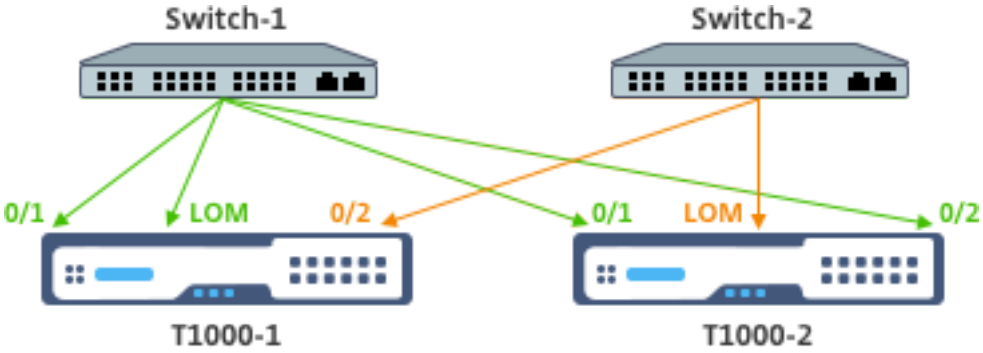
Management Network

September 14, 2021

Connectivity

Most Citrix ADC devices offer redundant 1GbE OAM ports, notated as 0/1 and 0/2. To provide for redundancy in case of a switch failure, you should connect the relevant ports to different upstream switches.

A high-level overview of the recommended connectivity is outlined in the following diagram:



After the Citrix ADC appliance is connected to the management network, subsequent configuration steps can be performed remotely using SSH or web connectivity to the CLI and GUI respectively.

Routing

The add route command may be used to configure any routes appropriate to the management network. The relevant gateway should be reachable on the NSIP subnet, as shown below.

Example:

```
add route <network>  <netmask>  <gateway>
<!--NeedCopy-->
```


Licensing

September 14, 2021

A valid license file should be installed on the Citrix ADC appliance. The license should support at least as many Gbps as the expected maximum Gi-LAN throughput.

License files should be copied through an SCP client to the /nsconfig/license of the appliance, as shown in the screen capture below.

EXample:

```
shell ls /nsconfig/license/

CNS_V3000_SERVER_PLT_Retail.lic ssl
<!--NeedCopy-->
```

Do a warm restart to apply the new license, as shown in the screen capture below.

EXample:

```
reboot -warm

Are you sure you want to restart NetScaler (Y/N)? [N]:y

Done
<!--NeedCopy-->
```

After the restart completes, verify that the license has been properly applied, by using the show license CLI.

In the example below a 3Gbps Premium license has been successfully installed.

Example:

```
> show license

            License status:

                                  Web Logging: YES

                                          ...

                               Model Number ID: 3000

                                  License Type: Premium License

```



```
Done

<!--NeedCopy-->
```

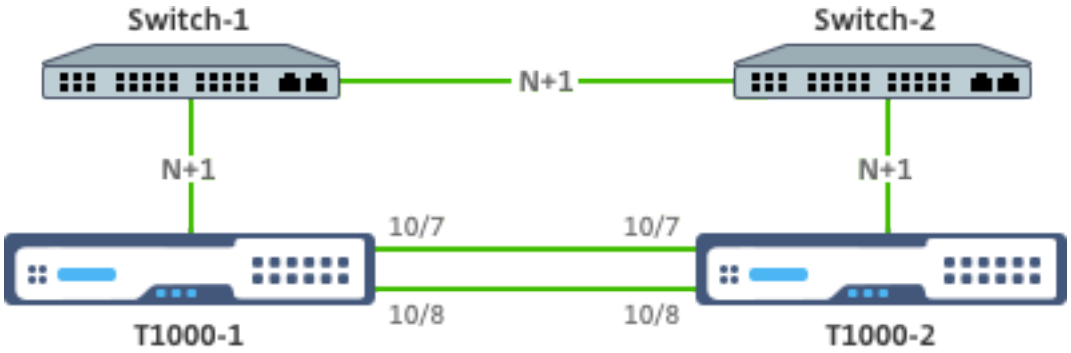
High Availability

September 14, 2021

High availability (HA) refers to an active-standby operational mode of a Citrix ADC device pair. Each device has its own dedicated management IP address. All other IP addresses are owned by the active device in the pair.

Connectivity

While there are multiple connectivity options for a Citrix ADC HA pair, the most recommended one is depicted in the following diagram:



In the above diagram, the N+1 red links between each T1000 and the respective switch imply N+1 redundancy - as explained in Connectivity. For instance, considering a 45 Gbps Gi-LAN N=5 is an appropriate value, with 6x10GbE LACP channels between each switch and the respective T1000 as well as between the two switches.

An extra pair of links is recommended between the Citrix ADC pair, to provide for HA communication isolation from the OAM network.

Gi-LAN Integration

November 22, 2021

Typically, a Citrix ADC appliance is inserted as a separate L3 inline node in the Gi-LAN, similarly to an L3 router.

Figure: A simple depiction of a Gi-LAN

Connectivity

A physical Citrix ADC connectivity to upstream switches is recommended to provide for sufficient redundancy. For example, assuming that a Citrix ADC appliance is inserted in a Gi-LAN that is handling a total (uplink+downlink) of 24Gbps, connectivity with 4x10GbE or more interfaces is recommended. This effectively provides for N+1 redundancy in case of a link failure.

The relevant ports on the upstream switch should be configured for LACP port aggregation. The relevant configuration on Citrix ADC is outlined below:

Connectivity Configuration:

```
set interface 10/1 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/2 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/3 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/4 -tagall ON -lacpMode ACTIVE -lacpKey 1
<!--NeedCopy-->
```

You can verify the appropriate functionality of LACP using the “show interface” command:

show interface:

```
sh interface LA/1

    1)      Interface LA/1 (802.3ad Link Aggregate) #39

             flags=0x4100c020 <ENABLED, UP, AGGREGATE, UP, HAMON, 802.1
                q>

             MTU=1500, native vlan=1, MAC=02:e0:ed:33:88:b0, uptime 340
                h11m56s

             Requested: media NONE, speed AUTO, duplex NONE, fctl NONE,
```



```
                throughput 0

                Actual: throughput 4000

                LLDP Mode: NONE,

                RX: Pkts(918446) Bytes(110087414) Errs(0) Drops(795989)
                    Stalls(0)

                TX: Pkts(124113) Bytes(15255532) Errs(0) Drops(0) Stalls
                    (0)

                NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
                    Muted(0)

                Bandwidth thresholds are not set.

Disable the remaining unused interfaces and turn off the monitor.

set interface 10/5  -haMonitor OFF
<!--NeedCopy-->
```

Command:

```
set interface 10/24  -haMonitor OFF

disable interface 10/5

disable interface 10/24
<!--NeedCopy-->
```

Configuration of physical interfaces is not shared across the two Citrix ADC units. Hence, the above commands must be run across both Citrix ADC nodes in case of an HA pair deployment.

HA Configuration

All other configuration parameters are shared between the Citrix ADC nodes of an HA pair. Hence, HA sync should be enabled prior to any other configuration commands being run. Basic HA configuration involves the following steps:

1. Using the exact same Citrix ADC hardware, software, and license: HA pairs are not supported between different models (i.e. a T1100 and an MPX21550) or same models with different firmware levels. Refer to the appropriate instructions on upgrading an existing HA pair - Upgrading to Release 11.1.

2. Establishing the HA pair.

Example:

```
netscaler-1> add HA node 1 <netscaler-2-NSIP>

netscaler-2> add HA node 1 <netscaler-1-NSIP>
<!--NeedCopy-->
```

3. Verify the HA pair establishment running the following command in either node; both nodes should be visible, one of them as Primary (active), the other as a Secondary (standby).

Example:

```
show HA node
<!--NeedCopy-->
```

4. Enable failsafe mode and maxFlips. This ensures that in case of a route monitor failure on both nodes at least one node remains active without active/standby status constantly switching.

Example:

```
set HA node  - failsafe ON

set HA node -maxFlips 3 -maxFlipTime 1200
<!--NeedCopy-->
```

5. Finally, enable HA sync to occur over the dedicated intra-Citrix ADC ports rather than the OAM network.

Example:

```
add vlan 4080 -aliasName syncVlan

set HA node -syncvlan 4080
<!--NeedCopy-->
```

> **Note**
>
> The VLAN 4080 in the commands in the above example shouldn't be taken literally. Any unused VLAN-ID might be reserved.

VLAN Configuration

After the physical interfaces have been appropriately configured, you might configure the appropriate Gi-LAN VLANs. For instance, consider a rather simple Gi-LAN environment with an ingress/egress VLAN pair with 100/101 VLAN-identifier respectively.

The following commands configure the relevant VLANs on top of the LACP channel created in the prior step.

```
add vlan 100
add vlan 101
bind vlan 100 -ifnum LA/1 -tagged
bind vlan 101 -ifnum LA/1 -tagged
<!--NeedCopy-->
```

IPv4 Configuration

Typically, a Citrix ADC appliance requires one SNIP per VLAN. The example below assumes that the networks outlined in the Gi-LAN integration diagram, given in the begining of this page, have a /24 subnet mask:

```
add ns ip 192.168.1.254 255.255.255.0 -vserver DISABLED -mgmtAccess
    DISABLED
add ns ip 192.168.2.254 255.255.255.0 -vserver DISABLED -mgmtAccess
    DISABLED
<!--NeedCopy-->
```

After the SNIPs have been configured they should be associated with the appropriate VLAN:

```
bind vlan 100 -IPAddress 192.168.1.254 255.255.255.0
bind vlan 101 -IPAddress 192.168.2.254 255.255.255.0
<!--NeedCopy-->
```

IPv4 Static Routing

The example outlined in the Management Network section calls for only a couple of static routing rules:

- A 10.0.0.0/8 static route to the clients through the ingress router
- A default route to the internet through the egress router

Example:

```
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```


IPv4 Policy-Based (VLAN - VLAN) routing

A Citrix ADC appliance allows for policy-based routing instead of static routing, with routing decisions usually keyed against the incoming interface and/or VLAN rather than destination IP. Policy-based routing is either a convenient alternative, in case the client source IP address range is subject to periodic changes, or a mandatory consideration, in case a packet's destination IP address is not sufficient by itself to reach a routing decision (i.e. in case of overlapping client IP addresses across multiple VLANs).

Example:

```
add ns pbr fromWirelessToInternet ALLOW -nextHop 192.168.2.1 -vlan
    100 -priority 10

Done

 add ns pbr fromInternetToWireless ALLOW -nextHop 192.168.1.1 -vlan
     200 -priority 20

Done

apply ns pbrs
<!--NeedCopy-->
```

IPv6 Configuration

The following commands assign IPv6 SNIP per vlan. The example below assumes that the networks outlined in the Figure: A simple depiction of a Gi-LAN in this page have a /64 subnet mask:

Command:

```
add ns ip6 fd00:192:168:1::254/64 -vServer DISABLED -mgmtAccess
    DISABLED
add ns ip6 fd00:192:168:2::254/64 -vServer DISABLED -mgmtAccess
    DISABLED
bind vlan 100 -IPAddress fd00:192:168:1::254/64
bind vlan 200 -IPAddress fd00:192:168:2::254/64
<!--NeedCopy-->
```

IPv6 Routing

After IPv6 addressing is complete, IPv6 static routing might be configured:

- A fd00:10::/64 static route to the clients via the ingress router

- A default route to the internet via the egress router

Example:

```
add route6 fd00:10::/64 fd00:192:168:1::1
add route6 ::/0 fd00:192:168:2::1
<!--NeedCopy-->
```

Or using policy-based routing:

Example:

```
add ns pbr6 fromWirelessToInternetv6 ALLOW -vlan 100 -priority 10 -
    nextHop fd00:192:168:2::1

add ns pbr6 fromInternetToWirelessv6 ALLOW -vlan 200 -priority 20 -
    nextHop fd00:192:168:1::1

apply ns pbr6
<!--NeedCopy-->
```

LACP Redundancy and Failover

In case of an HA configuration, it's recommended to leverage the throughput option to configure a low threshold for the LACP channel. For instance, consider a 25Gbps Gi-LAN and a 4x10GbE channel between each Citrix ADC appliance in the HA pair and the upstream switch to provide N+1 link redundancy:

Example:

```
set interface LA/1 -haMonitor ON -throughput 29000
<!--NeedCopy-->
```

In case of a double-link failure between the primary appliance and the upstream switch the maximum Gi-LAN throughput that can be supported would fall to 20Gbps. A 29Gbps low threshold per the example above would result in a redundancy switchover event to the secondary appliance (which has not suffered similar link failures) so that Gi-LAN traffic is not affected.

Route Monitors

In addition to LACP redundancy, route monitor checks might be configured and associated with the HA pair configuration. Route monitor checks can be useful to detect failures between the Citrix ADC appliance and the next-hop routers, especially if said routers are not directly connected but through an upstream switch.

A typical HA route monitor configuration per the sample Gi-LAN in section 2.5.1 is outlined below:

```
add route 192.168.1.0 255.255.255.0 192.168.1.1 -msr ENABLED -monitor
    arp
add route 192.168.2.0 255.255.255.0 192.168.2.1 -msr ENABLED -monitor
    arp
bind HA node -routeMonitor 192.168.1.0 255.255.255.0
bind HA node -routeMonitor 192.168.2.0 255.255.255.0
<!--NeedCopy-->
```

TCP optimization configuration

September 14, 2021

Before configuring TCP optimization, apply the following basic configuration settings on the Citrix ADC appliance:

Initial configuration:

```
enable ns feature LB IPv6PT
enable ns mode FR L3 USIP MBF Edge USNIP PMTUD
disable ns feature SP
disable ns mode TCPB
set lb parameter -preferDirectRoute NO
set lb parameter -vServerSpecificMac ENABLED
set l4param -l2ConnMethod Vlan
set rsskeytype -rsstype SYMMETRIC
set ns param -useproxyport DISABLED
<!--NeedCopy-->
```

> **Note**
>
> Restart the Citrix ADC appliance if you change the rsskeytype system parameter.

TCP termination

For Citrix ADC T1 to apply TCP optimization it needs to first terminate incoming TCP traffic. Towards this end, a wildcard TCP vserver should be created and configured to intercept ingress traffic and then forward it to the Internet router.

Static or dynamic routing environment

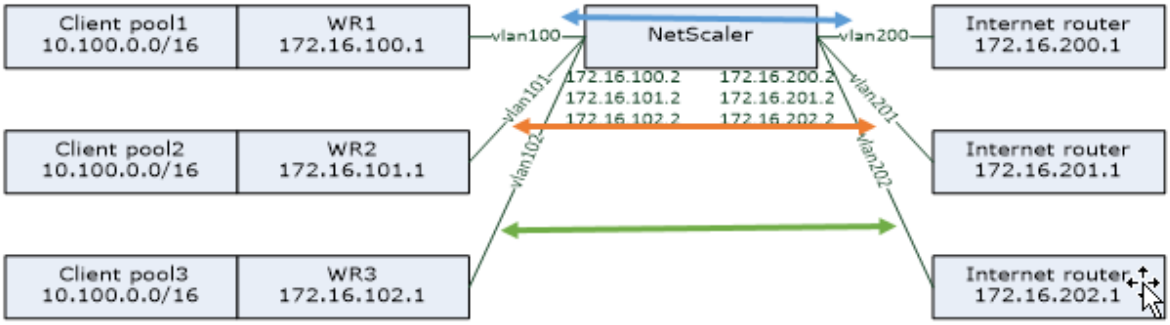
For environments with static or dynamic routing in place, vserver can rely on routing table info to forward packets towards internet router. Default route must point to the internet router and also routing entries for client subnets towards wireless router should be in place:

Example:

```
add lb vserver vsrv-wireless TCP * * -persistenceType NONE -
    Listenpolicy "CLIENT.VLAN.ID.EQ(100) && SYS.VSERVER(\"vsrv-wireless
    \").STATE.EQ(UP)" -m IP -cltTimeout 9000
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

VLAN-to-VLAN (PBR) environment

There are customer environments where subscriber traffic is segmented to multiple flows and needs to be forwarded to different routers based on incoming traffic parameters. Policy Based Routing (PBR) can be used to route packets based on incoming packet parameters, such as VLAN, MAC address, Interface, source IP, source port, destination IP address, and destination port.



Example:

```
add lb vserver vsrv-wireless TCP * * -m IP -l2Conn ON -listenpolicy "
    CLIENT.VLAN.ID.EQ(100) || CLIENT.VLAN.ID.EQ(101) || CLIENT.VLAN.ID.
    EQ(102)"

add ns pbr pbr-vlan100-to-vlan200 ALLOW -vlan 100 -nexthop 172.16.200.1

add ns pbr pbr-vlan101-to-vlan201 ALLOW -vlan 101 -nexthop 172.16.201.1

add ns pbr pbr-vlan102-to-vlan202 ALLOW -vlan 102 -nexthop 172.16.202.1
<!--NeedCopy-->
```


Using Policy Based Routing to route TCP optimized traffic is a new feature added in release 11.1 50.10. For previous releases, having multiple "mode MAC" vserver entities per VLAN is an alternative solution for multi-VLAN environments. Each vserver has a bound service representing the internet router for the particular flow.

Example:

```
add server internet_router_1 172.16.200.1

add server internet_router_2 172.16.201.1

add server internet_router_3 172.16.202.1

add service svc-internet-1 internet_router_1 TCP * -usip YES -
   useproxyport NO

add service svc-internet-2 internet_router_2 TCP * -usip YES -
   useproxyport NO

add service svc-internet-3 internet_router_3 TCP * -usip YES -
   useproxyport NO

bind service svc-internet-1 -monitorName arp

bind service svc-internet-2 -monitorName arp

bind service svc-internet-3 -monitorName arp

add lb vserver vsrv-wireless-1 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (100) && SYS.VSERVER(\"vsrv-wireless-1\").STATE.EQ(UP)" -m MAC -
   l2Conn ON

add lb vserver vsrv-wireless-2 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (101) && SYS.VSERVER(\"vsrv-wireless-2\").STATE.EQ(UP)" -m MAC -
   l2Conn ON

add lb vserver vsrv-wireless-3 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
   (102) && SYS.VSERVER(\"vsrv-wireless-3\").STATE.EQ(UP)" -m MAC -
   l2Conn ON

bind lb vserver vsrv-wireless-1 svc-internet-1

bind lb vserver vsrv-wireless-2 svc-internet-2

bind lb vserver vsrv-wireless-3 svc-internet-3
```



```
<!--NeedCopy-->
```

> **Note:**
>
> The vserver mode is MAC in contrast to previous examples where it is mode IP. This is required to retain the destination IP information when we have service(s) bound to vserver. Also, the additional PBR configuration need to route non-optimized traffic.

TCP optimization

Out-of-the-box Citrix ADC TCP termination is configured for TCP pass-through functionality. TCP pass-through essentially means that Citrix ADC T1 may transparently intercept a client-server TCP stream but does not retain separate client/server buffers or otherwise apply any optimization techniques.

To enable TCP optimization a TCP profile, named as nstcpprofile, is used to specify TCP configurations that is used if no TCP configurations are provided at the service or virtual server level and it should be modified as follows:

Command:

```
add ns tcpProfile nstcpprofile -WS ENABLED -SACK ENABLED -WSVal 8 -mss
    1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
    ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
    spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
    fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT
<!--NeedCopy-->
```

> **Note:**
>
> If there is not any profile explicitly created and bound to vserver and service, the profile nstcp_default_profile is bound by default.
>
> In case of multiple TCP profiles requirement, extra TCP profiles can be created and associated with the appropriate virtual server

Command:

```
add ns tcpProfile custom_profile -WS ENABLED -SACK ENABLED -WSVal 8 -
    mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
    ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
    spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
    fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

set lb vserver vsrv-wireless -tcpProfileName custom_profile
```



```
<!--NeedCopy-->
```

> **Note:**
>
> For deployments with vserver -m MAC and service, same profile should be associated with service.

```
set service svc-internet -tcpProfileName custom_profile
<!--NeedCopy-->
```

TCP optimization capabilities

Most of the relevant TCP optimization capabilities of a Citrix ADC appliance are exposed through a corresponding TCP profile. Typical CLI parameters that should be considered when creating a TCP profile are the following:

1. **Window Scaling (WS)**: TCP Window scaling allows increasing the TCP receive window size beyond 65535 bytes. It helps improving TCP performance overall and specially in high bandwidth and long delay networks. It helps with reducing latency and improving response time over TCP.
2. **Selective acknowledgment (SACK)**: TCP SACK addresses the problem of multiple packet loss which reduces the overall throughput capacity. With selective acknowledgement the receiver can inform the sender about all the segments which are received successfully, enabling sender to only retransmit the segments which were lost. This technique helps T1 improve overall throughput and reduce the connection latency.
3. **Window Scaling Factor (WSVal)**: Factor used to calculate the new window size. It must be configured with a high value in order to allow the advertised window by NS to be at least equal to the buffer size.
4. **Maximum Segment Size (MSS)**: MSS of a single TCP segment. This value depends on the MTU setting on intermediate routers and end clients. A value of 1460 corresponds to an MTU of 1500.
5. **maxBurst**: Maximum number of TCP segments allowed in a burst.
6. **Initial Congestion Window size(initialCwnd)**: TCP initial congestion window size determines the number of bytes which can be outstanding in beginning of the transaction. It enables T1 to send those many bytes without bothering for congestion on the wire.
7. **Maximum OOO packet queue size(oooQSize)**: TCP maintains Out Of Order queue to keep the OOO packets in the TCP communication. This setting impacts system memory if the queue size is long as the packets need to be kept in runtime memory. Thus this needs to be kept at optimized level based on the kind of network and application characteristics.
8. **Minimum RTO(minRTO)**: The TCP retransmission timeout is calculated on each received ACK based on internal implementation logic. The default retransmission timeout happens at 1 second to start with and this can be tweaked with this setting. For second retransmission of these packets RTO will be calculated by N*2 and then N*4 … N*8… goes on till last retransmission

attempt.

9. **bufferSize/sendBuffsize**: these refer to the maximum amount of data that the T1 may receive from the server and buffer internally without sending to the client. They should be set to a value larger (at least double) than the Bandwidth Delay Product of the underlying transmission channel.
10. **flavor**: this refers to the TCP congestion control algorithm. Valid values are Default, BIC, CUBIC, Westwood and Nile.
11. **Dynamic receive buffering**: allows the receive buffer to be adjusted dynamically based on memory and network conditions. It will fill up the buffer as much as it's required to keep the client's download pipe full instead of filling up, by reading ahead from server, a fixed size buffer, as latter is specified in TCP profile and typically based on criteria such as 2*BDP, for a connection. Citrix ADC T1 monitors the network conditions to the client and estimates how much it should read ahead from the server.
12. **Keep-Alive (KA)**: Send periodic TCP keep-alive (KA) probes to check if peer is still up.
13. rstWindowAttenuate: Defending TCP against spoofing attacks. It will reply with corrective ACK when a sequence number is invalid.
14. **rstMaxAck**: Enable or disable acceptance of RST that is out of window yet echoes highest ACK sequence number.
15. **spoofSynDrop**: Drop of invalid SYN packets to protect against spoofing.
16. **Explicit Congestion Notification(ecn)**: It sends notification of the network congestion status to the sender of the data and takes corrective measures for data congestion or data corruption.
17. **Forward RTO-Recovery**: In case of spurious retransmissions, the congestion control configurations are reverted to their original state.
18. **TCP maximum congestion window (maxcwnd)**: TCP maximum congestion window size that is user configurable.
19. **Forward acknowledgment (FACK)**: To avoid TCP congestion by explicitly measuring the total number of data bytes outstanding in the network, and helping the sender (either T1 or a client) control the amount of data injected into the network during retransmission timeouts.
20. **tcpmode**: TCP optimization modes for specific profile. There are two TCP optimization modes - Transparent and Endpoint.

- Endpoint. In this mode, the appliance manages the client and server connections seperately.
- Transparent. In the transparent mode the clients need to access the servers directly, with no intervening virtual server. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a NetScaler appliance is placed between the client and the server, so the traffic must pass through the appliance.

Silently dropping idle connections

In a telco network, almost 50 percent of a Citrix ADC appliance's TCP connections become idle, and the appliance sends RST packets to close them. The packets sent over radio channels activate those channels unnecessarily, causing a flood of messages that in turn cause the appliance to generate a flood of service-reject messages. The default TCP profile now includes DropHalfClosedConnOnTimeout and DropEstConnOnTimeout parameters, which by default are disabled. If you enable both of them, neither a half closed connection nor an established connection causes a RST packet to be sent to the client when the connection times out. The appliance just drops the connection.

```
set ns tcpProfile nstcpprofile  -DropHalfClosedConnOnTimeout ENABLED
set ns tcpProfile nstcpprofile  -DropEstConnOnTimeout ENABLED
<!--NeedCopy-->
```

Analytics and Reporting

September 14, 2021

The TCP Speed Reporting is a Citrix ADC feature which extracts TCP connection statistics, as a measure of TCP download and upload performance, and is utilized in TCP Insight reports of the Citrix Application Delivery Management (ADM). To achieve this, Citrix ADC monitors each TCP connection, locates packet bursts on an idle time-out basis and reports key metrics (such as byte count, retransmitted byte count and duration) for the identified maximum burst. TCP Speed Reporting feature is enabled by default, support both TCP and HTTP vServers and depends on the Appflow/ULFD reporting infrastructure.

Real-time Statistics

September 14, 2021

The stat command might be used to verify that TCP optimization is properly applied:

Command:

```
> stat lb vserver vsrv-wireless
Virtual Server Summary
                      vsvrIP  port     Protocol        State   Health
                         actSvcs
vsrv...eless              *     0          TCP           UP      100
           1
```



```
5
6             inactSvcs
7 vsrv...eless       0
8 Virtual Server Statistics
9                                                Rate (/s)
                                                    Total
10 Vserver hits                                          0
       10
11 Requests                                              0
                          0
12 Responses                                             0
                          0
13 Request bytes                                         0
       1580
14 Response bytes                                        0
       532594360
15 Total Packets rcvd                                    0
       216463
16 Total Packets sent                                    0
       369898
17 Current client connections                           --
                          0
18 Current Client Est connections                       --
                          0
19 Current server connections                           --
                          0
20 Requests in surge queue                              --
                          0
21 Requests in vserver's surgeQ                         --
                          0
22 Requests in service's surgeQs                        --
                          0
23 Spill Over Threshold                                 --
                          0
24 Spill Over Hits                                      --
                          0
25 Labeled Connection                                   --
                          0
26 Push Labeled Connection                              --
                          0
27 Deferred Request                                      0
                          0
28 Invalid Request/Response                             --
                          0
29 Invalid Request/Response Dropped                     --
```



```
                          0
30 Bound Service(s) Summary
31                             IP  port          Type         State     Hits
                                  Hits/s
32 svc-internet     192.168.2.2     0           TCP            UP       10
          0/s
33
34                  Req    Req/s      Rsp    Rsp/s Throughp ClntConn
                       SurgeQ
35 svc-internet       0      0/s        0      0/s         0        0
             0
36                   SvrConn   ReuseP  MaxConn ActvTran  SvrTTFB     Load
37 svc-internet         0        0        0        0        0        0
```

The Total counters should constantly increase for an operational system. In addition, the Rate counters should be non-zero.

> **Note**
>
> The preceding output is from an operational yet idle lab system, explaining the zero rate.

SNMP

September 14, 2021

SNMP agent can be queried for system specific information from a remote device (SNMP Manager). Based on the query, the agent searches for the equal object identifier (OID) in the management information base (MIB) for the data requested and sends the information to the SNMP Manager. The following are the most useful SNMP OIDs for Telco deployments:

Memory

- **resMemUsage (1.3.6.1.4.1.5951.4.1.1.41.2)**

Percentage of memory utilization on Citrix ADC.

Packet Engine CPU

- **resCpuUsage (1.3.6.1.4.1.5951.4.1.1.41.1)**

CPU utilization percentage.

- **nsCPUTable (1.3.6.1.4.1.5951.4.1.1.41.6)**

This table contains information about each CPU in Citrix ADC.

Indexed on: nsCPUname

- **nsCPUname (1.3.6.1.4.1.5951.4.1.1.41.6.1.1)**

The name of the CPU.

- **nsCPUusage (1.3.6.1.4.1.5951.4.1.1.41.6.1.2)**

CPU utilization percentage.

Throughput

- **allNicTotRxMbits (1.3.6.1.4.1.5951.4.1.1.71.1)**

Number of megabits received by the Citrix ADC appliance.

- **allNicTotTxMbits (1.3.6.1.4.1.5951.4.1.1.71.2)**

Number of megabits transmitted by the Citrix ADC appliance.

- **ipTotRxPkts (1.3.6.1.4.1.5951.4.1.1.43.25)**

IP packets received.

- **ipTotRxMbits (1.3.6.1.4.1.5951.4.1.1.43.27)**

Megabits of IP data received.

- **ipTotTxPkts (1.3.6.1.4.1.5951.4.1.1.43.28)**

IP packets transmitted.

- **ipTotTxMbits (1.3.6.1.4.1.5951.4.1.1.43.30)**

Megabits of IP data transmitted.

Connections

Active connections:

- **tcpActiveServerConn (1.3.6.1.4.1.5951.4.1.1.46.8)**

Connections to a server currently responding to requests.

Total connections:

- **tcpCurServerConn (1.3.6.1.4.1.5951.4.1.1.46.1)**

Server connections, including connections in the Opening, Established, and Closing state.

- **tcpCurClientConn (1.3.6.1.4.1.5951.4.1.1.46.2)**

Client connections, including connections in the Opening, Established, and Closing state.

Note: Because of SYN-Cookie, this doesn't include Client in Opening state

- **tcpTotZombieCltConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.26)**

Client connections that are flushed because the client has been idle for some time.

- **tcpTotZombieSvrConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.27)**

Server connections that are flushed because there have been no client requests in the queue for some time.

Errors

- **tcpErrSynGiveUp (1.3.6.1.4.1.5951.4.1.1.46.37)**

Attempts to establish a connection on the Citrix ADC that timed out.

- **tcpErrRetransmitGiveUp (1.3.6.1.4.1.5951.4.1.1.46.60)**

Number of times Citrix ADC terminates a connection after retransmitting the packet seven times on that connection. Retransmission happens when recieving end doesn't acknowledges the packet.

- **ifInDiscards (1.3.6.1.2.1.2.2.1.13)**

The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

- **ifOutDiscards (1.3.6.1.2.1.2.2.1.19)**

The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

- **ifErrTxOverflow (1.3.6.1.4.1.5951.4.1.1.54.1.36)**

Number of packets that have passed through the overflow queues, during transmission on the specified interface, since the Citrix ADC appliance was started or the interface statistics were cleared. This gets incremented only on congested ports.

Optimized/Bypass connections

- **tcpOptimizationEnabled (1.3.6.1.4.1.5951.4.1.1.46.131)**

Total number of connections enabled with TCP optimization.

- **tcpOptimizationBypassed (1.3.6.1.4.1.5951.4.1.1.46.132)**

Total number of connections bypassed TCP Optimization.

Technical Recipes

September 14, 2021

The Citrix ADC T1 models provide advanced features and a powerful policy configuration language that allow for evaluation of complex decision in runtime.

While it is not possible to evaluate all capabilities that are potentially unlocked by the T1000 features and policy configuration guide, technical receipes consider implementation of various requirements brought in by Telco operators. Feel free to re-use the "recipes" as is or adapt to your environment.

Per-user Connection Limit

The Citrix ADC T1 model can be configured to limit the number of connections per unique subscriber IP. With the below configuration, N concurrent TCP connections per IP (CLIENT.IP.SRC) is allowed. For every attempt for connection beyond the configured threshold, T1 sends an RST. For maximum 2 concurrent connections per user:

Command:

```
add stream selector streamSel_usrlimit CLIENT.IP.SRC
add ns limitIdentifier limitId_usrlimit -threshold 2 -mode CONNECTION -
    selectorName streamSel_usrlimit
add responder policy respPol_usrlimit "SYS.CHECK_LIMIT(\"
    limitId_usrlimit\")" RESET
bind lb vserver vsrv-wireless -policyName respPol_usrlimit -priority 1
    -gotoPriorityExpression END
<!--NeedCopy-->
```

Smooth Insertion/Deletion of Vserver

Many operators concern about TCP connections disruption when the Citrix ADC T1 model is activated inline for TCP optimization or when it is disabled for maintenance purposes. To avoid breaking existing connections when vserver is introduced, the following configuration needs to be applied before configuring or activating vserver for TCP optimization:

Command:

```
add ns acl acl-ingress ALLOW  - vlan 100
add forwardingSession fwd-ingress  - aclname acl-ingress
apply ns acls
<!--NeedCopy-->
```


Forwarding sessions are effective on top of routing (either static or dynamic or PBR) and create session entries for traffic that is routed (L3 mode). Any existing connection is handled by forwarding session due to corresponding sessions, and upon vserver introduction it starts capturing only new TCP connections.

ACLs can be configured to capture only specific ports like vserver, in order to avoid creating sessions for unnecessary traffic, which is memory consuming. Another option is to remove specific configuration after vserver activation.

For maintenance purposes, vserver should be disabled and its state appears as OUT OF SERVICE. When this happens, the vserver terminates all connections immediately by default. To make vserver to still serve the existing connections and not accept new, the following configuration should be applied:

Command:

```
set lb vserver vsrv-wireless  -downStateFlush DISABLED
<!--NeedCopy-->
```

New connections go through the routing table, and corresponding session entries are created due to forwarding sessions.

Policy-Based TCP Profiling

Policy-based TCP Profile selection allows operators to configure TCP profile dynamically for clients coming from different traffic domains (i.e. 3G or 4G). Some of the QoS metrics are different for these traffic domains, and in order to achieve better performance, you need to change some of the TCP parameter dynamically. Consider a case where clients coming from 3G and 4G hit same vserver and use same TCP profile, which have negative impact on some client's performance. AppQoE functionality can classify these clients and dynamically change TCP profile on vserver.

Example:

```
enable feature AppQoE

add ns tcpProfile nstcpprofile1 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 4000000 -flavor BIC -KA ENABLED -
    sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -tcpmode
    ENDPOINT

add ns tcpProfile nstcpprofile2 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 15 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 128000 -flavor BIC -KA ENABLED -
```



```
    sendBuffsize 6000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 64000 -fack ENABLED -tcpmode ENDPOINT
6
7 add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1
8
9 add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2
10
11 add appqoe policy appqoe_4G -rule "CLIENT.VLAN.ID.EQ(100)" -action
    action_1
12
13 add appqoe policy appqoe_3G -rule "CLIENT.VLAN.ID.EQ(200)" -action
    action_2
14
15 bind lb vserver vsrv-wireless -policyName appqoe_4G -priority 100
16
17 bind lb vserver vsrv-wireless -policyName appqoe_3G -priority 110
18 <!--NeedCopy-->
```

The Citrix ADC T1 model is capable to receive the subscriber information dynamically through Gx or Radius or Radius and Gx interface and apply different TCP profile on a per-subscriber basis.

Command:

```
1 add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1
2
3 add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2
4
5 add appqoe policy appqoe_4G -rule "SUBSCRIBER.RULE_ACTIVE(\"3G\")" -
    action action_1
6
7 add appqoe policy appqoe_3G -rule "SUBSCRIBER.RULE_ACTIVE(\"4G\")" -
    action action_2
8 <!--NeedCopy-->
```

For integration of the Citrix ADC T1 model with operator control-plane network, see Telco Subscriber Management.

Scalability

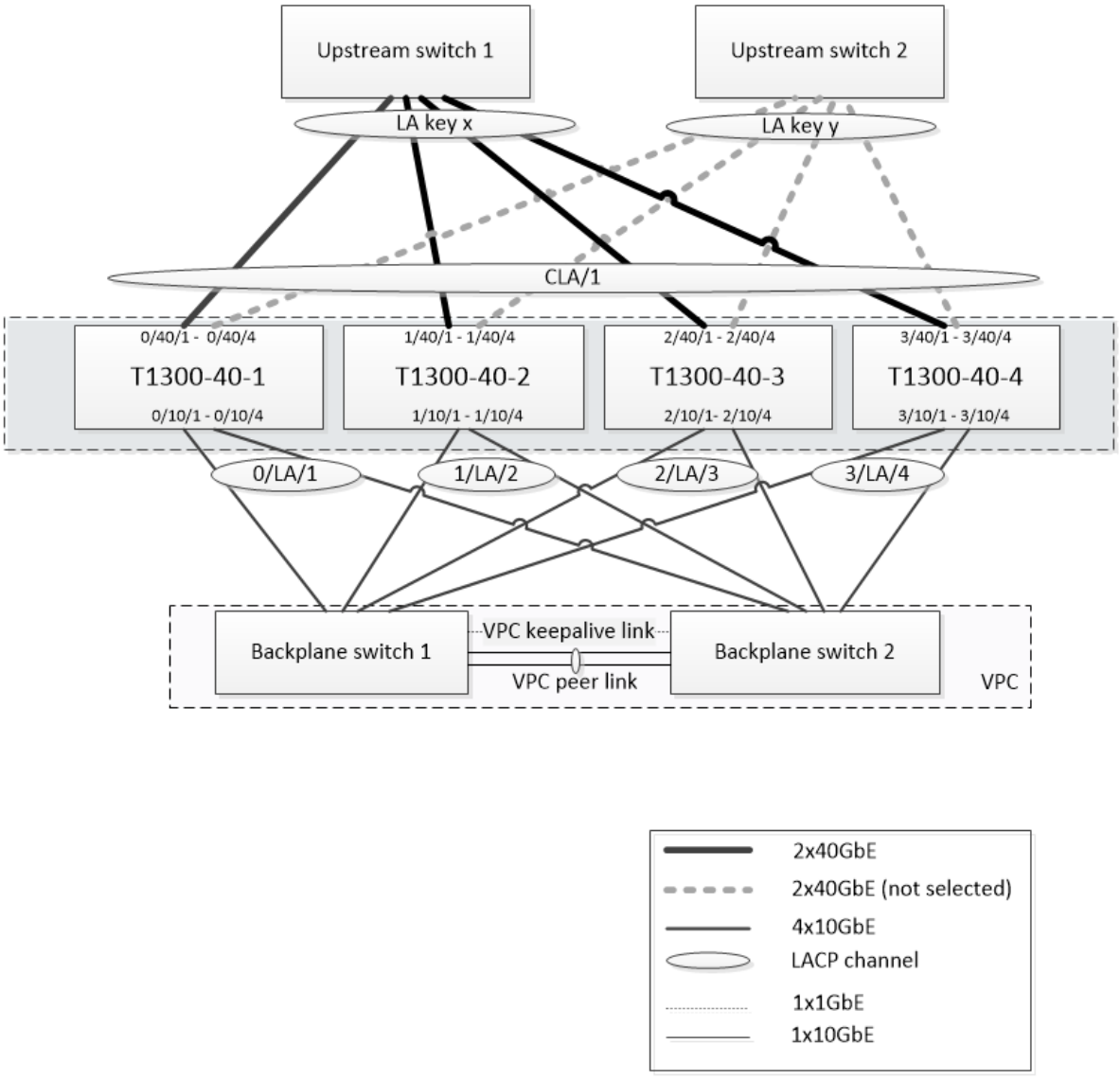
September 14, 2021

Because TCP optimization is resource intensive, a single Citrix ADC appliance, even a high end –appliance, might not be able to sustain high Gi-LAN throughputs. To expand the capacity of your

network, you can deploy Citrix ADC appliances in an N+1 cluster formation. In a cluster deployment, the Citrix ADC appliances work together as a single system image. The client traffic is distributed across the cluster nodes with the help of external switch device.

Topology

Figure 1 is an example of a cluster consisting of four T1300-40G nodes.



The setup shown in Figure 1 has the following properties:

1. All cluster nodes belong to the same network (also known as an L2 cluster).
2. Data plane and backplane traffic are handled by different switches.
3. Assuming Gi-LAN throughput is 200 Gbps and that a T1300-40G appliance can sustain 80Gbps of throughput, we need three T1300-40G appliances. To provide redundancy in case of single

cluster node failure, we deploy four appliances in total.

4. Each node will receive up to 67Gbps of traffic (50Gbps in normal operating conditions and 67Gbps in case of single cluster node failure), so it needs 2x40Gbps connections to the upstream switch. To provide redundancy in case of switch failure, we deploy a couple of upstream switches and double the number of connections.
5. Cluster Link Aggregation (CLAG) is used to distribute traffic across cluster nodes. A single CLAG handles both client and server traffic. Link Redundancy is enabled on the CLAG, so only one "subchannel" is selected at any given time and handles the traffic. If some link fails or throughput falls below specified threshold, the other subchannel is selected.
6. The upstream switch performs symmetric port-channel load balancing (for example, source-dest-ip-only algorithm of Cisco IOS 7.0(8) N1(1)) so that forward and reverse traffic flows are handled by the same cluster node. This property is desirable because it eliminates packet reordering, which would degrade TCP performance.
7. Fifty percent of data traffic is expected to be steered to backplane, which means each node will steer up to 34Gbps to other cluster nodes (25Gbps in normal operating conditions and 34Gbps in case of single cluster node failure). Thus, each node needs at least 4x10G connections to the backplane switch. To provide redundancy in case of switch failure, we deploy a couple of backplane switches and double the number of connections. Link redundancy is not currently supported for backplane, so Cisco VPC or equivalent technology is desired to achieve switch-level redundancy.
8. MTU size of steered packets is 1578 bytes, so backplane switches must support an MTU more than 1500 bytes.

Note: The design depicted in Figure 1 is also applicable to T1120 and T1310 appliances. For T1310 we would use 40GbE interfaces for the backplane connections, since it lacks 10GbE ports.

Note: While this document uses Cisco VPC as an example, if working with non-Cisco switches alternate equivalent solutions could be used, such as Juniper's MLAG.

Note: While other topologies such as ECMP instead of CLAG are possible, they are not currently supported for this particular use case.

Configuring TCP Optimization in a Citrix ADC T1000 Cluster

After physical installation, physical connectivity, software installation, and licensing are completed, you can proceed with the actual cluster configuration. The configurations described below apply to the cluster depicted in Figure 1.

Note: For more information about cluster configuration, see Setting up a Citrix ADC cluster.

Assume that the four T1300 nodes in Figure 1 have the following NSIP addresses:

Four T1300 nodes with NSIP address:


```
T1300-40-1: 10.102.29.60
T1300-40-2: 10.102.29.70
T1300-40-3: 10.102.29.80
T1300-40-4: 10.102.29.90
```

The cluster will be managed through the cluster IP (CLIP) address, which is assumed to be 10.78.16.61.

Setting Up the Cluster

To begin configuring the cluster shown in Figure 1, log on to the first appliance that you want to add to the cluster (for example, T1300-40-1) and do the following.

1. At the command prompt, enter the following commands:

   Command:

   ```
   > add cluster instance 1
   > add cluster node 0 10.102.29.60 -state ACTIVE
   > add ns ip 10.102.29.61 255.255.255.255 -type clip
   > enable cluster instance 1
   > save ns config
   > reboot  - warm
   ```

2. After the appliance restarts, connect to the Cluster IP (CLIP) address and add the rest of the nodes to the cluster:

   Command:

   ```
   > add cluster node 1 10.102.29.70 -state ACTIVE
   > add cluster node 2 10.102.29.80 -state ACTIVE
   > add cluster node 3 10.102.29.90  - state ACTIVE
   > save ns config
   ```

3. Connect to the NSIP address of each of the newly added nodes and join the cluster:

   Command:

   ```
   > join cluster -clip 10.102.29.61 -password nsroot
   > save ns config
   > reboot  - warm
   ```

4. After the nodes restart, proceed with backplane configuration. On the cluster IP address, enter the following commands to create an LACP channel for the backplane link of each cluster node:

   Command:


```
> set interface 0/10/[1-8] -lacpkey 1 -lacpmode ACTIVE
> set interface 1/10/[1-8] -lacpkey 2 -lacpmode ACTIVE
> set interface 2/10/[1-8] -lacpkey 3 -lacpmode ACTIVE
> set interface 3/10/[1-8] -lacpkey 4 -lacpmode ACTIVE
```

5. Similarly, configure dynamic LA and VPC on the backplane switches. Make sure the MTU of the backplane switch interfaces is at least 1578 bytes.
6. Verify the channels are operational:

   Command:

```
> show channel 0/LA/1
> show channel 1/LA/2
> show channel 2/LA/3
> show channel 3/LA/4
```

7. Configure the cluster node backplane interfaces.

   Command:

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/LA/3
> set cluster node 3 -backplane 3/LA/4
```

8. Check the cluster status and verify that the cluster is operational:

```
> show cluster instance
> show cluster node
```

For more information on cluster setup, see Setting up a Citrix ADC cluster

Distributing Traffic Across Cluster Nodes

After you have formed theCitrix ADC cluster, deploy Cluster Link Aggregation (CLAG) to distribute traffic across cluster nodes. A single CLAG link will handle both client and server traffic.

On the cluster IP address, execute the following commands to create the Cluster Link Aggregation (CLAG) group shown in Figure 1:

Command:

```
> set interface 0/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
```



```
4 > set interface 3/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
```

Configure dynamic link aggregation on the external switches.

Then, enable Link Redundancy as follows:

Code:

```
1 > set channel CLA/1 -linkRedundancy ON -lrMinThroughput 240000
```

Finally, check the channel status by entering:

Command:

```
1 > show channel CLA/1
```

The channel should be UP and the actual throughput should be 320000.

For more information about cluster link aggregation, see the following topics:

- Dynamic Cluster Link Aggregation
- Link Redundancy in a Cluster with LACP.

Because we will be using MAC-based forwarding (MBF), configure a linkset and bind it to the CLAG group as follows:

Command:

```
1 > add linkset LS/1
2 > bind linkset LS/1 -ifnum CLA/1
```

More information about linksets, see the following topics:

- Configuring Linksets
- Using Cluster LA Channel with Linksets

Configuring VLAN and IP Addresses

We will be using striped IP configuration, which means that IP addresses are active on all nodes (default setting). See Striped, Partially Striped, and Spotted Configurations for more information about this topic.

1. Add the ingress and egress SNIPs:

   Command:

   ```
   1 > add ns ip 172.16.30.254 255.255.255.0 -type SNIP
   2 > add ns ip 172.16.31.254 255.255.255.0 -type SNIP
   3 > add ns ip6 fd00:172:16:30::254/112 -type SNIP
   ```



```
4 > add ns ip6 fd00:172:16:31::254/112 -type SNIP
```

2. Add the corresponding ingress and egress VLANs:

   Command:

```
1 > add vlan 30 -aliasName wireless
2 > add vlan 31 -aliasName internet
```

3. Bind VLANs with IPs and linkset:

   Command:

```
1 > bind vlan 31 -ifnum LS/1 -tagged
2 > bind vlan 30 -ifnum LS/1 -tagged
3 > bind vlan 30 -IPAddress 172.16.30.254 255.255.255.0
4 > bind vlan 31 -IPAddress 172.16.31.254 255.255.255.0
5 > bind vlan 30 -IPAddress fd00:172:16:30::254/112
6 > bind vlan 31 -IPAddress fd00:172:16:31::254/112
```

More ingress and egress VLANs can be added if needed.

Configuring TCP Optimization

At this point, we have applied all cluster specific commands. To complete the configuration, follow the steps described in TCP optimization configuration.

Configuring Dynamic Routing

A Citrix ADC cluster can be integrated into the dynamic routing environment of the customer's network. Following is an example of dynamic routing configuration using BGP routing protocol (OSPF is also supported).

1. From the CLIP address, enable BGP and dynamic routing on ingress and egress IP addresses:

   Command:

```
1 > enable ns feature bgp
2 > set ns ip 172.16.30.254 -dynamicRouting ENABLED
3 > set ns ip 172.16.31.254 -dynamicRouting ENABLED
```

2. Open vtysh and configure BGP for the egress side:

   Code:


```
> shell
root@ns# vtysh
ns# configure terminal
ns(config)# router bgp 65531
ns(config-router)# network 10.0.0.0/24
ns(config-router)# neighbor 172.16.31.100 remote-as 65530
ns(config-router)# neighbor 172.16.31.100 update-source
    172.16.31.254
ns(config-router)# exit
ns(config)# ns route-install propagate
ns(config)# ns route-install default
ns(config)# ns route-install bgp
ns(config)# exit
```

3. Configure the egress-side BGP peer to advertise the default route to the Citrix ADC cluster. For example:

   Command:

```
router bgp 65530
 bgp router-id 172.16.31.100
 network 0.0.0.0/0
 neighbor 172.16.31.254 remote-as 65531
```

4. Follow similar steps to configure the ingress side.
5. From vtysh verify that configuration is propagated to all cluster nodes, by entering:

   Command:

```
ns# show running-config
```

6. Finally, log on to NSIP address of each cluster node and verify routes advertised from BGP peer:

   Command:

```
> show route | grep BGP
```

Optimizing TCP Performance using TCP Nile

September 14, 2021

TCP uses the following optimization techniques and congestion control strategies (or algorithms) to avoid network congestion in data transmission.

Congestion Control Strategies

The Transmission Control Protocol (TCP) has long been used to establish and manage Internet connections, handle transmission errors, and smoothly connect web applications with client devices. But network traffic has become more difficult to control, because packet loss does not depend only on the congestion in the network, and congestion does not necessarily cause packet loss. Therefore, to measure congestion, a TCP algorithm should focus on both packet loss and bandwidth.

NILE Algorithm

Citrix Systems has developed a new congestion-control algorithm, NILE, a TCP optimization algorithm designed for high-speed networks such as LTE, LTE advanced and 3G. Nile addresses unique challenges caused by fading, random or congestive losses, link layer retransmissions and carrier aggregation.

The NILE algorithm:

- Bases queue-latency estimates on round-trip time measurements.
- Uses a congestion-window-increase function that is inversely proportional to the measured queue latency. This method results in approaching the network congestion point more slowly than does the standard TCP method, and reduces the packet losses during congestion.
- Can distinguish between random loss and congestion based loss on the network by using the estimated queue latency.

The telecom service providers can use the NILE algorithm in their TCP infrastructure to:

- Optimize mobile and long-distance networks— The NILE algorithm achieves higher throughput compared to standard TCP. This feature is especially important for mobile and long-distance networks.
- Decrease application perceived latency and enhance subscriber experience— The Nile algorithm uses packet loss information to determine whether the transmission-window size should be increased or decreased, and uses queuing delay information to determine the size of the increment or decrement. This dynamic setting of transmission-window size decreases the application latency on the network.

To configure NILE support using the command line interface

At the command prompt, type the following:

```
set ns tcpProfile <name> [-flavor NILE]
<!--NeedCopy-->
```


Configuring NILE support using the configuration utility

1. Navigate to **System** > **Profiles** > **TCP Profiles** and click **TCP profiles**.
2. From the **TCP Flavor** drop-down list, select **NILE**.

Example:

```
set ns tcpProfile tcpprofile1 -flavor NILE
<!--NeedCopy-->
```

Proportional Rate Recovery (PRR) Algorithm

TCP Fast Recovery mechanisms reduce web latency caused by packet losses. The new Proportional Rate Recovery (PRR) algorithm is a fast recovery algorithm that evaluates TCP data during a loss recovery. It is patterned after Rate-Halving, by using the fraction that is appropriate for the target window chosen by the congestion control algorithm. It minimizes window adjustment, and the actual window size at the end of recovery is close to the Slow-Start threshold (ssthresh).

TCP Fast Open (TFO)

TCP Fast Open (TFO) is a TCP mechanism that enables speedy and safe data exchange between a client and a server during TCP's initial handshake. This feature is available as a TCP option in the TCP profile bound to a virtual server of a Citrix ADC appliance. TFO uses a TCP Fast Open Cookie (a security cookie) that the Citrix ADC appliance generates to validate and authenticate the client initiating a TFO connection to the virtual server. By using the TFO mechanism, you can reduce an application's network latency by the time required for one full round trip, which significantly reduces the delay experienced in short TCP transfers.

How TFO works

When a client tries to establish a TFO connection, it includes a TCP Fast Open Cookie with the initial SYN segment to authenticate itself. If authentication is successful, the virtual server on the Citrix ADC appliance can include data in the SYN-ACK segment even though it has not received the final ACK segment of the three-way handshake. This saves up to one full round-trip compared to a normal TCP connection, which requires a three-way handshake before any data can be exchanged.

A client and a backend server perform the following steps to establish a TFO connection and exchange data securely during the initial TCP handshake.

1. If the client does not have a TCP Fast Open Cookie to authenticate itself, it sends a Fast Open Cookie request in the SYN packet to the virtual server on the Citrix ADC appliance.

2. If the TFO option is enabled in the TCP profile bound to the virtual server, the appliance generates a cookie (by encrypting the client's IP address under a secret key) and responds to the client with a SYN-ACK that includes the generated Fast Open Cookie in a TCP option field.
3. The client caches the cookie for future TFO connections to the same virtual server on the appliance.
4. When the client tries to establish a TFO connection to the same virtual server, it sends SYN that includes the cached Fast Open Cookie (as a TCP option) along with HTTP data.
5. The Citrix ADC appliance validates the cookie, and if the authentication is successful, the server accepts the data in the SYN packet and acknowledges the event with a SYN-ACK, TFO Cookie, and HTTP Response.

Note: If the client authentication fails, the server drops the data and acknowledges the event only with a SYN indicating a session timeout.

1. On the server side, if the TFO option is enabled in a TCP profile bound to a service, the Citrix ADC appliance determines whether the TCP Fast Open Cookie is present in the service to which it is trying to connect.
2. If the TCP Fast Open Cookie is not present, the appliance sends a cookie request in the SYN packet.
3. When the backend server sends the Cookie, the appliance stores the cookie in the server information cache.
4. If the appliance already has a cookie for the given destination IP pair, it replaces the old cookie with the new one.
5. If the cookie is available in the server information cache when the virtual server tries to reconnect to the same backend server by using the same SNIP address, the appliance combines the data in SYN packet with the cookie and sends it to the backend server.
6. The backend server acknowledges the event with both data and a SYN.

Note: If the server acknowledges the event with only a SYN segment, the Citrix ADC appliance immediately resends the data packet after removing the SYN segment and the TCP options from the original packet.

Configuring TCP Fast Open

To use the TCP Fast Open (TFO) feature, enable the TCP Fast Open option in the relevant TCP profile and set the TFO Cookie Timeout parameter to a value that suits the security requirement for that profile.

To enable or disable TFO by using the command line

At the command prompt, type one of the following commands to enable or disable TFO in a new or existing profile.

Note: The default value is DISABLED.

```
add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - tcpFastOpen
<!--NeedCopy-->
```

Examples:

add tcpprofile Profile1 – tcpFastOpen
Set tcpprofile Profile1 – tcpFastOpen Enabled
unset tcpprofile Profile1 – tcpFastOpen

To set TCP Fast Open cookie timeout value by using the command line interface

At the command prompt, type:

```
set tcpparam  - tcpfastOpenCookieTimeout <Timeout Value>
<!--NeedCopy-->
```

Example:

```
set tcpprofile  - tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```

To configure the TCP Fast Open by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **TCP Fast Open** checkbox.
3. Click **OK** and then **Done**.

To Configure the TCP Fast Cookie timeout value by using the GUI

Navigate to **Configuration** > **System** > **Settings** > **Change TCP Parameters** and then **Configure TCP Parameters** page to set the TCP Fast Open Cookie timeout value.

TCP Hystart

A new TCP profile parameter, hystart, enables the Hystart algorithm, which is a slow-start algorithm that dynamically determines a safe point at which to terminate (ssthresh). It enables a transition to congestion avoidance without heavy packet losses. This new parameter is disabled by default.

If congestion is detected, Hystart enters a congestion avoidance phase. Enabling it gives you better throughput in high-speed networks with high packet loss. This algorithm helps maintain close to maximum bandwidth while processing transactions. It can therefore improve throughput.

Configuring TCP Hystart

To use the Hystart feature, enable the Cubic Hystart option in the relevant TCP profile.

To configure Hystart by using the command line interface (CLI)

At the command prompt, type one of the following commands to enable or disable Hystart in a new or existing TCP profile.

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

Examples:

```
add tcpprofile Profile1 - tcpFastOpen
Set tcpprofile Profile1 - tcpFastOpen Enabled
unset tcpprofile Profile1 - tcpFastOpen
<!--NeedCopy-->
```

To configure Hystart support by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **Cubic Hystart** check box.
3. Click **OK** and then **Done**.

Optimization Techniques

TCP uses the following optimization techniques and methods for optimized flow controls.

Policy based TCP Profile Selection

Network traffic today is more diverse and bandwidth-intensive than ever before. With the increased traffic, the effect that Quality of Service (QoS) has on TCP performance is significant. To enhance QoS, you can now configure AppQoE policies with different TCP profiles for different classes of network traffic. The AppQoE policy classifies a virtual server's traffic to associate a TCP profile optimized for a particular type of traffic, such as 3G, 4G, LAN, or WAN.

To use this feature, create a policy action for each TCP profile, associate an action with AppQoE policies, and bind the policies to the load balancing virtual servers.

Configuring Policy Based TCP Profile Selection

Configuring policy based TCP profile selection consists of the following tasks:

- Enabling AppQoE. Before configuring the TCP profile feature, you must enable the AppQoE feature.
- Adding AppQoE Action. After enabling the AppQoE feature, configure an AppQoE action with a TCP profile.
- Configuring AppQoE based TCP Profile Selection. To implement TCP profile selection for different classes of traffic, you must configure AppQoE policies with which your Citrix ADC appliance can distinguish the connections and bind the correct AppQoE action to each policy.
- Binding AppQoE Policy to Virtual Server. Once you have configured the AppQoE policies, you must bind them to one or more load balancing, content switching, or cache redirection virtual servers.

Configuring using the command line interface

To enable AppQoE by using the command line interface:

At the command prompt, type the following commands to enable the feature and verify that it is enabled:

```
enable ns feature appqoe

show ns feature
<!--NeedCopy-->
```

To bind a TCP profile while creating an AppQoE action using the command line interface

At the command prompt, type the following AppQoE action command with tcpprofiletobind option.

Binding a TCP Profile:

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS |
    NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <
    string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth
     <positive_integer>] [-priqDepth <positive_integer>] [-
    dosTrigExpression <expression>] [-dosAction ( SimpleResponse |
    HICResponse )] [-tcpprofiletobind <string>]

show appqoe action
```



```
<!--NeedCopy-->
```

To configure an AppQoE policy by using the command line interface

At the command prompt, type:

```
add appqoe policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

To bind an AppQoE policy to load balancing, cache redirection or content switching virtual servers by using the command line interface

At the command prompt, type:

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <
    priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <
    priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <
    priority>
<!--NeedCopy-->
```

Example:

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
    ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500 -
    slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
    sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack ENABLED
    -tcpmode ENDPOINT

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
    action appact1

bind lb vserver lb2 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST

bind cs vserver cs1 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```


Configuring Policy based TCP Profiling using the GUI

To enable AppQoE by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Advanced Features**.
3. In the **Configure Advanced Features** dialog box, select the **AppQoE** check box.
4. Click **OK**.

To configure AppQoE policy by using the GUI

1. Navigate to **App-Expert** > **AppQoE** > **Actions**.
2. In the details pane, do one of the following:
3. To create a new action, click **Add**.
4. To modify an existing action, select the action, and then click **Edit**.
5. In the **Create AppQoE Action** or the **Configure AppQoE Action** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):
    a) Name—name
    b) Action type—respondWith
    c) Priority—priority
    d) Policy Queue Depth—polqDepth
    e) Queue Depth—priqDepth
    f) DOS Action—dosAction
6. Click **Create**.

To bind AppQoE policy by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select a server and then click **Edit**.
2. In the **Policies** section and click (+) to bind an AppQoE policy.
3. In the **Policies** slider, do the following:
    a) Select a policy type as AppQoE from the drop-down list.
    b) Select a traffic type from the drop-down list.
4. In the **Policy Binding** section, do the following:
    a) Click **New** to create a new AppQoE policy.
    b) Click **Existing Policy** to select an AppQoE policy from the drop-down list.
5. Set the binding priority and click **Bind** to the policy to the virtual server.

6. Click **Done**.

SACK Block Generation

TCP performance slows down when multiple packets are lost in one window of data. In such a scenario, a Selective Acknowledgement (SACK) mechanism combined with a selective repeat retransmission policy overcomes this limitation. For every incoming out-of-order packet, you must generate a SACK block.

If the out-of-order packet fits in the reassembly queue block, insert packet info in the block, and set the complete block info as SACK-0. If an out-of-order packet does not fit into reassembly block, send the packet as SACK-0 and repeat the earlier SACK blocks. If an out-of-order packet is a duplicate and packet info is set as SACK-0 then D-SACK the block.

Note: A packet is considered as D-SACK if it is an acknowledged packet, or an out of order packet which is already received.

Client Reneging

A Citrix ADC appliance can handle client reneging during SACK based recovery.

Memory checks for marking end_point on PCB is not considering total available memory

In a Citrix ADC appliance, if the memory usage threshold is set to 75 percent instead of using the total available memory, it causes new TCP connections to bypass TCP optimization.

Unnecessary retransmissions due to missing SACK blocks

In a non-endpoint mode, when you send DUPACKS, if SACK blocks are missing for few out of order packets, triggers additional retransmissions from the server.

SNMP for number of connections bypassed optimization because of overload

The following SNMP ids have been added to a Citrix ADC appliance to track number of connections bypassed TCP optimization due to overload.

1. 1.3.6.1.4.1.5951.4.1.1.46.13 (tcpOptimizationEnabled). To track the total number of connections enabled with TCP optimization.
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed). To track the total number of connections bypassed TCP Optimization.

Dynamic Receive Buffer

To maximize TCP performance, a Citrix ADC appliance can now dynamically adjust the TCP receive buffer size.

Troubleshooting Guidelines

September 14, 2021

Technical Support

All troubleshooting and escalation queries require a recent Citrix ADC techsupport bundle, which captures current configuration, firmware version installed, log files, outstanding cores, and others.

Example:

```
show techsupport

showtechsupport data collector tool - $Revision: #5 $!
...
<!--NeedCopy-->
```

All the data will be collected under

```
...
Archiving all the data into "/var/tmp/support/collector_P_192
   .168.121.117_18Jun2015_09_53.tar.gz" ....
Created a symbolic link for the archive with /var/tmp/support/support.
   tgz
/var/tmp/support/support.tgz  ---- points to ---> /var/tmp/support/
   collector_P_192.168.121.117_18Jun2015_09_53.tar.gz
<!--NeedCopy-->
```

After a techsupport bundle has been generated, it might be copied using SCP.

Traces

Citrix ADC TCP optimization issues normally require Citrix ADC traces to troubleshoot properly. Note that one should try to capture traces under similar conditions, i.e. on the same cell, during the same time of day, using the same user equipment and application, and others.

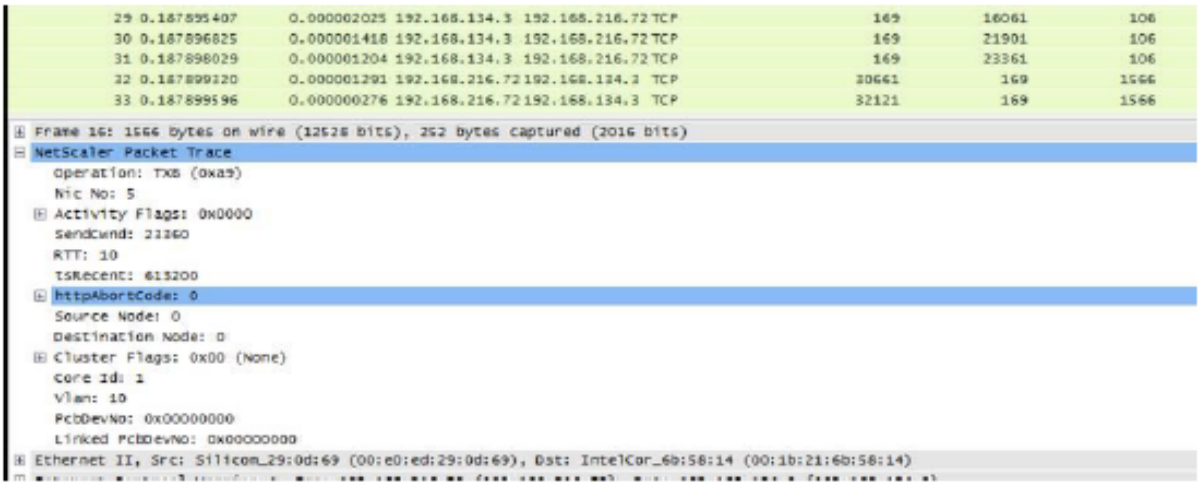
The start nstrace and stop nstrace commands might be used to capture traces:

- It's strongly recommended that the appropriate filter is used to avoid capturing extraneous, unnecessary packets on the trace. For instance use start nstrace -filter 'IP == 10.20.30.40' to only capture packets being sent to or received from IP address 10.20.30.40, which is the user equipment IP address.
- Do not use the -tcpdump option, since it strips the nstrace headers which are required for debugging.

Trace Analysis

After a Citrix ADC trace has been captured, it might be viewed with Wireshark 1.12 or later. Verify that the captured traces include the appropriate Citrix ADC Packet Trace headers, as shown in the screen capture below:

The additional debug headers are also visible per the illustration below:



Connection Table

When the issue is related to TCP optimization and it can be reproduced or it's on-going, it is best to get also the connection table when the issue occurs from the primary T1 node.

To get the table you shall need to switch to the BSD shell and run the following command:


```
shell
...

nscli -U 127.0.0.1:nsroot:nsroot show connectiontable -detail full link
    > /var/tmp/contable.log
<!--NeedCopy-->
```

> **Note**
>
> The command might be executed for a longer time and management CPU might be stressed at that time (depends on the number of connection table entries), but it's not service affecting.

Frequently Asked Questions

September 14, 2021

Timeouts

> **Important**
>
> Before using *any nsapimgr* knob, consult with Citrix Customer Support.

The following is a list of different idle connection timeouts that can be set on Citrix ADC T1 virtual servers and services. Idle timeout set for client or server connections at the vserver or service level are applicable only for the connections in TCP ESTABLISHED state and are idle.

- Load Balancing virtual server cltTimeout parameter specifies the time in seconds that a connection from a client to a Load Balancing virtual server must be idle, before the appliance closes the connection.
- Service svrTimeout parameter specifies the time in seconds that a connection from the appliance to a service or server must be idle, before the appliance closes the connection.
- Service cltTimeout parameter specifies the time in seconds that a connection from a client to a service must be idle, before the appliance closes the connection.

When a service is bound to a Load Balancing virtual server, then the cltTimeout for the Load Balancing virtual server takes precedence, and the service cltTimeout for service is ignored.

In case of there is not service bound to Load Balancing virtual server, global idle timeout, namely tcpServer, is used for server side connections. It can be configured as follows:

Command:

```
set ns timeout  - tcpServer 9000
```



```
<!--NeedCopy-->
```

Connections in other state have different timeout values:

- Half open connections idle timeout: 120 seconds (hardcoded value)
- TIME_WAIT connections idle timeout: 40 seconds (hardcoded value)
- Half close connections idle timeout. By default it is 10s and can be configured between 1s and 600s using the snippet

Command:

```
 set ns timeout -halfclose 10
<!--NeedCopy-->
```

When half-close timeout is triggered, connection is moved to zombie state. When zombie timeout expires, zombie cleanup kicks in and T1 sends RST on both client and server side for given connection by default.

- Zombie timeout: Interval at which the zombie cleanup process must run to clean up inactive TCP connections. Default timeout value is 120s and can be configured between 1s and 600s.

Command:

```
set ns timeout -zombie 120
<!--NeedCopy-->
```

Maximum Segment Size Table

A Citrix ADC T1 appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted. Specific function is enabled by default without option to disable.

However, there is caveat as standard SYN cookies limit connections to the use of only eight Maximum Segment Size (MSS) values. If connection MMS does not match with any predefined value, it will pick up the next available lower value towards both client and server side.

The predefined TCP Maximum Segment Size (MSS) values are the following and can be configured through a new nsapimgr knob.

1460	1440	1330	1220	956	536	384	128

The new MSS table:

- Need not contain Jumbo-Frame support. Even though by default 8 values are reserved in the MSS table for jumbo frames, the table settings can be modified to include standard Ethernet-sized frames only.
- Should have 16 values
- Should have values in descending order
- Should include 128 as the last value

If the new MSS table is valid, the table is stored and the old values are switched out at the SYN-cookie rotation time. Otherwise the new table returns an error. Changes are applied to new connections while existing connections preserve the old MSS table until the connections expire or are terminated.

To display the current MSS table in a Citrix ADC appliance, type the following command.

Command:

```
>shell

#nsapimgr -d mss_table
```

Example:

```
#nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1440,1330,1212,956,536,384,128
     }


Done.
```

To change the mss table, type the following command:

Command:

```
>shell

#nsapimgr -s mss_table=<16 comma seperated values>
```


Example:

```
#nsapimgr -ys mss_table
   =9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128
     }


Done.
```

An example using standard Ethernet-sized values is depicted below:

Example:

```
#nsapimgr -ys mss_table
   =1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128
     }


Done.
```

To make this change permanent even after the Citrix ADC appliance restarts, include the command `##nsapimgr -ys mss_table=<16 comma seperated values>` in the "/nsconfig/rc.netscaler" file. If the "rc.netscaler" file doesn't exist, create it under the "/nsconfig" folder, and then append the command.

Memory Overload Protection

A Citrix ADC Packet Processing Engine (PPE) starts bypassing connections from TCP optimization if the memory in use by that one PPE is more than a specified high watermark value. If a PPE memory

utilization goes above ~2.6GB, then it starts bypassing *any new* connections from optimization. The existing connections (ones admitted for optimization previously) continues getting optimization. This watermark value has been purposefully selected and is not recommended for tuning.

> **Note**
>
> If you believe that there is a good reason to change that watermark value, contact Customer Support.

Support for Happy Eyeballs Clients

If the Citrix ADC appliance receives a SYN for a destination for which the state is unknown, the appliance first checks the reachability of the server and then acknowledges the client. This probing mechanism enables clients with dual IP stacks to discover the reachability of dual-stack internet servers. If the client discovers that both IPv6 and IPv4 access are available, it establishes a connection to the server that responds more quickly, and resets the other. For the connection for the Citrix ADC appliance receives a reset, it will reset the corresponding server side connection.

Note: This feature has no user configurable TCP settings to be disabled/enabled on the Citrix ADC appliance.

For more information on Happy Eyeballs support, see RFC 6555.

Citrix ADC Video Optimization

September 14, 2021

The Citrix ADC appliance provides optimization techniques and capabilities to optimization ABR video traffic for video traffic over mobile networks. This improves user experience and reduces the overall network bandwidth consumption.

The section includes the following topics:

- Getting Started
- Licensing
- Configuring Video Optimization over TCP
- Configuring Video Optimization over UDP

Getting Started

September 14, 2021

Media files have been driving an increasing amount of traffic over mobile networks, and migration to faster networking technologies has dramatically increased the volume of encrypted video traffic. The traditional media delivery technology (Progressive Download) is failing to deliver acceptable quality of experience (QoE) at a high transmission rate. This has led to the introduction of the Adaptive Bit Rate (ABR) protocol. It can adapt the streaming bit rate to the available network bandwidth and restrict streaming quality to match the capability of the handset receiving the video. However, the ABR protocol does not work as well in mobile networks as it does over the internet. Mobile operators must, therefore, optimize ABR traffic.

A Citrix ADC appliance has unique capabilities to detect incoming video traffic and selectively optimize ABR videos.

How Citrix ADC Video Optimization Works

A Citrix ADC appliance can identify and optimize encrypted ABR traffic (including Facebook video traffic) over TCP, and YouTube ABR traffic over QUIC. The appliance has the following capabilities:

1. Detect Progressive Download (PD) videos over HTTP.
2. Detect and optimize ABR videos over HTTP.
3. Detect and optimize ABR videos over HTTPS.
4. Detect and optimize YouTube ABR videos over QUIC.

Also, the appliance uses the following support domains for detecting video traffic over TCP and QUIC protocols.

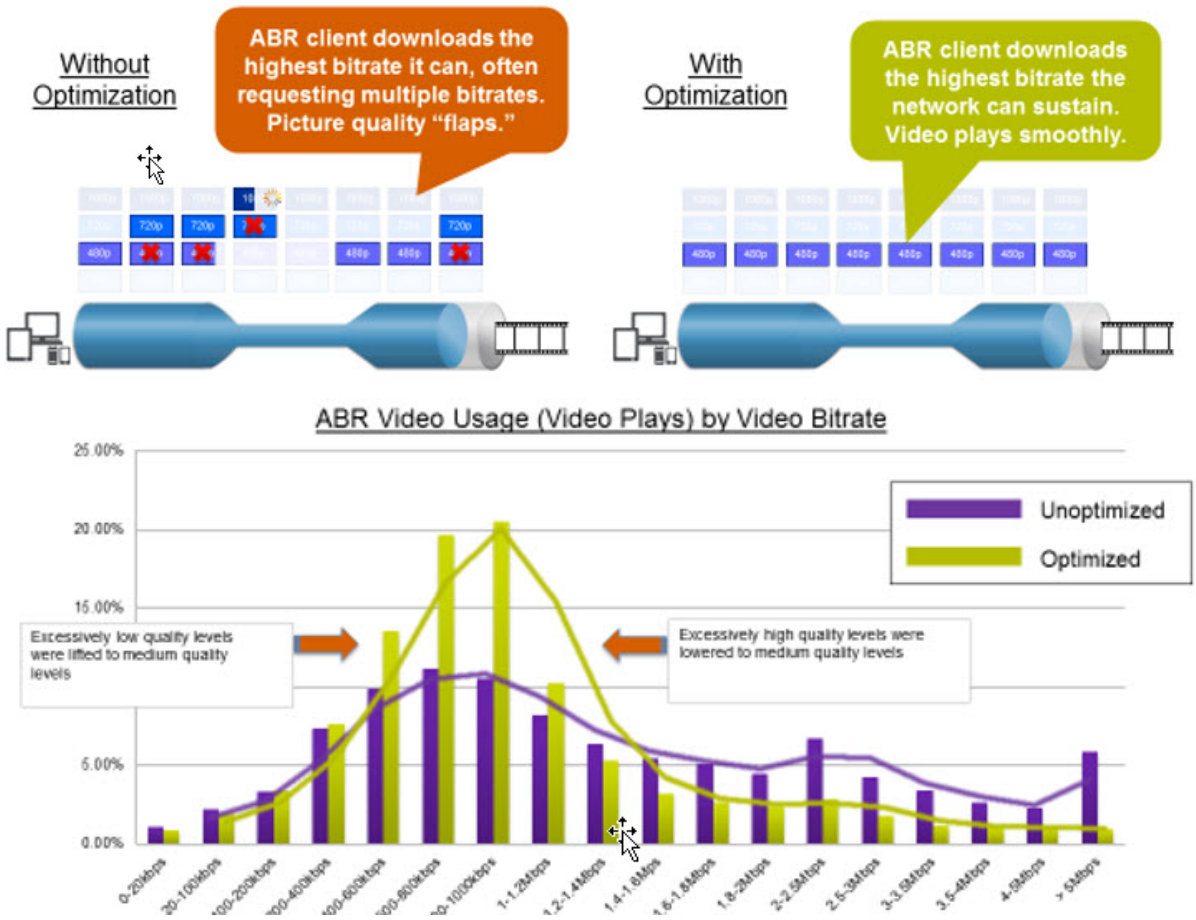
- Unencrypted ABR videos over TCP. Appliance detects all standard compliant video streaming web sites. The appliance detects ABR sessions by inspecting the response video payload header, URL, and HTTP headers.
- Encrypted ABR video over TCP. Appliance detects ABR sessions using a generic and heuristic algorithm based on domain, SSL header and traffic patterns. Using this, the appliance has a built-in support to detect top video web sites, with 95 percent accuracy and we continue to add support for new video types. Citrix ADC also has a program to provide additional verification for top encrypted ABR sites for a region or country to ensure network coverage.
- Encrypted ABR videos over QUIC. The appliance detects ABR sessions for QUIC based video provider, such as YouTube. The detection algorithm is on the basis of a heuristic leveraging the QUIC headers and domain. Citrix ADC will continue to add support for newer video sites using QUIC.

Benefits

Optimizing the ABR video traffic can provide the following benefits:

- Manage the network during congestion in peak hours.

- Improve video play consistency and reduce video stalling.
- Enable new video service offerings (for example, Binge-on video services).
- Enable customers to select the best sustainable video quality.
- Provide a consistent user experience for the subscriber.



Video Optimization over TCP

Citrix ADC optimization of ABR traffic over TCP works as follows:

1. HTTP or HTTPS traffic that the appliance receives over TCP is sent to the corresponding load balancing virtual server.
2. The built-in detection policies bound to the virtual server combined with other proprietary detection algorithms evaluate the traffic.
3. The policies use a set of built-in video detection signatures to detect the video type. The policy that matches traffic applies an action that categorizes the video type as one of the following:
   a) Clear-text PD
   b) Clear-text ABR
   c) Encrypted ABR

d) Other

4. The optimization policies bound to the same virtual server evaluate the traffic and determine the optimization bit rate to apply to the traffic.
5. The optimization bit rate is applied if the traffic is either clear-text ABR or encrypted ABR.

A mobile service provider can improve the quality of experience (QoE) by setting the download speed for 2G, 3G, and 4G mobile traffic. This reduces the video start times or buffering events. Optimization can also reduce the amount of network bandwidth consumed by video sessions.

The optimization techniques include dynamic burst control and random sampling.

Dynamic Burst Control

Citrix ADC ABR optimization adapts dynamically to changing network conditions. It allows an initial burst rate of 1.3 times the configured pacing rate for 15 seconds. The initial burst rate applies to the beginning of every optimized ABR video session, even when multiple sessions use the same TCP connection or group of TCP connections.

The appliance also supports recovery bursts in the event that the bit rate supported by the network drops below the configured pacing rate. For example, if the effective bit rate drops at the 7th second and recovers at 15th second of the initial burst, the appliance recovers the loss during the next burst cycle. By doing so, the appliance dynamically optimizes the network bandwidth for all subscribers so that the quality of video remains consistent per pixel.

Note: When a recovery burst happens during an initial burst, the pacing bit rate must not exceed the maximum recovery-burst and initial-burst rates (you must not add the Recovery Burst factor on top of the Initial Burst factor). Otherwise, it might be so fast that the media player shifts to a higher quality mode. However, if necessary, you can extend the duration of the Initial Burst to compensate the unused bandwidth.

Random Sampling

To estimate the savings from video optimization, the Citrix ADC appliance implements random sampling. With this technique, the appliance randomly selects a configurable percentage of the detected video traffic (the random sampling parameter is an integer number ranging from 0 to 100, so less than 1 percent is not possible). These randomly selected and non-optimized transactions (and sessions) become a reference group, and they are identified in the transaction logs (along with other characteristics, such as byte size and timer fields. The characteristics of the optimized sessions are also logged, and the reporting engine compares statistics of the optimized and reference groups to estimate the savings from optimization (including the savings from ABR Optimization).

Video Optimization over UDP

Google has introduced a new transport protocol called QUIC. Google's QUIC protocol is very similar to TCP+TLS+HTTP/2 and is implemented on top of UDP. Citrix ADC can detect YouTube ABR videos streamed over the QUIC protocol, and apply ABR video optimization in a similar way as ABR over TCP.

Licensing

September 14, 2021

The Video Optimization feature works on Telco platforms with the purchase of a basic CBM license and CBM Premium license and for other Citrix ADC platforms, the feature works with the purchase of a CNS Premium license. Before you configure the video optimization feature, your appliance must have a suitable license.

License support for Telco platforms:

- **CBM_TXXX_SERVER_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **CNS_WEBF_SSERVER_Retail.lic**

Where XXX is the throughput, for example, Citrix ADC T1000.

License support for other Citrix ADC platforms:

- **CNS_XXX_SERVER_PLT_Retail.lic**

Where XXX is the throughput.

To upload a Premium license file, follow the steps given below:

1. A valid license file should be installed on the Citrix ADC appliance. The license should support at least as many Gbps as the expected maximum Gi-LAN throughput.

   License files should be copied through an SCP client to the /nsconfig/license of the appliance, as shown in the screen capture below.

   ```
   > shell ls /nsconfig/license/
   CNS_V3000_SERVER_PLT_Retail.lic ssl
   <!--NeedCopy-->
   ```

2. Do a warm restart to apply for the new license, as shown in the screen capture below.

   ```
   > reboot -warm
   Are you sure you want to restart NetScaler (Y/N)? [N]:y
    Done
   <!--NeedCopy-->
   ```


3. After the restart completes, verify that the license has been properly applied, by using the show license CLI.

In the example below a Premium license with Premium edition has been successfully installed.

```
    > show license

                License status:

                                        Video Optimization: YES

                                                    ...

                                    Model Number ID: 110050

                                       License Type: Premium License
<!--NeedCopy-->
```

Configuring Video Optimization over TCP

September 21, 2021

> **Warning:**
>
> As part of video optimization, the video pacing functionality is deprecated and will be removed from the Citrix ADC appliance in forthcoming releases.

To optimize video traffic over TCP, begin by enabling the video optimization feature. The appliance then activates the built-in detection policies to detect the incoming video traffic and identify the type of video. User configurable optimization policies for each video type specify the optimization bit rate needed for optimizing the traffic.

Configuring Video Optimization over TCP by using the CLI

To configure video optimization on a Citrix ADC appliance, you perform the following tasks:

1. Enable the video optimization feature.
2. Add virtual servers for HTTP and HTTPS traffic.
3. Bind all the built-in detection policies to a load balancing virtual server for HTTP traffic.
4. Bind all the built-in detection policies to an SSL-bridge load balancing virtual server for HTTPS traffic.
5. Add the desired optimization policies for HTTP and HTTPS traffic.

6. Bind optimization policies to a load balancing virtual server for HTTP Traffic.
7. Bind optimization policies to an SSL-bridge load balancing virtual server for HTTPS traffic.

Enabling Video Optimization

If you want the Citrix ADC appliance to detect, optimize, and report video traffic, you must enable the Video Optimization feature and set optimization to ON. After enabling the feature you can use built-in detection policies to identify the incoming video traffic, and you can configure optimization policies to optimize encrypted ABR traffic. To optimize ABR video traffic, you must configure the download bit rate (also called the *pacing rate*).

You must also enable the load balancing feature, and if you want to use video optimization for HTTPS traffic you must enable the SSL feature.

To enable the video optimization feature

At the command prompt, type the following command:

```
enable ns feature VideoOptimization
<!--NeedCopy-->
```

> **Note**
>
> If you want to monitor the video optimization performance and video insight reports you must enable the AppFlow feature and then access the Video Analytics feature on Citrix Application Delivery Management (ADM). For more information, see Video Insight documentation.

Creating Virtual Servers for HTTP and HTTPS Video Traffic

A Citrix ADC appliance uses different virtual servers for detecting and optimizing the different types of incoming video traffic. The appliance supports the following types of virtual servers for TCP traffic.

- **HTTP Load Balancing virtual server**. For detecting HTTP video traffic, the appliance uses an HTTP load balancing virtual server. It manages HTTP video requests that the appliance receives from clients.
- **SSL-Bridge Load Balancing virtual server**. To detect encrypted video traffic, you must configure an SSL bridge virtual server on the appliance.

To add an HTTP Load Balancing virtual server for detecting HTTP video traffic

At the command prompt, type the following:


```
add lb vserver <name> HTTP * 80 -persistenceType NONE
<!--NeedCopy-->
```

Example:

```
add lb vserver ProxyVserver-HTTP HTTP * 80 -persistenceType NONE -
    cltTimeout 120
<!--NeedCopy-->
```

To add an SSL Bridge virtual server for detecting HTTPS video traffic

At the command prompt, type the following:

```
add lb vserver <name> SSL_BRIDGE * 443 -persistenceType NONE
<!--NeedCopy-->
```

Example:

```
add lb vserver ProxyVserver-SSL SSL_BRIDGE * 443 -persistenceType NONE
    -cltTimeout 180
<!--NeedCopy-->
```

Binding Built-In Detection Policies to an HTTP Load Balancing Virtual Server

To detect video traffic over an HTTP connection, you must bind all the built-in detection policies to a load balancing virtual server. You must bind the policies to either request-time or response-time processing, depending on the policy type.

> **Note:**
>
> The `ns_videoopt_http_body_detection` video optimization policy does not support the `CONNECT` HTTP request method.

To bind detection policies for different video types to an HTTP load balancing virtual server

At the command prompt, type the appropriate command for each type. The available commands are:

```
bind lb vserver <name> -policyName ns_videoopt_http_abr_netflix -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_netflix2 -
    priority <integer> -type (REQUEST | RESPONSE)

```



```
bind lb vserver <name>   -policyName ns_videoopt_http_abr_youtube -priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube -priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube2  -priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>   -policyName ns_videoopt_http_pd_youtube3 -priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>  -policyName ns_videoopt_http_abr_generic -priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

Example:

```
bind lb vserver ProxyVserver-HTTP    -policyName ns_videoopt_http_abr_netflix -priority 400  type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName ns_videoopt_http_abr_netflix2 -priority 500 -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName ns_videoopt_http_abr_youtube -priority 600  -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName ns_videoopt_http_pd_youtube -priority 800  -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName ns_videoopt_http_pd_youtube2 -priority 900 -type RESPONSE

bind lb vserver  ProxyVserver-HTTP   -policyName ns_videoopt_http_pd_youtube3 -priority 1000 -type REQUEST

bind lb vserver  ProxyVserver-HTTP   -policyName ns_videoopt_http_abr_generic -priority 1100 -type RESPONSE
<!--NeedCopy-->
```

Binding the HTTP Body Content Detection Policy to Load Balancing Virtual Server

To detect video traffic over HTTP, you must bind the body content detection policy to the load balancing virtual server. You can use the following command:


```
bind lb vserver <name> -policyName ns_videoopt_http_body_detection -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

Example:

```
bind lb vserver ProxyVserver-HTTP -policyName
    ns_videoopt_http_body_detection -priority 1500 -type REQUEST
<!--NeedCopy-->
```

Binding Built-In Detection Policies to an SSL-Bridge Load Balancing Virtual Server

To detect video traffic over an HTTPS connection, you must bind built-in detection policies to an SSL Bridge load balancing virtual server.

To bind a detection policy to an SSL bridge load balancing virtual server

At the command prompt, type the appropriate command for each type. The available commands are:

```
bind lb vserver <name> -policyName ns_videoopt_https_abr_netflix -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_youtube -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_generic -
    priority <positive_integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

Example:

```
bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_netflix -priority 120 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_youtube -priority 140 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_generic -priority 150 -type REQUEST
<!--NeedCopy-->
```


Adding Optimization Policies for Pacing ABR traffic

To optimize ABR traffic, you have to configure optimization policies and the associated actions. You then bind the policies to the same load balancing virtual servers to which you bound the detection policies. For each policy, create the action first, so that you can include it when you create the policy.

To add an optimization action

At the command prompt, type:

```
add videooptimization pacingaction <action Name> -rate <integer> [-
   comment <string>]
<!--NeedCopy-->
```

Where the **rate** parameter specifies the rate in Kbps at which to send the traffic (the pacing rate).

Example:

```
add videooptimization pacingaction MyOptAct2000 -rate 2000
<!--NeedCopy-->
```

To add an optimization policy

At the command prompt, type:

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
   string>
<!--NeedCopy-->
```

Example:

```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
   MyOptAct2000
<!--NeedCopy-->
```

Binding Optimization Policies to an HTTP Load Balancing Virtual Server

To optimize ABR video traffic over an HTTP connection, you must bind the optimization policies to a load balancing virtual server to which the detection policies are bound.

To bind an optimization policy to a Load Balancing virtual server

At the command prompt, type the following command:


```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type  (REQUEST | RESPONSE)
<!--NeedCopy-->
```

Example:

```
bind lb vserver ProxyVserver-HTTP -policyName myOptPolicy2000 -priority
     3400 -type  REQUEST
<!--NeedCopy-->
```

Binding Optimization Policies to SSL-bridge Virtual Servers

To optimize ABR video traffic over an HTTPS connection, you must bind the optimization policies to the SSL Bridge virtual server to which the built-in detection policies are bound.

To bind an optimization policy to SSL Bridge virtual server for pacing encrypted traffic

At the command prompt, type the following command:

```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type  (REQUEST |RESPONSE)
<!--NeedCopy-->
```

Example:

```
bind lb vserver ProxyVserver-SSL -policyName myOptPolicy2000 -priority
    3400 -type  REQUEST
<!--NeedCopy-->
```

Setting video optimization pacing parameters

The CLI enables you to set the video optimization pacing parameters, such as random sampling percentage.

To set the random sampling percentage

At the command prompt, type the following command:

```
set videooptimization parameter  - RandomSamplingPercentage <realNumber>
<!--NeedCopy-->
```


Where, a realNumber is a value from 0.0 to 100.0.

Example:

```
set videooptimization parameter -RandomSamplingPercentage 50
<!--NeedCopy-->
```

Configuring Video Optimization over TCP by using the GUI

The GUI enables you to:

- Enable video optimization feature.
- Create HTTP load balancing virtual server.
- Create SSL-bridge load balancing virtual server.
- Bind built-in detection policies to HTTP load balancing virtual server.
- Bind built-in detection policies to SSL-bridge load balancing virtual server.
- Create an optimization policy.
- Create an optimization action.
- Configuring optimization pacing parameter.
- Bind optimization policy to load balancing virtual server for HTTP traffic.
- Bind optimization policy to SSL-bridge load balancing virtual server for HTTPS traffic.

To enable video optimization feature

1. In the navigation pane, expand **System**, and then click **Settings**.
2. On the **Settings** page, click the **Configure Advanced Features** link.
3. On the **Configure Advanced Features** page, select the **Video Optimization** check box.
4. Click **OK**, and then click **Close**.

To create load balancing virtual server for HTTP traffic

1. Sign in to the Citrix ADC appliance and navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. In the details pane, click **Add**.
3. On the Load Balancing Virtual Server screen, set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. Select protocol type as HTTP
   c) **IP Address Type**. IP address type: IPv4 or IPv6.
   d) **IP Address**. IPv4 or IPv6 address assigned to the virtual server.
   e) **Port**. Port number of the virtual server.

4. Click **OK** to continue with configuration of other, optional, parameters. For more information, see Creating a Virtual Server.
5. Click **Create** and **Close**.

To create load balancing virtual server for HTTPS traffic

1. Sign in to the Citrix ADC appliance and navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. In the details pane, click **Add**.
3. On the **Load Balancing Virtual Server** screen, set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. Select protocol type as SSL-bridge.
   c) **IP Address Type**. IP address type: IPv4 or IPv6.
   d) **IP Address**. IPv4 or IPv6 address assigned to the virtual server.
   e) **Port**. Port number of the virtual server.
4. Click **OK** to continue with the configuration of other, optional, parameters. For more information, see Creating a Virtual Server.
5. Click **Create** and then **Close**.

To bind a built-in detection policy to a load balancing virtual server

1. Sign in to the Citrix ADC appliance and navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** screen.
2. In the details pane, select the load balancing virtual server and click **Edit**.
   a) In the **Advanced Setting** section, click **Policies**.
   b) In the **Policies** section, click the **+** icon to access the **Policies** slider.
   c) In the **Policies** section, set the following parameters.
   d) Choose Policy. Select a video optimization detection policy from the drop-down list.
   e) Choose Type. Select the policy type as Request.
   f) Click **Continue**.
3. Select the video detection policy from the list and click **Close**.

To bind a built-in detection policy to a SSL-bridge load balancing virtual server

1. Log on to the Citrix ADC appliance and navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** screen.
2. In the details pane, select the SSL-bridge load balancing virtual server and click **Edit**.
3. In the **Advanced Setting** section, click **Policies**.
4. In the **Policies** section, click the **+** icon to access the **Policies** slider.
5. In the **Policies** section, set the following parameters.

a) Choose Policy. Select video optimization detection policy from the drop-down list.
b) Choose Type. Select the policy type as Request.

6. Click **Continue**.
7. Select the video detection policy from the list and click **Close**.

To create a video optimization action

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization** > **Pacing** > **Actions**.
2. In the details pane, click **Add**.
3. On the **Create Video Optimization Pacing Action** page, set the following parameters.
   a) **Name**. Name of the optimization action.
   b) **ABR Optimization Rate (Kbps).** Pacing rate at which to send the ABR video traffic. The default rate for ABR optimization is 1000 Kbps. The minimum value is 1, and the maximum value is 2147483647.
   c) **Comment**. A short description of the action.
4. Click **Create** and **Close**.

To create a video optimization policy

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization** > **Pacing** > **Policies**.
2. In the details pane, click **Add**.
3. On the **Create Video Optimization Pacing Policy** page, set the following parameters.
   a) **Name**. Name of the optimization policy
   b) **Expression**. Custom regex expressions that implement the policy.
   c) **Action**. Optimization action associated with the policy to handle the incoming video traffic.
   d) **UNDEF Action.** Undefined event if the incoming request does not match the optimization policy.
   e) **Comment**. A short description of the policy.
   f) **Log Action.** Select the audit log action that creates the desired log messages.
4. Click **Create**, and then click **Close.**

To set video optimization pacing parameters

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization.**
2. In the **Video Optimization** page, click **Change Video Optimization Settings** link.
3. In the **Video Optimization Settings** page, set the following parameter.

a) **Random Sampling Percentage (%)**. Percentage of packets selected for random sampling.

4. Click **OK** and **Close**.

To bind a video optimization policy to an HTTP load balancing virtual server

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization**.
2. On the **Video Optimization** page, click the **Video Optimization Pacing Policy Manager** link.
3. Set the following parameters.
   a) **Bind Point**. The point at which to apply the optimization policy during request or response processing.
   b) **Connection Type**. Connection type as Request or Response.
   c) **Virtual Server**. The load balancing virtual server to which to bind the policy.
   d) Click **Continue**.
4. In the **Bind Point** section, do one of the following:
   a) Select a policy from the list.
   b) Click **Add Binding** to access the **Policies Binding** slider.
      i. Select an existing policy or add a new policy.
      ii. Enter binding details and click **Bind**.
5. Click **Close**.

To bind a video optimization policy to an SSL-bridge load balancing virtual server

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization**.
2. On the **Video Optimization** page, click the **Video Optimization Pacing Policy Manager** link.
3. On the **Video Optimization Policy Manager** page, set the following parameters.
   a) Bind Point. The point at which to apply the optimization policy during request/response processing.
   b) Connection Type. Connection type as Request or Response.
   c) Virtual Server. The SSL-bridge load balancing virtual server to which to bind the policy.
4. Click **Continue**.
5. In the **Bind Point** section, do one of the following:
   a) Select a policy binding from the list.
   b) Click **Add Binding** to access the **Policies Binding** slider.
      i. Select an existing policy or add a new policy.
      ii. Enter binding details and click **Bind**.
6. Click **Close**.

Configuring Video Optimization over UDP

September 14, 2021

To optimize QUIC ABR video traffic over UDP, begin by enabling the video optimization feature. After you complete the configuration, the appliance detects QUIC based ABR video traffic and applies the optimization bit rate that is configured on the appliance.

Configuring video optimization for QUIC by using the CLI

To configure video optimization for QUIC video traffic over UDP, you must perform the following tasks:

1. Enable Video Optimization.
2. Create a QUIC service.
3. Create a QUIC load balancing virtual server.
4. Bind QUIC web service to the loading balancing virtual server.
5. Create video optimization policy for pacing QUIC based UDP traffic.
6. Bind optimization policy to a QUIC based load balancing virtual server.

Enabling video optimization for QUIC traffic

If you want the Citrix ADC appliance to detect, optimize, and report video traffic, you must enable the Video Optimization feature and set optimization ON.

> **Note**
>
> If you want to use video optimization for QUIC traffic, you must enable the load balancing and AppFlow features.

To enable the video optimization

At the command prompt, type the following command:

```
enable ns feature VideoOptimization
<!--NeedCopy-->
```

Creating a service for QUIC traffic

A Citrix ADC appliance uses a QUIC service for the load balancing virtual server to connect to the egress router in the static routing mode.

> **Note**
>
> Currently, dynamic routing is not supported.

To create a load balancing web service for QUIC video traffic

At the command prompt, type:

```
add service <name> <router-IP> <serviceType> <port> -usip yes -
   useproxyport  [yes | no]
<!--NeedCopy-->
```

Example:

```
add service svc-quic 10.102.29.200 QUIC 443  - usip yes  - useproxyport
   no

where IP address is the internet router address.
<!--NeedCopy-->
```

Creating a load balancing virtual server for QUIC traffic

A Citrix ADC appliance uses a load balancing virtual server for detecting and optimizing QUIC video traffic over UDP.

To create a load balancing virtual server for QUIC video traffic

At the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port> -m MAC
<!--NeedCopy-->
```

Example:

```
add lb vserver vs-quic QUIC * 443 -persistenceType NONE -m MAC -
   cltTimeout 120
<!--NeedCopy-->
```

Binding a QUIC web service to the load balancing virtual server

After you have created the web services and load balancing virtual server for QUIC traffic, you must bind the services to the virtual server.

To bind a web service to load balancing virtual server for QUIC video traffic

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver vs-quic svc-quic
<!--NeedCopy-->
```

Creating Video Optimization Policy for QUIC based UDP traffic

To optimize QUIC based UDP traffic, you have to configure optimization pacing policies and its actions. You must then, bind the policies to QUIC based load balancing virtual servers. For each policy, create an action first so that you can associate it to the policy.

To add an optimization action

At the command prompt, type:

```
add videooptimization pacingaction <action Name> -rate <integer> [-
   comment <string>]
<!--NeedCopy-->
```

Where, the **rate** parameter specifies the rate in Kbps at which to send the traffic (the pacing rate).

Example:

```
set videooptimization parameter -QUICPacingRate 1000
<!--NeedCopy-->
```

where 1000 represents the desired pacing rate in Kbits/sec.

To add an optimization policy

At the command prompt, type:

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
   string>
<!--NeedCopy-->
```

Example:


```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
   MyOptAct2000
<!--NeedCopy-->
```

Binding Optimization Policies to a QUIC Load Balancing Virtual Server

To optimize QUIC video traffic over a UDP connection, you must bind the optimization policies to a QUIC load balancing virtual server.

To bind an optimization policy to a QUIC Load Balancing virtual server

At the command prompt, type the following command:

```
bind lb vserver <name> -policyName <policy_name> -priority <
   positive_integer> -type (REQUEST)
<!--NeedCopy-->
```

> **Note**
>
> The pacing policies must be bound to a QUIC load balancing virtual server only at request time.

Example:

```
bind lb vserver vs-quic -policyName myOptPolicy2000 -priority 3400 -
   type  REQUEST
<!--NeedCopy-->
```

Configuring video optimization for QUIC by using the GUI

To configure the feature on the appliance through the GUI, you must perform the following tasks:

1. Enable video optimization
2. Configure a QUIC servers
3. Configure QUIC service
4. Configure a QUIC load balancing virtual server
5. Bind the QUIC web service to the load balancing virtual server
6. Create optimization policy.
7. Create optimization action.
8. Configuring optimization pacing parameter.
9. Bind optimization policy to load balancing virtual server for QUIC traffic.

To enable video optimization

1. Log on to the Citrix ADC appliance and navigate to **System** > **Settings.**
2. On the details page, select **Configure Advanced Features** link.
3. On the **Configure Advanced Features** page, select the **Video Optimization** check box.

To create a QUIC servers

1. Log on to the Citrix ADC appliance and navigate to the **Traffic Management** > **Load Balancing** > **Servers** screen.
2. In the details pane, click **Add**.
3. On the **Create Server** page, set the following parameters:
   a) Name. Name of the QUIC server.
   b) IP address. IP address of the QUIC server
   c) Traffic Domain. Domain name of the server.
   d) Enabling after creating. Initial state of the server.
   e) Comments. Brief information about the server.
4. Click **Create.**

To create a QUIC service

1. Log on to the Citrix ADC appliance and navigate to the **Traffic Management** > **Load Balancing** > **Services screen**.
2. In the details pane, click **Add**.
3. On the **Load Balancing Service** page, set the following parameters:
   a) **Service Name**. Name of the QUIC service.
   b) **IP address**. IP address assigned to the QUIC service.
   c) **Protocol**. Select protocol as QUIC.
   d) **Port**. Port number of the web service.
4. Click **OK** to continue. You can then configure other, optional, parameters. For more, see Configuring Services.
5. Once you configure the optional parameters, click **OK** and **Close**.

To create a load balancing virtual server

1. Log on to the Citrix ADC appliance and navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** screen.
2. In the details pane, click **Add**.
3. On the **Load Balancing Virtual Server** page, set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. The protocol used by the service to send QUIC requests.

c) IP Address Type. IP address type: IPv4 or IPv6.
d) **IP Address**. IP 4 or IP6 IP address assigned to the virtual server.
e) **Port**. Port number of the virtual server.

4. Click OK to continue with the configuration of other, optional, parameters. For more information, see Creating a Virtual Server.

To bind a load balancing virtual server to a QUIC service

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and select a virtual server.
2. Click **Services and Service Groups** to access the **Load Balancing Virtual Server Service Binding** screen.
3. Select a QUIC based web service and click **Bind**.
4. Click **Done**.

To bind a load balancing virtual server to a QUIC service

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and select a virtual server.
2. Click **Services and Service Groups** to access the **Load Balancing Virtual Server Service Binding** screen.
3. Select a QUIC based web service and click **Bind**.
4. Click **Done**.

To create a video optimization action for QUIC traffic

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization** > **Pacing** > **Actions**.
2. In the details pane, click **Add**.
3. On the **Create Video Optimization Pacing Action** page, set the following parameters.
   a) **Name**. Name of the optimization action.
   b) **ABR Optimization Rate (Kbps).** Pacing rate at which to send the ABR video traffic. The default rate for ABR optimization is 1000 Kbps. The minimum value is 1, and the maximum value is 2147483647.
   c) **Comment**. A short description of the action.
4. Click **Create** and **Close**.

To create a video optimization policy for QUIC traffic

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization** > **Pacing** > **Policies**.

2. In the details pane, click **Add**.
3. On the **Create Video Optimization Pacing Policy** page, set the following parameters.
   a) Name. Name of the optimization policy
   b) Expression. Custom regrex expressions that implement the policy.
   c) Action. Optimization action associated with the policy to handle the incoming video traffic.
   d) UNDEF Action. Undefined event if the incoming request does not match the optimization policy.
   e) Comment. A short description of the policy.
   f) Log Action. Select the audit log action that creates the desired log messages.
4. Click **Create**, and then click **Close**.

To bind a video optimization policy to an QUIC load balancing virtual server

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization**.
2. On the **Video Optimization** page, click the **Video Optimization Pacing Policy Manager** link.
3. On the **Video Optimization Policy Manager** page, set the following parameters.
   a) Bind Point. The point at which to apply the optimization policy during request processing. **Note**: The pacing policies must be bound to a QUIC load balancing virtual server only at request time.
   b) Connection Type. Connection type as Request or Response.
   c) Virtual Server. The load balancing virtual server to which to bind the policy.
4. Click **Continue**.
5. In the **Bind Point** section, do one of the following:
   a) Select a policy from the list.
   b) Click **Add Binding** to access the **Policies Binding** slider.
      i. Select an existing policy or add a new policy.
      ii. Enter binding details and click **Bind**.
6. Click **Close**.

Citrix ADC URL Filtering

September 14, 2021

URL Filtering provides policy-based control of websites by using the information contained in URLs. This feature helps network administrators monitor and control user access to malicious websites on mobile networks.

As an administrator, you can configure a URL filtering policy by using either the URL Categorization feature or the URL List feature.

URL List. Controls access to blacklisted websites and web pages by blocking access to URLs that are in a URL set imported into the appliance.

URL Categorization. Controls access to websites and web pages by filtering traffic on the basis of a predefined list of categories.

URL List

September 14, 2021

The URL List feature enables you to control access to customized URL lists (up to one million entries). The feature filters websites by applying a URL filtering policy bound to a virtual server.

As an administrator, you must import the URL List into the Citrix ADC appliance. This imported list is internally stored as a Policy data set called a *URL Set*. The appliance then applies a unique fast URL matching algorithm to the incoming URL requests. If the incoming URL request matches an entry in the set, the appliance applies the associated policy action to control access.

URL List Types

Each entry in a URL set can include a URL and, optionally, its metadata (URL category, category groups, or any other related data). For URLs with a metadata, the appliance uses a policy expression that evaluates the metadata. For more information, see URL Sets.

Custom URL List. You can create a customized URL set of up to 1,000,000 URL entries and import it as a text file into your appliance. The list can contain URLs with or without metadata (which could be like a URL category). TheCitrix ADC platform automatically detects whether metadata is present. It also supports storing the imported lists securely. For more information, see URL Set.

You can host the URL list and configure the Citrix ADC appliance to periodically update the list without requiring manual intervention. Once the URL list is updated, the appliance can automatically detect the metadata and the categories by using policy expressions to evaluate each incoming URL and then apply actions such as allow, block, redirect, or notify the user.

URL List Policy Expressions

The following table describes the basic expressions you can use to evaluate incoming traffic. After you import an URL List to the appliance, it is called a *URL Set*.

Expression	Operation
<URL expression>.URLSET_MATCHES_ANY (<URLSET>)	Evaluates to TRUE if the URL exactly matches any entry in the URL set.
<URL expression>. GET_URLSET_METADATA(<URLSET>)	The GET_URLSET_METADATA() expression returns the associated metadata if the URL exactly matches any pattern within the URL set. An empty string is returned if there is no match.
<URL expression>.GET_ URLSET_METADATA(<URLSET>).EQ(< METADATA>)	Evaluates to TRUE if the matched metadata is equal to <METADATA>.
<URLexpression>.GET_URLSET_METADATA (<URLSET>).TYPECAST_LIST_T(‘,’).GET (0).EQ(<CATEGORY>)	Evaluates to TRUE if the matched metadata is at the beginning of the category. This pattern can be used to encode separate fields within metadata, but match only the 1st field.
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	Joins the host and URL parameters, which can then be used as a <URL expression> for matching.

URL List Policy Actions

The most common enforcement action for URLs that match a URL list is to restrict access. Create a URL list policy with a desired URL list matching expression and enforcement action. The policy group usage depends upon the incoming traffic type (HTTP or HTTPS) and the virtual server configured on the appliance. You can use a Responder policy for HTTP traffic or a Video Optimization policy for HTTPS traffic. Specify actions to apply to the URLs that match the expressions in the policies. The following table lists the available actions.

Action Type	Policy	Description
ALLOW	Responder	Allow the request to access the target URL.
REDIRECT	Responder	Redirect the request to the URL specified as the target.
DENY	Responder	Deny the request.
RESET	Responder, VideoOptimization	Reset the connection.

Action Type	Policy	Description
DROP	Responder, VideoOptimization	Drop the connection.

Prerequisites

To configure URL List feature, make sure you have configured the following server.

DNS Server for DNS Requests

You must configure a DNS server if you import a URL Set from a hostname URL.

At the command prompt, type:

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>) [-state (
   ENABLED | DISABLED )] [-type <type>] [-dnsProfileName <string>]
<!--NeedCopy-->
```

Example:

```
add dns nameServer 10.140.50.5
<!--NeedCopy-->
```

Importing a custom URL list

To import a URL set, see URL Set topic.

Configuring a URL List for HTTP traffic

The Citrix ADC appliance supports HTTP and HTTPS traffic. To configure a load balancing virtual server for HTTP traffic and bind URL list policies to the server, do the following:

- Add URL List actions.
- Add URL List policies.
- Add an HTTP load balancing virtual server for HTTP traffic
- Bind the URL List policies to the HTTP load balancing virtual server for HTTP traffic

To add a URL list action

At the command prompt, type the following:


```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
    string>]
<!--NeedCopy-->
```

To add a HTTP load balancing virtual server for HTTP traffic

At the command prompt, type the following:

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
    imeout <secs>]
<!--NeedCopy-->
```

Example:

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout
    120
<!--NeedCopy-->
```

To bind URL list policy to HTTP load balancing virtual server

At the command prompt, type the following:

```
bind lb vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

Configuring URL List for HTTPS traffic

The Citrix ADC appliance supports HTTP and HTTPS traffic. To configure a SSL-bridge load balancing virtual server for HTTPS traffic and bind URL list policies to the server, do the following:

- Add URL List actions.
- Add URL List policies.
- Add a SSL-bridge load balancing virtual server for HTTP traffic
- Bind the URL List policies to the SSL-bridge load balancing virtual server for HTTP traffic

To add a URL List policy for HTTPS traffic

At the command prompt, type:


```
add videooptimization detectionpolicy <name> -rule <expression> -action
     <string> [-undefAction <string>] [-comment <string>] [-logAction <
    string>]
<!--NeedCopy-->
```

To add a SSL-bridge load balancing virtual server

At the command prompt type:

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
    imeout <secs>]
<!--NeedCopy-->
```

Example:

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -
    cltTimeout 180
<!--NeedCopy-->
```

To bind URL List policy with SSL-bridge load balancing by using the CLI

At the command prompt type:

```
bind lb vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

Configuring a URL List by using the GUI

The GUI enables you to:

- Import a URL list.
- Add a URL list.
- Configure URL list actions.
- Configure URL list policies for HTTP traffic.
- Add an HTTP load balancing virtual server for HTTP traffic.
- Add an SSL-bridge load balancing virtual server for HTTPS traffic.
- Bind URL list policies to the HTTP load balancing virtual server.
- Bind a URL list policies to the SSL-bridge load balancing virtual server.

To import a URL list

1. In the navigation pane, expand **AppExpert** > **URL Sets**.
2. In the details pane, click **Import**.
3. On the **Configure URL Set** page, set the following parameters.
   a) **Name**. Name of the URL set.
   b) **URL**. Web address of the location at which to access the URL Set.
   c) **Overwrite**. Overwrite a previously imported URL set.
   d) **Delimiter**. Character sequence that delimits a CSV file record.
   e) **Row Separator**. Row separator used in the CSV file. A single character value is permitted for example "/n".
   f) **Interval**. Interval in seconds, rounded off to the nearest 15 minutes, at which the URL set is updated.
   g) **Private Set.** Option to prevent exporting the URL set
   h) **Canary URL**. Internal URL for testing whether the content of the URL set is to be kept confidential. The maximum length of the URL is 2047 characters
4. Click **Create**, and then **Close**.

To add a URL list

1. In the navigation pane, expand **AppExpert** > **URL Sets**.
2. In the details pane, click **Add**.
3. On the **Create URL Set** page, set the following parameters.
   a) **Name**. The name of the URL set that was given when it was imported.
   b) **Comments**. A short description about the URL set.
4. Click **Create**.

To configure a URL list action

1. Log on to the Citrix ADC appliance and navigate to **Configuration** tab page.
2. In the menu pane, navigate to **AppExpert** > **Responder** > **Actions**.
3. In the details pane, click **Add**.
4. On the **Create Responder Action** page, set the following parameters.
   a) **Name**. Name of the URL List policy action.
   b) **Type**. Select an action type.
   c) **Expression**. Use the expression editor to create the policy expression.
   d) **Comments**. A short description about the policy action.
5. Click **Create** and **Close**.

To configure a URL list policy

1. In the navigation pane, expand **AppExpert** > **Responder** > **Policies**.
2. In the details pane, click **Add**.
3. On the **Create Responder Policy** page, set the following parameters.
   a) **Name**. Name of the URL List policy action.
   b) **Action**. Select the URL List action that you prefer to associate with the policy.
   c) **Log Action**. Select the log action.
   d) **AppFlow**. Select an AppFlow action.
   e) **Expression**. Use the expression editor to create the policy expression.
   f) **Comments**. A short description about the policy.
4. Click **Create** and **Close**.

To add an HTTP load balancing virtual server

1. Navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. In the details pane, click **Add**.
3. On the **Load Balancing Virtual Server** screen, set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. Choose protocol type as HTTP.
   c) **IP Address Type**. IP addressable type.
   d) **IP Address**. IP 4 or IP6 IP address assigned to the virtual server.
   e) **Port**. Port number of the virtual server.
4. Click **OK** to continue with the configuration of other, optional, parameters. For more information, see Creating a Virtual Server.

To bind a URL List policy to the HTTP load balancing virtual server

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** screen.
2. In the details pane, select the load balancing virtual server and click **Edit**.
3. In the **Advanced Setting** section, click **Policies**.
4. In the **Policies** section, click the **+** icon to access the **Policies** slider.
5. In the **Policies** section, set the following parameters.
   a) Choose Policy. Select a URL categorization policy from the drop-down list.
   b) Choose Type. Select the policy type as Request.
6. Click **Continue**.
7. In the Policies page, select the URL List policy from the list and click **Select**.
8. In the **Policies** slider, click **Bind** and **Close**.

To add URL List policy for HTTPS traffic

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization** > **Detection**.
2. On the **Detection** page, click the **Video Optimization Detection Policies** link.
3. On the **Video Optimization Detection Policies** page, click **Add**.
4. On the **Create Video Optimization Detection Policy** page, set the following parameters.
   a) **Name**. Name of the optimization policy
   b) **Expression**. Configure policy using custom expressions.
   c) **Action**. Optimization action associated with the policy to handle the incoming video traffic.
   d) **UNDEF Action**. Undefined event if the incoming request does not match the optimization policy.
   e) **Comment**. A short description of the policy.
   f) **Log Action**. Select an audit log action that specifies the action to be performed for the log messages.
5. Click **Create** and **Close**.

To add a SSL-bridge load balancing virtual server for HTTPS traffic

1. Navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. In the details pane, click **Add**.
3. On the **Load Balancing Virtual Server** screen, set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. Select protocol type as SSL-bridge.
   c) **IP Address Type**. IP address type: IPv4 or IPv6.
   d) **IP Address**. IPv4 or IP6vIP address assigned to the virtual server.
   e) **Port**. Port number of the virtual server.
4. Click **OK** to continue with the configuration of other, optional, parameters. For more information, see “Creating a Virtual Server” topic.

To bind a URL List Policy to the SSL-bridge load balancing virtual server

1. Navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** screen.
2. In the details pane, select the SSL-bridge load balancing virtual server and click **Edit**.
3. In the **Advanced Setting** section, click **Policies**.
4. In the **Policies** section, click the **+** icon to access the **Policies** slider.
5. Set the following parameters.
   a) **Choose Policy.** Select video detection policy from the drop-down list.
   b) **Choose Type**. Select the policy type as Request.

6. Click **Continue**.
7. Select the video detection policy from the list and click **Close**.

Configuring Audit Log Messaging

Audit logging enables you to review a condition or a situation in any phase of URL List process. When a Citrix ADC appliance receives an incoming URL, if the responder policy has an URL Set advanced policy expression, the audit log feature collects URL Set information in the URL and stores the details as a log message for any target allowed by audit logging.

The log message contains the following information:

1. Timestamp.
2. Log message type.
3. The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency).
4. Log message information, such as URL set name, policy action, URL.

To configure audit logging for URL List feature, you must complete the following tasks:

1. Enable Audit Log.
2. Create Audit Log message action.
3. Set URL List responder policy with Audit Log message action.

For more information, see Audit Logging.

URL List Semantics

The following table lists the URL Match patterns and describes how the URLs within a URL list are matched against the incoming-request URLs. For example, the pattern www.example.com/bar matches only with one page at www.example.com/bar. To match all the pages whose URL starts with ‘ www.example.com/bar’, you would add an asterisk (*) to the end of the URL.

Semantics	URL Pattern	Matched	Unmatched
Subdomain matching	domain.com	domain.com; www.domain.com; sub.one.domain.com	yourdomain.com; wwwdomain.com
URL matching, exact path	domain.com/example/bar/index.html	domain.com/example/bar/index.html; www.domain.com/example/bar/index.html; sub.domain.com/example/bar/index.html	www.domain.com/example/bar/index.html/example/bar/index.html/c

Semantics	URL Pattern	Matched	Unmatched
URL matching, exact path	domain.com/example/I	domain.com/example/I www.domain.com/exar s.domain.com/example	wwwdomaincom/examp do- main.com/example/bar/index.html/
URL matching, subpath matching	domain.com/example/bar/	domain.com/example/bar/ www.domain.com/ example/bar/ index.html; do- main.com/example/bar/index.html/one.jpg	www.domain.com/example/bar/index

URL Categorization

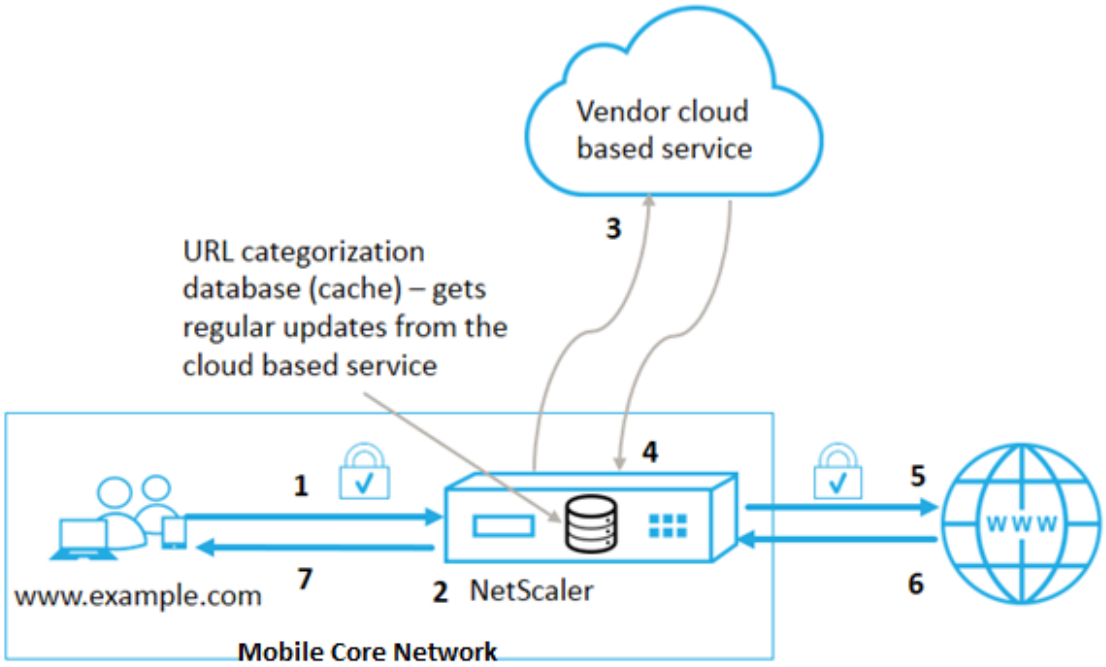
September 14, 2021

URL Categorization restricts user access to specific websites and website categories. As a subscribed service in collaboration with NetSTAR, the feature enables enterprise customers to filter web traffic using a commercial categorization database. The NetSTAR database has a vast number (billions) of URLs classified into different categories, such as social networking, gambling, adult content, new media, and shopping. In addition to categorization, each URL has a reputation score kept up to date based on the site's historical risk profile. We can use NetSTAR data to filter the traffic by configuring advanced policies based on categories, category groups (such as Terrorism, Illegal drugs), or site-reputation scores.

For example, you might block access to dangerous sites, such as sites infected with malware, or selectively restrict access to adult content or entertainment streaming media.

How URL Categorization Works

The following figure shows how the Citrix ADC URL Filtering service is integrated with a commercial URL Categorization database and cloud services for frequent updates.



The components interact as follows:

1. Client sends internet bound URL request.
2. A Citrix ADC policy attempts to evaluate the request in terms of categorization details (such as category, category group, and site-reputation score) retrieved from the URL categorization database. If the database returns the category details, the process jumps to step 5.
3. If the database does not return categorization details, the request is sent to a cloud-based lookup service maintained by a URL categorization vendor. However, the appliance does not wait for a response. Instead, it marks the URL as Uncategorized and jumps to step 5. However, it continues to monitor the cloud query feedback and uses it to update the cache so that future requests can benefit from the cloud lookup.
4. The Citrix ADC appliance receives the URL category details (category, category group, and reputation score) from the cloud-based service and stores it in the cloud cache.
5. If the policy allows the URL, the request is sent to the origin server. Otherwise, the appliance drops or redirects the request, or responds with a custom HTML page.
6. The origin server responds with the requested data to the Citrix ADC appliance.
7. The appliance sends the response to the client.

You can use the URL filtering feature to detect sites that violate safe internet usage mandates issued by the government and implement policies to block these sites. Sites that host adult content, streaming media, or social networking identified as unsafe for children or banned as illegal.

Prerequisites

The feature works on Telco platforms with the purchase of a basic CBM license and CBM Premium license and for other Citrix ADC platforms, the feature works with the purchase of a CNS Premium license.

Note: In addition to a Basic CBM license and a CBM Premium license, the appliance must have a URL Threat Intelligence license with a subscription service for 1 year or 3 years. Before enabling and configuring the feature, you must install the following licenses:

License support for Telco platforms:

- **CBM_TXXX_SERVER_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **CNS_WEBF_SSERVER_Retail.lic**

Where XXX is the throughput, for example, Citrix ADC T1000.

License support for other Citrix ADC platforms:

- **CNS_XXX_SERVER_PLT_Retail.lic**

Where XXX is the throughput.

URL Categorization Policy Expressions

The following table lists the different URL categorization policy expressions for identifying incoming URLs and applies a configured action.

Expression	Operation
<text>. URL_CATEGORIZE (< min_reputation>, <max_reputation>)	Returns a URL_CATEGORY object. Reputation score is a number from 1 to 4. To get objects all reputation scores use 0.0 as <min reputation> and . If is greater than 0, the returned object does not contain a category with reputation lower than . If is greater than 0, the returned object does not contain a category with reputation higher than . If the category fails to resolve in a timely manner, the undef value is returned.
<url_category>. CATEGORY	Returns the category string for this object. If the URL does not have a category, or if the URL is malformed, the returned value is “Uncategorized”.

Expression	Operation
<url_category>. GROUP	Returns a string identifying the object's category group. This is a higher level grouping of categories, which is useful in operations that require less detailed information about the URL category. If the URL does not have a category, or if the URL is malformed, the returned value is "Uncategorized".
<url_category>. REPUTATION	Returns the reputation score as a number from 1 to 4, where 4 indicates the riskiest reputation. If the category is "Uncategorized," the reputation value is 2.

Sample Policy Expressions

Policy	Policy Expressions
Policy to select requests for URLs that are in the Search Engine category	add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY.EQ("Search Engine")
Policy to select requests for URLs that are in the Adult category group	add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO GROUP.EQ("Adult")'
Policy to select requests for Search Engine URLs with a reputation score equal to 4.	add responder policy p2 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY.EQ("Search Engine")'
Policy to select requests for Search Engine and Shopping URLs	add policy patset good_categories; bind policy good_categories "Search Engine"; bind policy good_categories "Shopping"; add responder policy p3 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY .EQUALS_ANY("good_categories")
Policy to select requests for Search Engine URLs with a reputation score equal to 4.	add responder policy p5 'CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(4,0). CATEGORY.EQ("Search Engine")

URL Categorization Policy Actions

A URL filtering policy evaluates the traffic to identify requests belonging to a particular category. The following table lists the actions that you can assign to a URL filtering policy.

Policy Action	Policy Group	Description
ALLOW	Responder	Allow the incoming request to access the target URL
REDIRECT	Responder	Redirect the incoming request to the URL specified as the target.
DENY	Responder	Deny incoming request.
RESET	Responder, VideoOptimization	Reset connection.
DROP	Responder, VideoOptimization	Drop connection.

> **Note**
>
> For encrypted traffic, the VideoOptimization policy includes actions that implement the URL Filtering actions.

Configuring URL Categorization

To configure URL categorization, begin by enabling the URL Filtering feature. You must then configure the cache memory limits, categorization policy, and virtual servers for HTTP and HTTPS traffic. Configuring URL Categorization by using the CLI.

To use the CLI configure URL categorization on a Citrix ADC appliance, do the following:

- Set up URL Categorization.
  - Enable the URL Filtering feature.
  - Configure shared memory to limit cache memory.
  - Configure URL categorization parameters.
- Configure URL categorization for HTTP traffic.
  - Add URL categorization actions.
  - Add URL categorization policies.
  - Add a load balancing virtual server for HTTP traffic.
  - Bind URL categorization policies to the load balancing virtual server.
- Configure URL categorization for HTTPS traffic.

- Add URL categorization policies.
- Add a SSL-Bridge load balancing virtual server.
- Bind URL categorization policies to the load balancing virtual server.

Setting up URL Categorization

To set up the feature, you must enable the URL Categorization feature, configure the filtering parameters and set the shared memory limit.

To enable URL Filtering feature

At the command prompt, type:

```
enable ns feature URLFiltering VideoOptimization Responder IC SSL AppFlow
```

To configure shared memory limit

At the command prompt, type:

```
1 set cache parameter [-memLimit <megaBytes>]
2 <!--NeedCopy-->
```

Where memLimit is the memory limit for caching.

Example:

```
set cache parameter -memLimit 10
```

To configure URL categorization parameters

At the command prompt, type:

```
1 set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>]
     [-TimeOfDayToUpdateDB <HH:MM>]
2 <!--NeedCopy-->
```

***Example:**

```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
03:00
```


Configuring URL categorization for HTTP traffic

To configure the URL categorization feature for HTTP traffic, you must configure a loading balancing virtual server, add URL categorization policies and bind the policies to the virtual server. By doing so, the virtual server receives the HTTP traffic and based on policy evaluation, the system assigns a filtering action.

To add URL categorization action for HTTP traffic

At the command prompt, type:

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment <string
>] [-responseStatusCode <positive_integer>] [-reasonPhrase <string>]
```

Example:

```
add responder action act_url_categorize respondwith "\"HTTP/1.1 200 OK\r\n\
r\n\" + HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).CATEGORY
 + \"\n\""
```

To add URL categorization policy for HTTP traffic

At the command prompt, type:

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <
string>] [-logAction <string>] [-appflowAction <string>]
```

Example:

```
add responder policy pol_url_categorize_http "HTTP.REQ.HOSTNAME.APPEND(
HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Adult\")|| HTTP.REQ.HOSTNAME.
APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Gambling\")"RESET
```

To add an HTTP load balancing virtual server

If a virtual server for HTTP traffic is not already configured, at the command prompt, type:

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-clt Timeout
<secs>]
```

Example:

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout 120
```


To bind URL categorization policy with load balancing virtual server

At the command prompt, type:

```
bind lb vserver <name> -policyName <string> [-priority <positive_integer>]
```

Example:

```
bind lb vserver vsrv-HTTP -policyName pol_url_categorize_http -priority 10
-gotoPriorityExpression END -type REQUEST
```

Configuring URL categorization for HTTPS traffic

To configure the URL categorization feature for HTTPS traffic, you must configure a SSL-bridge loading balancing virtual server, add URL categorization policies and bind the policies to the SSL-bridge virtual server. By doing so, the server receives the HTTPS traffic and based on policy evaluation, the system assigns a filtering action.

To add URL categorization policy for HTTPS traffic

At the command prompt, type:

```
add videooptimization detectionpolicy <name> -rule <expression> -action <
string> [-undefAction <string>] [-comment <string>] [-logAction <string>]
```

Example:

```
add videooptimization detectionpolicy pol_url_categorize_https_block_adult -
rule "CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(0,0).CATEGORY.EQ("Adult")' -
action RESET
```

To add SSL-Bridge load balancing virtual server

At the command prompt, type:

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT imeout
<secs>]
```

Example:

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -cltTimeout
 180
```

To bind categorization policy with SSL-Bridge virtual server

At the command prompt, type:


```
bind lb vserver <name> -policyName <string> [-priority <positive_integer>]
```

Example:

```
bind lb vserver vsrv-HTTPS -policyName pol_url_categorize_https_block_adult
 -priority 20 -type REQUEST
```

Configuring URL Categorization by using the GUI

The GUI enables you to:

- Enable the URL Categorization feature.
- Add URL Categorization actions for HTTP traffic.
- Add URL Categorization policies for HTTP traffic.
- Add URL Categorization policies for HTTPS traffic.
- Add a load balancing virtual server for HTTP traffic.
- Add an SSL bridge load balancing virtual server for HTTPS traffic.
- Bind URL Categorization policies to the load balancing virtual server.
- Bind URL Categorization policies to the SSL-Bridge load balancing virtual server.
- Configure shared memory limit.
- Configure URL categorization parameters.

To enable URL categorization

1. In the navigation pane, expand **System** and then click **Settings**.
2. On the **Settings** page, click **Configure Advanced Features** link.
3. On the **Configure Advanced Features** page, select **URL Filtering** check box.
4. Click **OK** and **Close**.

To add a URL categorization action

1. In the navigation pane, expand **AppExpert** > **Responder** > **Action**.
2. In the details pane, click **Add**.
3. On the **Create Responder Action** page, set the following parameters.
   a) **Name**. Name of the URL categorization policy action.
   b) **Type**. Select an action type.
   c) **Expression**. Use the expression editor to create the policy expression.
   d) **Comments**. A short description of the policy action.
4. Click **Create** and **Close**.

To add a URL categorization policy for HTTP traffic

1. In the navigation pane, expand **AppExpert** > **Responder** > **Policies**.
2. On the details pane, click **Add**.
3. On the **Create Responder Policy** page, set the following parameters.
   a) **Name**. Name of the URL categorization policy action.
   b) **Action**. Select the URL Categorization action that you prefer to associate with the policy.
   c) **Log Action**. Select the log action.
   d) **AppFlow**. Select an AppFlow action.
   e) **Expression**. Use the expression editor to create the policy expression.
   f) **Comments**. A short description about the policy action.
4. Click **Create** and **Close**.

To add a categorization policy for HTTPS traffic

1. Log on to the Citrix ADC appliance and navigate to **Configuration** > **Optimization** > **Video Optimization** > **Detection**.
2. On the **Detection** page, click the **Video Optimization Detection Policies** link.
3. On the Video Optimization Detection Policies page, click **Add**.
4. On the **Create Video Optimization Detection Policy** page, set the following parameters.
   a) **Name**. Name of the optimization policy
   b) **Expression**. Configure policy using custom expressions.
   c) **Action**. Optimization action associated with the policy to handle the incoming video traffic.
   d) **UNDEF Action**. Undefined event if the incoming request does not match the optimization policy.
   e) **Comment**. A short description about the policy.
   f) **Log Action.** Select an audit log action that specifies the action to be performed for the log messages.
5. Click **Create** and **Close**.

To add a load balancing virtual server for HTTP traffic

1. Navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. In the details pane, click **Add**.
3. On the **Load Balancing Virtual Server** page set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. Choose protocol type as HTTP.
   c) **IP Address Type.** IPv4 or IPv6.
   d) **IP Address.** IPv4 or IPv6, VIP address assigned to the virtual server.

e) **Port**. Port number of the virtual server.

4. Click **OK** to continue with the configuration of other, optional, parameters.
5. Click **Create** and **Close**.

To add an SSL-bridge load balancing virtual server

1. Navigate to the **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. On the details pane, click **Add**.
3. On the **Load Balancing Virtual Server** page, set the following parameters:
   a) **Name**. Name of the load balancing virtual server.
   b) **Protocol**. Select protocol type as SSL-bridge.
   c) **IP Address Type**. IP addressable type.
   d) **IP Address**. IP 4 or IP6 IP address assigned to the virtual server.
   e) **Port**. Port number of the virtual server.
4. Choose **OK** to continue configuration other optional parameters.
5. Click **Create** and then **Close**.

To bind a URL categorization policy to the HTTP load balancing virtual server

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** page.
2. On the details pane, select the load balancing virtual server and click **Edit**.
3. In the **Advanced Setting** section, click **Policies**.
4. In the **Policies** section, click the **+** icon to access the **Policies** slider.
5. Set the following parameters.
   a) **Choose Policy**. Select URL categorization policy from the drop-down list.
   b) **Choose Type**. Select the policy type as Request.
6. Click **Continue**.
7. Select the URL categorization policy from the list and click **Close**.

To bind a categorization policy to the SSL-bridge load balancing virtual server

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** screen.
2. In the details pane, select the SSL-bridge load balancing virtual server and click **Edit**.
3. In the **Advanced Setting** section, click **Policies**.
4. In the **Policies** section, click **+** icon to access the **Policies** slider.
5. In the **Policies** section, set the following parameters.
   a) **Choose Policy**. Select video detection policy from the drop-down list.
   b) **Choose Type**. Select the policy type as Request.
6. Click **Continue**.
7. Select the video detection policy from the list and click **Close**.

To configure the shared memory limit

1. Sign on to the appliance and navigate to **Optimization** > **Integrated Caching**.
2. In the details pane, click **Change cache settings** link.
3. On the **Cache Global Settings** page, set the following parameters.
    a) **Memory Usage Limit (MB).**
    b) **Active memory Usage Limit**.
    c) **Via Header**.
    d) **Maximum Post Body Length to be Cached**
    e) **Global Undefined-Result Action**
    f) **Enable HA Object Persist**
    g) **Verify Cached Object Persist**
    h) **Prefetches**
4. Click **OK** and **Close**.

To configure URL categorization parameters

1. Sign to the appliance and navigate to **Security**.
2. On the details pane, click **Change URL filtering settings** link.
3. In the **Configuring URL Filtering Params** page, set the following parameters.
    a) Hours Between DB Updates. URL Filtering hours between database updates. Minimum value: 0 and Maximum value: 720.
    b) Time of Day to Update DB. URL Filtering time of day to update the database.
4. Click **OK** and **Close.**

Configuring Audit Log Messaging

When a Citrix ADC appliance receives an incoming URL, if the responder policy has a URL Filtering expression, the audit log feature collects categorization information and displays it as log messages to any target audit log server that is configured. The info is logged.

- Source IP address (the IP address of the client that made the request).
- Destination IP address (the IP address of the requested server).
- Requested URL containing the schema, the host, and the domain name (`http://www.example.com`).
- URL category that the URL filtering framework returns.
- URL category group that the URL filtering framework returned.
- URL reputation number that the URL filtering framework returned.
- Audit log action taken by URL Categorization policy.

To configure audit logging for the URL List feature, you must complete the following tasks:

1. Enable Audit Log.
2. Create Audit Log message action.
3. Set URL List responder policy with Audit Log message action.

For more information, see Audit Logging topic.

Storing Failure Errors Using SYSLOG Messaging

At any stage of the URL Filtering process, if there is a system-level failure, the Citrix ADC appliance uses the audit log mechanism to store logs in the ns.log file. The errors are stored as text messages in SYSLOG format so that, an administrator can view it later in a chronological order of event occurrence. These logs are also sent to an external SYSLOG server for archival. For more information, see article CTX229399.

For example, if a failure occurs when you initialize the URL Filtering SDK, the error message is stored in the following messaging format.

Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: Error initializing NetStar SDK (SDK error=-1). (status=1).

The Citrix ADC appliance stores the error messages under four different failure categories:

- Download failure. If an error occurs when you try to download the categorization database.
- Integration failure. If an error occurs when you integrate an update into the existing categorization database.
- Initialization failure. If an error occurs when you initialize the URL Categorization feature, set categorization parameters, or end a categorization service.
- Retrieval failure. If an error occurs when the appliance retrieves the categorization details of the request.

URL Reputation Score

The URL Categorization feature provides policy-based control to restrict blacklisted URLs. You can control access to websites based on URL category, reputation score, or URL category and reputation score. If a network administrator monitors a user accessing highly risky websites, he or she can use a responder policy bound to the URL reputation score to block such risky websites.

Upon receiving an incoming URL request, the appliance retrieves the category and reputation score from the URL categorization database. Based on the reputation score returned by the database, the appliance assigns a reputation rating for websites. The value can range from 1 to 4, where 4 is the riskiest type of websites, as shown in the following table.

URL Reputation Rating	Reputation Comment
1	Clean site.
2	Unknown site.
3	Potentially dangerous or affiliated to dangerous site.
4	Malicious site.

FAQs

September 14, 2021

This section provides the FAQ on the following Citrix ADC features

- Admin Partition
- AppFlow
- Call Home
- Clustering
- Connection Management
- Content Switching
- Debugging
- Hardware
- High Availability
- Integrated Caching
- Installing, Upgrading, and Downgrading
- Load Balancing
- NetScaler GUI
- SSL

Admin Partition

September 14, 2021

Where can I get the Citrix ADC configuration file for a partition?

The configuration file (*ns.conf*) for the default partition is available in the */nsconfig* directory. For admin partitions, the file is available in the */nsconfig/partitions/<partitionName>* directory.

How can I configure integrated caching in a partitioned Citrix ADC appliance?

> **Note**
>
> Integrated caching in admin partitions is supported from NetScaler 11.0 onwards.

To configure integrated caching (IC) on a partitioned Citrix ADC, after defining the IC memory on the default partition, the superuser can configure the IC memory on each admin partition such that the total IC memory allocated to all admin partitions does not exceed the IC memory defined on the default partition. The memory that is not configured for the admin partitions remains available for the default partition.

For example, if a Citrix ADC appliance with two admin partitions has 10 GB of IC memory allocated to the default partition, and the IC memory allocation for the two admin partitions is as follows:

- Partition1: 4 GB
- Partition2: 3 GB

Then, the default partition has 10 - (4 + 3) = 3 GB of IC memory available for use.

> **Note**
>
> If all IC memory is used by the admin partitions, no IC memory is available for the default partition.

What is the scope for L2 and L3 parameters in admin partitions?

> **Note**
>
> - Applicable from NetScaler 11.0 onwards.
> - For ARP to work in non-default partition, you must enable the “proxyArp” parameter in the “set l2param” command.

On a partitioned Citrix ADC appliance, the scope of updating the L2 and L3 parameters is as follows:

- For L2 parameters that are set by using the “set L2Param” command, the following parameters can be updated only from the default partition, and their values are applicable to all the admin partitions:

  maxBridgeCollision, bdgSetting, garpOnVridIntf, garpReply, proxyArp, resetInterfaceOnHAfailover, and skip_proxying_bsd_traffic.

The other L2 parameters can be updated in specific admin partitions, and their values are local to those partitions.

- For L3 parameters that are set by using the “set L3Param” command, all parameters can be updated in specific admin partitions, and their values are local to those partitions. Similarly, the values that are updated in the default partition are applicable only to the default partition.

How to enable dynamic routing in an admin partition?

> **Note**
>
> Dynamic routing in admin partitions is supported from NetScaler 11.0 onwards.

While dynamic routing (OSPF, RIP, BGP, ISIS, BGP+) is by default enabled on the default partition, in an admin partition, it must be enabled by using the following command:

```
> set L3Param -dynamicRouting ENABLED
```

> **Note**
>
> A maximum of 63 partitions can run dynamic routing (62 admin partitions and 1 default partition).

On enabling dynamic routing on an admin partition, a virtual router (VR) is created.

- Each VR maintains its own vlan0 which will be displayed as vlan0_<partition-name>.
- All unbound IP addresses that are exposed to ZebOS are bound to vlan0.
- The default VR (of the default partition) shows all the VRs that are configured.
- The default VR shows the VLANs that are bound to these VRs (except default VLANs).

Where can I find the logs for a partition?

Citrix ADC logs are not partition-specific. Log entries for all partitions must be stored in the */var/log/* directory.

How can I get audit logs for an admin partition?

In a partitioned Citrix ADC, you cannot have specific log servers for a specific partition. The servers that are defined at the default partition are applicable across all admin partitions. Therefore, to view the audit logs for a specific partition, you have to use the “show audit messages” command.

> **Note**
>
> The users of an admin partition do not have access to the shell and therefore are not able to access the log files.

How can I get web logs for an admin partition?

You can get the web logs for an admin partition as follows:

- **For NetScaler 11.0 and later versions**

  The web logging feature must be enabled on each of the partitions that require web logging. Using the Citrix ADC Web Logging (NSWL) client, the Citrix ADC retrieves the web logs for all the partitions with which the user is associated.

- **For versions prior to NetScaler 11.0**

  Web logs can be obtained only by `nsroot` and other superusers. Also, even though web logging is enabled on the default partition, the Citrix ADC Web Logging (NSWL) client fetches web logs for all the partitions.

To view the partition for each log entry, customize the log format to include the %P option. You can then filter the logs to view the logs for a specific partition.

How can I get the trace for an admin partition?

You can get the trace for an admin partition as follows:

- **For NetScaler 11.0 and later versions**

  In a partitioned Citrix ADC appliance, the `nstrace` operation can be performed on individual admin partitions. The trace files are stored in the */var/partitions/<partitionName>/nstrace/*directory.

  Note: You cannot get the trace of an admin partition by using the GUI. You must use the CLI.

- **For versions prior to NetScaler 11.0**

  The `nstrace` operation can only be performed on the default partition. Therefore, packet captures are available for the entire Citrix ADC system. To get partition-specific packet captures, use VLAN-ID based filters.

How can I get the technical support bundle specific to an admin partition?

To get the tech support bundle for a specific partition, you must run the following command from the default partition:

```
> show techsupport -scope partition -partitionname <string>
```

Note: This command also gives system-specific information.

AppFlow

September 14, 2021

- **Which build of Citrix ADC supports AppFlow?**

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

- **What is the format used by AppFlow to transmit data?**

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

- **What do AppFlow records contain?**

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

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

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

- **What does the transaction ID in an AppFlow records contain?**

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

- **What is an AppFlow action ?**

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

- **What commands can I run on the Citrix ADC appliance to verify that the AppFlow action is a hit?**

  The show AppFlow action. For example:

```
1   > show appflow action
```



```
1) Name: aFL-act-collector-1
  Collectors: collector-1
  Hits: 0
  Action Reference Count: 2
 2) Name: apfl-act-collector-2-and-3
  Collectors: collector-2, collecter-3
  Hits: 0
  Action Reference Count: 1
 3) Name: apfl-act-collector-1-and-3
  Collectors: collector-1, collecter-3
  Hits: 0
  Action Reference Count: 1
<!--NeedCopy-->
```

- **What is an AppFlow collector?**

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

- **What Citrix ADC verison is required for using AppFlow?**

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

- **What transport protocol does AppFlow use?**

  AppFlow uses UDP as the transport protocol.

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

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

- **How can I change the default port AppFlow uses?**

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

```
> add appflowCollector coll1 -IPAddress 10.102.29.251 -port 8000
 Done
<!--NeedCopy-->
```

- **What does setting clientTrafficOnly do?**

  Citrix ADC generates AppFlow records only for client-side traffic.

- **How many collectors can be configured at a time?**

You can configure up to four AppFlow collectors at a time on the Citrix ADC appliance. Note that the maximum number of collectors that can be configured on a Citrix ADC appliance is four.

Call Home

September 14, 2021

- **What is Call Home on a Citrix ADC appliance?**

  Call Home monitors, and notifies critical events on a Citrix ADC appliance. By enabling Call Home, you can automate the error notification process. You not only avoid calling Citrix support, raise a service request, and upload system data before Citrix support can troubleshoot the issue, but also identify and resolve issues before it occurs.

- **Is Call Home enabled by default on a Citrix ADC appliance?**

  Yes, Call Home is enabled by default on the appliance. If you upgrade to the latest software from an older version where Call Home was disabled by default, the upgrade process automatically enables the feature. If you later choose to disable it, the updated setting is remembered for all further upgrades. For information, see Call Home.

- **What are the pre-requisites for Call Home to work?**

  Access to an Internet connection.

  Note: If your Citrix ADC appliance does not have an Internet connectivity, you can configure a proxy server through which Citrix ADC can generate System Logs and upload it to the Citrix Technical Support server (CIS).

- **What are the benefits of using Call Home?**

  - Monitor hardware and software error conditions.
  - Notify about the occurrence of critical events that impact your network.
  - Send performance data and System Logs to Citrix to:
    * Analyze and improve product quality.
    * Provide real-time troubleshooting information for proactive issue identification, and faster issue resolution.

- **Which release of Citrix ADC software supports Call Home?**

  Citrix ADC release 10.0 and later.

- **What Citrix ADC platform models support Call Home?**

  Call Home feature is enabled by default on all Citrix ADC platforms and all appliance models (MPX, VPX, and SDX).

  - Citrix ADC MPX: All MPX models.
  - Citrix ADC VPX: All VPX models. In addition, it is supported on VPX appliances that obtain their licenses from external or central licensing pools. However, the feature remains the same as for a standard VPX appliance.
  - Citrix ADC SDX: Monitors the disk drive and assigned SSL chips for any errors or failures. The VPX instances, however, do not have access to the Power Supply Unit (PSU) and therefore their status is not monitored. In an SDX platform, you can configure Call Home either directly on an individual instance or through the SVM.

- **Should I configure SNMP alarm for Call Home to notify error conditions?**

  No, you need not configure SNMP for Call Home to monitor error conditions, because SNMP and Call Home uploads are independent of each other. If you want to be notified each time an error condition occurs, you can configure the CALLHOME-UPLOAD-EVENT SNMP alarm to generate an SNMP alert whenever a Call Home upload happens. The SNMP alert notifies the local administrator about the occurrences of critical events.

- **How do I contact a technical support?**

  For all critical hardware-related events, Call Home automatically creates a service request to Citrix. For other errors, after you review the System Logs, you can contact the Citrix Technical Support team to open a service request for further investigation. For more information, see http://support.citrix.com/article/CTX200021.

- **What error conditions does Call Home monitor in a Citrix ADC appliance?**

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

  - Compact flash drive errors
  - Hard disk drive errors
  - Power supply unit failure
  - SSL card failure
  - Warm restart
  - Memory anomalies
  - Rate limit drops

- **Do you need a separate license for Call Home?**

  No, Call Home does not require a separate license. You can enable it in all Citrix ADC platform licenses.

- **What data does Call Home send to Citrix Support server and how frequently is it sent?**

  Call Home collects and sends two types of data to the CIS. They are:

  - Basic System information (running Citrix ADC version, deployment mode (standalone, HA, cluster), hardware details and so on). It is sent at the time of Call Home registration and

as part of periodic heartbeats. The heartbeat is sent once every 30 days, but you can configure this interval anywhere from 1 to 30 days. However, a value of less than 5 days is not recommended, because frequent uploads are usually not very useful.

- An abbreviated version of the `show tech support bundle` when there is an error condition. It is sent upon the first occurrence of a particular error condition since the appliance was last started. That is, a reoccurrence of the same error condition does not trigger another upload unless the appliance was rebooted after the previous occurrence.

- **Can Call Home generate and upload system logs through a proxy server?**

  Yes. If your Citrix ADC appliance does not have direct Internet connectivity, you can configure a proxy server and upload System Logs to the Citrix Technical Support server (CIS).

- **Can I review Call Home data before it is sent to CIS?**

  Unfortunately, you cannot review Call Home data before it is sent to CIS. Call Home does not collect any other data in addition to the data that you will provide when contacting the Citrix support team.

- **How secure and private are the Call Home uploads?**

  Call Home provides data security and privacy in the following ways:

  - Uses a secure SSL/TLS channel to transfer data to Citrix servers.
  - Uploaded data is reviewed only by authorized personnel and is not shared with any third party.

Clustering

September 14, 2021

Click here for FAQs on clustering.

Connection Management

September 14, 2021

- **What is an admin connection?**

  An admin connection establishes a connection to the NSIP address and allows administrators to configure and monitor the Citrix ADC appliance.

- **What are the types of admin connections?**

  There are two types of admin connections:

  - SSH connection – Admin users use an SSH client to log on through the NSIP address.
  - NITRO API connection – Admin users use NITRO APIs to automate the logon process to the Citrix ADC appliance.

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

- **How many concurrent admin connections are allowed on a Citrix ADC appliance?**

  The appliance allows up to 20 concurrent admin connections.

- **Which login credentials are required for an admin logon?**

  Admin logon requires a user name and a password.

  Note: An authentication key can be used instead of a password.

- **Which external authentication methods does a Citrix ADC appliance support?**

  The appliance supports the following external authentication methods:

  - RADIUS
  - LDAP
  - TACACS

- **What is a client?**

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

- **What is a session token?**

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

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

- **What is an active session on a Citrix ADC appliance?**

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

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

- **How does Citrix ADC enforce the concurrent connection limit?**

Every time the Citrix ADC appliance receives an admin connection request (SSH or NITRO API), it checks the number of admin connections it has open. If the number is lower than 20, a new connection is opened.

- **Which counter reflects the number of admin connections on a Citrix ADC appliance?**

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

- **Which counter reflects the number of active tokens on the Citrix ADC appliance?**

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

- **How does Citrix ADC appliance handle errors on a connection?**

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

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

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

- **Does a NITRO session count against the admin connection limit?**

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

- **What is the default timeout period for API, GUI, and CLI sessions on Citrix ADC appliance?**

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

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

- **How can you set the CLI sessions time out on a Citrix ADC appliance?**

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


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

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

  You can override the default timeout period for a NITRO API by setting the timeout duration in the “timeout” field of the login object. If the session timeout is set to zero, the session token has an infinite timeout.

  Note: An infinite timeout is not advisable, because sessions that do not time out continue to count against the admin connection count.

- **What happens if a user account is deleted from the Citrix ADC appliance after an admin session is created?**

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

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

- **Can NITRO API clients use a single session token to open multiple admin connections on the Citrix ADC appliance?**

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

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

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

- **Can a Citrix ADC admin log on to the Citrix ADC appliance after the maximum connections limit is reached?**

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

- **Can NITRO API endpoints open multiple admin connections on Citrix ADC the appliance?**

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

- **Can same client open multiple API sessions on a Citrix ADC appliance?**

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

- **Can API clients use the entire API session token limit?**

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

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

- **Do CLI sessions count against the API session token limit?**

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

- **Can admin users use telnet to open a CLI session?**

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

- **What is connection limit and API session limit applicable for various Citrix ADC releases?**

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

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

Note:

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

Content Switching

September 14, 2021

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

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

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

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

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

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

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

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

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

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

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

- **I understand that the content switching feature of the Citrix ADC appliance helps enhance the performance of the network. Is this correct?**

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

- **Which feature of the Citrix ADC appliance should I configure on the Citrix ADC appliance to enhance the site manageability and response time to the client requests?**

  You can configure the content switching feature of the Citrix ADC appliance to enhance the site manageability and response time to the client request. This feature enables you to create content groups within the same domain name and IP address. This approach is flexible, unlike the common approach of explicitly partitioning the content into different domain names and IP addresses, which are visible to the user.

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

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

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

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

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

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

  - **Method**- HTTP request method.
  - **URL**- URL in the HTTP header.
  - **URL TOKENS**- Special tokens in the URL.

- **VERSION**- HTTP request version.
- **URL QUERY**- Contains the URL Query LEN, URL LEN, and HTTP header.
- **SOURCEIP**- IP address of the client.

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

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

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

- **I want to configure a rule based policy along with a URL based policy for the same content switching virtual server. Is it possible to create both types of policies for the same content switching virtual server?**

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

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

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

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

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

- **What is a Wildcard URL policy?**

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

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

- **What is a Domain and Rule policy?**

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

- **What is the default precedence set for evaluating policies?**

  By default, the rule based policies are evaluated first.

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

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

- **What policy expression syntaxes are supported in content switching?**

  Content switching supports two types of policy expressions:

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

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

  Yes. You can bind a single content switching policy to multiple virtual servers by using policies with defined actions. Content switching policies that use an action can be bound to multiple content switching virtual servers because the target load balancing virtual server is no longer

specified in the content switching policy. The ability to bind a single policy to multiple content switching virtual servers helps to further reduce the size of the content switching configuration.

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

- See CTX122918 - How to Bind the Same Content Switching Policy to Two Content Switching virtual server on a Citrix ADC Appliance.
- See CTX122736 - How to Bind the Same Advanced Policy to Multiple Content Switching Virtual Servers using Policy Labels.
- Configuring Basic Content Switching.

- **Can I create an action based policy using classic expressions?**

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

Debugging

September 14, 2021

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

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

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

Hardware

September 14, 2021

Click here for FAQs about MPX hardware.

High Availability

September 14, 2021

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

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

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

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

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

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

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

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

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

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

- **What are the conditions that trigger synchronization?**

  Synchronization is triggered by any of the following conditions:

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

        * All Interface related commands. For example, set interface and unset interface.
        * All channel-related commands. For example, add channel, set channel, and bind channel.
    - The secondary node comes up after a restart.
    - The primary node becomes secondary after a failover.

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

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

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

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

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

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

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

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

  Forced synchronization fails in any of the following circumstances:

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

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

  Synchronizing configuration files fail if the secondary node is disabled.

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

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

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

  A forced failover fails in any of the following circumstances:

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

Integrated Caching

September 14, 2021

Content Groups

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

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

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

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

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

  `Minhit` is an integer value specifying the minimum number of select to a cache policy before the object is cached. This value is configurable at the content group level. Following is the syntax to configure this value from the CLI.

  ```
  add/set cache contentGroup \<Content_Group_Name> [-minHits \<Integer>]
  ```

- **What is the use of the expireAtLastByte option?**

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

  ```
  add cache contentGroup \<Group_Name> -expireAtLastByte YES
  ```


Cache policy

- **What is a caching policy?**

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

- **What is a policy hit?**

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

- **What is a miss?**

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

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

  ```
  add cache policy image_caching -rule exp1 | ns_ext_not_jpeg -action
  cache
  ```

  ```
  \> ERROR: No such command
  ```

  In the preceding command, the expression must be within the quotation marks. Without quotation marks, the operator is considered to be the pipe operator.

Memory Requirements

- **What are the commands that I can run on the Citrix ADC appliance to check the memory allocated to cache?**

  To display the memory allocated for cache in the Citrix ADC appliance, run any of the following commands from the CLI:

  - `show cache parameter`

    In the output, check the value of the Memory usage limit parameter. This is the maximum memory allocated for cache.

  - `show cache \<Content_Group_Name>`

    In the output, check the values of the Memory usage and Memory usage limit parameters indicating the memory used and allocated for the individual content group.

- **My Citrix ADC appliance has 2 GB of memory. Is there any recommended memory limit for cache?**

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

  ```
  set cache parameter -memLimit 1024
  ```

- **Is it possible to allocate memory for individual content groups?**

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

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

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

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

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

Verification commands

- **What are the options for displaying cache statistics?**

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

  - `stat cache`

    To display the summary of the cache statistics.

  - `stat cache -detail`

    To display the full details of the cache statistics.

- **What are the options for displaying the cached content?**

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

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

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

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

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

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

  ```
  1 add cache policy p2 -rule ns_url_path_cgibin -action CACHE -
      storeInGroup g1
  2 add cache policy p1 -rule ns_url_path_cgibin -action CACHE -
      storeInGroup g2
  3 <!--NeedCopy-->
  ```

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

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

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

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

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

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

  ```
  1 root@ns# /netscaler/nscachemgr -a
  2 //10.102.3.89:80/image8.png
  3 //10.102.3.97:80/staticdynamic.html
  4 //10.102.3.97:80/
  5 //10.102.3.89:80/image1.png
  6 //10.102.3.89:80/file5.html
  7 //10.102.3.96:80/
  ```



```
 8   //10.102.3.97:80/bg_logo_segue.png
 9   //10.102.3.89:80/file500.html
10   //10.102.3.92:80/
11   //10.102.3.96:80/cgi-bin/rfc/ccProxyReval.pl
12   Total URLs in IC = 10
13   <!--NeedCopy-->
```

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

Flushing Objects

- **How can I flush a selective object from the cache?**

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

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

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

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

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

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

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

Flash Cache

- **What is Flash Cache feature of the Citrix ADC appliance?**

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

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

- **What is the limit for Flash Cache clients?**

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

Default Behavior

- **Does the Citrix ADC appliance proactively receive objects upon expiry?**

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

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

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

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

  HOSTNAME_AND_IP is the default value.

- **Does the Citrix ADC appliance create log entries in the log files?**

  Yes. The Citrix ADC appliance creates log entries in the log files.

- **Are compressed objects stored in the cache?**

  Yes. Compressed objects are stored in the cache.

Interoperability with other features

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

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

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

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

- **When using web logging, how do I differentiate entries that indicate response served from cache from those served by the server?**

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

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

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

> **Note**
>
> When using the web logging, the responses from the integrated cache contains the value IC in the server log field. The server log field is present in the NSWL client with "%o1" format specifier.

Miscellaneous

- **What do you mean by configuring relexpiry and absexpiry?**

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

- **What do you mean by configuring weakpos and heuristic?**

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

- **What should I consider before configuring dynamic caching?**

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

- **How is hexadecimal encoding supported in the parameter names?**

  On the Citrix ADC appliance, the %HEXHEX encoding is supported in the parameter names. In the names that you specify for hitParams or invalParams, you can specify a name that contains

%HEXHEX encoding in the names. For example, name, name%65, and n %61m%65 are equivalent.

- **What is the process for selecting a hitParam parameter?**

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

```
POST /data2html.asp?param1=value1&param2=&param3&param4=value4
HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel,
application/msword, application/x-shockwave-flash, */*
Referer: http://10.102.3.97/forms.html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 10.102.3.97
Content-Length: 153
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDQGQGGRNY=NNLLKDADEENOAFLCCDGFGDMO
S1=This+text+is+only+text%2C+not+more+and+not+less%2C+%0D%0Ajust+
   text+to+be+itself%2C+namely+%22Text%22+to+be+posted+as+text
   +%28what+else...%29&B1=Submit
<!--NeedCopy-->
```

  In the preceding request, you can use S1 and B1, highlighted in bold face for your reference, as hitParams depending on your requirements. Also, if you use -matchCookies YES in the ASPSESSIONIDQGQGGRNY content group, then you can also use these parameters as hitParams.

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

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

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

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

- **When are the log entries created for the queued clients?**

  The log entries are created for the queued clients soon after the appliance receives the response

header. The log entries are created only if the response header does not make the object non-cacheable.

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

  The meanings are as follows:

  - `set cache parameter -verifyUsing HOSTNAME`

    The command ignores the destination IP address.

  - `set cache parameter -verifyUsing HOSTNAME_AND_IP`

    The command matches the destination IP address.

  - `set cache parameter -verifyUsing DNS`

    The command uses the DNS server.

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

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

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

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

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

- **What if Hits are zero?**

  Check to see if the server time and NS time are in sync. And the weakPosrelexpiry limit set must bear the time difference between NS and server as following:

```
1  root@ns180\# date
2  Tue May 15 18:53:52 IST 2012
3  <!--NeedCopy-->
```


- **Why are policies getting hits but nothing is being cached?**

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

- **Is it possible to zero the cache counters?**

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

Install, upgrade, and downgrade

September 14, 2021

Install and upgrade

How to download a specific Citrix ADC release build package?

For information on downloading a specific Citrix ADC release build package, see Download a Citrix ADC release package.

How to upgrade the system software of a Citrix ADC appliance?

For information on upgrading the system software of a Citrix ADC appliance, see Upgrade a Citrix ADC standalone appliance.

Where do I find the release notes for a Citrix ADC release build?

The release notes document for a Citrix ADC release build lists the following for the release build:

- Enhancements
- Fixed issues
- Known issues

The release notes document for a Citrix ADC release build is found at the following locations:

- Citrix ADC firmware or virtual appliance downloads page of a specific release build.
- Citrix ADC release notes page in Citrix docs site

Where do I find security updates for Citrix ADC appliances?

Citrix security team regularly releases security bulletins on Common Vulnerabilities and Exposures (CVE) for all related Citrix products. This information can be found at Citrix security bulletin. Alternatively, you can search for a specific CVE at Citrix Support site.

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

A Citrix ADC appliance uses ZebOS as the routing suite. The zebos.conf file available in a Citrix ADC release is the configuration file for ZebOS.

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

Yes. You can change the SSH port on the Citrix ADC appliance by editing the sshd_config file in the /nsconfig directory. If the file does not exist in the /nsconfig directory, copy it from the /etc directory.

In the sshd_config file, edit the entry for Port 22 to Port <Number>, where <Number> is the target port number. If you do not want to restart the appliance and make the changes effective, terminate the `sshd` process by using the kill command, and then restart the process.

The flash directory is missing from the Citrix ADC appliance. What procedure must I follow to mount the flash directory?

To mount the flash directory, do the following:

1. Start the Citrix ADC appliance in single-user mode.

   When the appliance starts, the following message appears:

   Select [Enter] to boot immediately, or any other key for the command prompt. Booting [kernel] in 10 seconds…" Select space and you must see the following prompt:

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

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

   boot –s

   After the appliance starts, the following message appears:

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

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


```
mount /dev/ad0s1a /flash

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

fsck /dev/ad0s1a

Run the mount command again to mount the flash directory.
```

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

```
df -kh
```

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

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

1. Run the following command to generate the public and private keys:

```
\# ssh-keygen -t rsa
```

2. Run the following command to copy the id_rsa.pub file to the .ssh directory of the remote host that you want to log on to:

```
\# scp id_dsa.pub \<user>@\<remote_host>/.ssh/id_dsa.pub
```

3. Log on to the remote host.
4. Change to the .ssh directory.
5. Run the following commands to add the public key of the client to the known public keys:

```
\# cat id_dsa.pub >> authorized_keys2

\# chmod 640 authorized_keys2

\# rm id_dsa.pub
```


What is the procedure to reset the Citrix ADC appliance BIOS? Under what circumstances must I reset the BIOS?

To reset the BIOS of the Citrix ADC appliance, complete the following procedure:

1. Connect to the appliance through the serial port.
2. Start the appliance and press Delete when the boot-up process starts.

   Pressing Delete during the POST process displays the appliance's BIOS settings.
3. Activate the Exit page of the BIOS settings.
4. Select the Load Optimal Defaults option. The Load Optimal Settings message box appears.
5. Select OK.
6. Make the following changes to the BIOS settings on the various tabs:

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

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

I need to reset the Citrix ADC appliance to the factory defaults. What procedure must I follow?

To reset the Citrix ADC appliance to the factory defaults, you need to reset two environments: the Citrix ADC application environment and the base FreeBSD environment.
To reset the Citrix ADC application environment of the appliance to the factory defaults, do the following:

1. Make a backup of the appliance's /nsconfig/ns.conf.
2. Delete the /nsconfig/ns.conf file.
3. Restart the appliance. To reset the FreeBSD environment of the appliance to the factory defaults, do the following:
   a) Install a fresh Citrix ADC code image on the appliance. This overwrites several FreeBSD-level configuration files with default values.
   b) Delete any users and groups that are added to the appliance, that is, all except the default users.

c) Delete the /etc/resolv.conf file.
d) Delete the entries that you have added to the /etc/hosts file.
e) If the /etc/rc.netscaler file exists, delete it.
f) Open the /etc/nsperm_group_suser file and make sure that all IOCTL entries are comment entries.
g) Open the /etc/rc.conf file and make sure that the syslogd_enable=NO entry is not changed to syslogd_enable=YES.
h) Open the /etc/syslog.conf file and make sure that there are no additional entries in the file.
i) Delete the contents of the /var/nslog, /var/nstrace, and /var/crash files.
j) If the syslog process is enabled on the appliance and the appliance creates log files locally, delete the contents of the log files listed in the /etc/syslog.conf file. The files are created in the /var/log directory. For example, if the syslog process writes system events to the /var/log/events file, and `sslvpn` access events to the /var/log/sslvpnevents file, delete these files.

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

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

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

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

After I upgraded the Citrix ADC release on the appliance, the appliance still displays the earlier release/build. What can be the reason?

The appliance displays the software version number from the /flash/boot/loader.conf file. If the kernel entry for the current Citrix ADC release is missing from that file, the appliance displays the last Citrix ADC release version for which the entry was available.
To resolve this issue, do the following:

1. Verify that the kernel file exists in the /nsconfig directory.

2. Check the /flash/boot/loader.conf file for an entry for the kernel.

   (You can expect the entry for the kernel of the release/build that you installed to be missing from the file.)

3. Open the loader.conf file in a text editor, such as the vi editor, and update the kernel entry for the new release/build.
4. Save and close the file.
5. Repeat step 2 through step 4 for the /flash/boot/loader.conf.local file.
6. Update the release/build entry in the ns.conf file.
7. Restart the appliance.

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

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

```
1 /netscaler/nslcd -k
```

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

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

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

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

1. When you restart the appliance, press the F1 key when the appliance console displays the Press F1 message.
2. Type kernel. GENERIC and press Enter.
3. Log in as the root user.
4. Reinstall the Citrix ADC release.
5. Restart the appliance.

After upgrading the appliance software, I am not able to log on to the appliance, and the following message appears. I tried to resolve this issue by using the password recovery procedure, but I was not successful. Have I done something incorrectly?

```
```
login: nsroot
Password:
connect: No such file or directory
nsnet_connect: No such file or directory
Login incorrect
<!--NeedCopy--> ```
```

You cannot resolve this issue by using the password recovery procedure. Citrix ADC release 12.1 or later use the new licensing system, based on the `Imgrd` daemon, which runs during the startup procedure. For this daemon to work properly, the host name of Citrix ADC appliance, which is set in the /nsconfig/rc.conf file, must be resolved by a name server to the NSIP address. Alternately, you can create a hosts file in the /nsconfig directory and add the 127.0.0.1 <Host_Name> entry in file.

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

**During an upgrade of a high availability pair, the following message appears repeatedly. What can be the reason?**

<auth.err> ns sshd[5035]: error: Invalid user name or password

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

**I want to change the netmask of the NSIP address on a Citrix ADC appliance. Can I do so without causing an outage?**

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

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

 
```

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

You can use the following procedure to restore the configuration:

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

```
save config

shell

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

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

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

Can the primary appliance and secondary appliance have separate builds?

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

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

No. In an HA pair, first upgrade the secondary node and then upgrade the primary node.

For details, refer Upgrading a High Availability Pair.

Does Citrix support firmware upgrades in the Amazon Web Services cloud?

Yes.

Can I upgrade the Citrix ADC instance independently of the SDX version?

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

Can I use the FTP server to upgrade the Citrix ADC appliance?

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

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

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

Downgrade

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

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

When downgrading the Citrix ADC release, I followed the instructions. However, the appliance displays the following message. How is the rollback procedure performed on a Citrix ADC appliance?

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

The rollback procedure is similar to the basic upgrade procedure. Select the target build that you want to roll back to and perform the downgrade. Before rolling back to a different release, Citrix recommends that you create a copy of your current configuration files. To downgrade from a release, see Downgrading a Citrix ADC Standalone Appliance.

Load Balancing

September 14, 2021

- **What are the various load balancing policies I can create on the Citrix ADC appliance?**

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

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

- **Can I achieve the Web farm security by implementing load balancing using the Citrix ADC appliance?**

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

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

- **What are various devices that I can use to load balance with a Citrix ADC appliance?**

  You can load balance the following devices with a Citrix ADC appliance:

  - Server farms
  - Caches or Reverse Proxies

  - Firewall devices
  - Intrusion detection systems
  - SSL offload devices
  - Compression devices
  - Content Inspection servers

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

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

  - Reduce the response time: When you implement the load balancing feature for the website, one of the major benefits is the boost you can look forward to in load time. With two or more servers sharing the load of the web traffic, each of the servers runs less traffic load than a single server alone. This means there are more resources available to fulfill the client requests. This results in a faster website.
  - Redundancy: Implementing the load balancing feature introduces a bit of redundancy. For example, if the website is balanced across three servers and one of them does not respond at all, the other two can keep running and the website visitors do not even notice any downtime. Any load balancing solution immediately stops sending traffic to the back-end server that is not available.

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

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

- **What are the various persistence types available on the Citrix ADC appliance?**

  The Citrix ADC appliance supports the following persistence types:

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

GUI

September 14, 2021

- **When I use Firefox to compare two Citrix ADC configurations, the browser seems to freeze?**

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

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

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

- **What should I do before accessing the GUI?**

  Before accessing a new version of the Citrix ADC software:

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

- **Which port should I open to access GUI using HTTP or HTTPS?**

  The following lists the default port numbers for HTTP and HTTPS management services (GUI) in the Citrix ADC MPX, VPX and CPX appliances:

  - Citrix ADC MPX and VPX appliances: 80 (HTTP) and 443 (HTTPS)
  - Citrix ADC CPX appliances: 9080 (HTTP) and 9443 (HTTPS)

  Also, you can configure ports for HTTP and HTTPS management services (GUI) other than port 80 and 443. For more information, see Configure HTTP and HTTPS management ports.

- **With which browsers is the GUI compatible for different operating systems?**

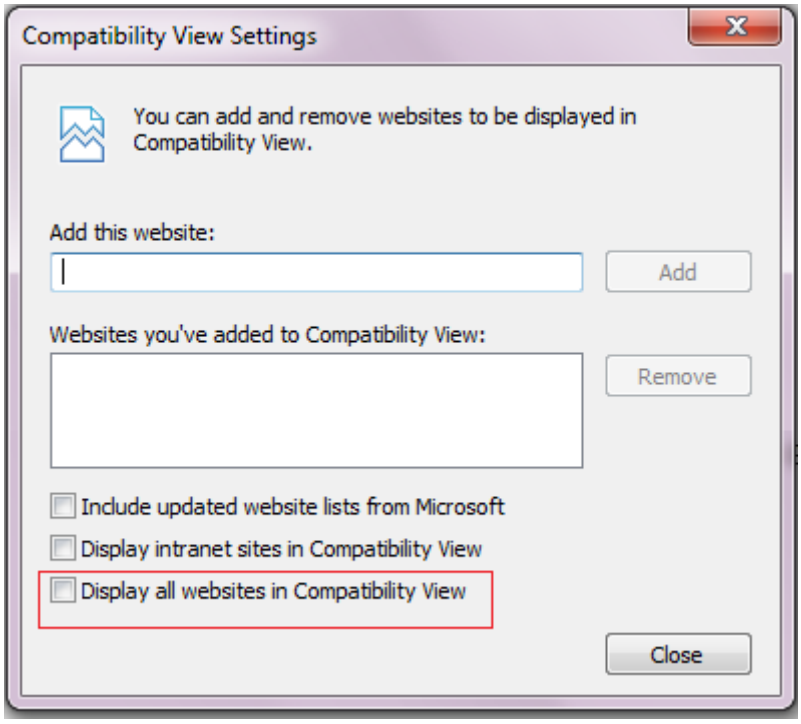
  The following table lists the compatible browsers for NetScaler GUI version 12.0, 12.1, and 13.0:

Operating System	Browser	Versions
Windows 7 & later	Internet Explorer	11, Edge, & later

Operating System	Browser	Versions
Windows 7 & later	Mozilla Firefox	45 & later
Windows 7 & later	Chrome	60 & later
MAC	Mozilla Firefox	45 & later
MAC	Safari	10.1.1 & later

- **When I access the GUI by using Internet Explorer version 8 or 9, the browser displays only a grey bar at the top of the screen. What should I do?**

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



- **Even after I disable compatibility mode in Internet Explorer version 8 or 9, the GUI does not appear. What should I do?**

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

- **After logging into the Citrix ADC appliance, the page appears blank. What should I do?**

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

To disable this option:

1. In your Internet Explorer browser settings, go to **Internet Options**.
2. Go to **Security** tab settings, click **Restricted sites** zone to disable **Enable Protected Mode** check box.
3. Click **Apply** and **OK**.

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

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

SSL

September 14, 2021

Click here for FAQs about SSL.

Authentication, authorization, and auditing application traffic

September 14, 2021

Many companies restrict website access to valid users only, and control the level of access permitted to each user. The authentication, authorization, and auditing feature allows a site administrator to manage access controls with the Citrix ADC appliance instead of managing these controls separately for each application. Doing authentication on the appliance also permits sharing this information across all websites within the same domain that are protected by the appliance.

To use authentication, authorization, and auditing, you must configure authentication virtual servers to handle the authentication process and traffic management virtual servers to handle the traffic to web applications that require authentication. You also configure your DNS to assign FQDNs to each virtual server. After configuring the virtual servers, you configure a user account for each user that will authenticate via the Citrix ADC appliance, and optionally you create groups and assign user accounts to groups. After creating user accounts and groups, you configure policies that tell the appliance how to authenticate users, which resources to allow users to access, and how to log user sessions. To put the policies into effect, you bind each policy globally, to a specific virtual server, or to the appropriate user accounts or groups. After configuring your policies, you customize user sessions by configuring

session settings and binding your session policies to the traffic management virtual server. Finally, if your intranet uses client certs, you set up the client certificate configuration.

To understand how authentication, authorization, and auditing works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the ADC does the following:

- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through the Lightweight Directory Access Protocol (LDAP). For more information, see Determining Attributes in Your LDAP Directory.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.
- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

Supported authentication types

- Local
- LDAP
- RADIUS
- SAML
- TACACS+
- Client certificate authentication (including smart card authentication)
- Web
- Advanced authentication
- Forms based authentication
- 401 based authentication
- Native OTP
- Push notification
- Email OTP
- reCaptcha

Citrix Gateway also supports RSA SecurID, Gemalto Protiva, and SafeWord. You use a RADIUS server to configure these types of authentication.

Before configuring authentication, authorization, and auditing, you must be familiar with and understand how to configure load balancing, content switching, and SSL on the Citrix ADC appliance.

Authentication without authorization

Authorization specifies the network resources to which users have access when they log on to the appliance. The default setting for authorization is to deny access to all network resources. Citrix recommends using the default global setting and then creating authorization policies to define the network resources users can access.

You configure authorization on the appliance by using an authorization policy and expressions. After you create an authorization policy, you can bind it to the users or groups that you configured on the appliance.

You can configure the appliance to use authentication only, without authorization. When you configure authentication without authorization, the appliance does not perform a group authorization check. The policies that you configure for the user or group are assigned to the user.

Enabling authentication, authorization, and auditing

To use the authentication, authorization, and auditing feature, you must enable it. You can configure authentication, authorization, and auditing entities—such as the authentication and traffic management virtual servers—before you enable the authentication, authorization, and auditing feature, but the entities do not function until the feature is enabled.

To enable authentication, authorization, and auditing by using the CLI

At the command prompt, type the following commands to enable authentication, authorization, and auditing and verify the configuration:

```
enable ns feature AAA
<!--NeedCopy-->
```

To enable authentication, authorization, and auditing by using the GUI

1. Navigate to **System > Settings**.
2. In the details pane, under **Modes and Features**, click **Change Basic Features**.
3. In the **Configure Basic Features** dialog box, select the **Authentication, Authorization and Auditing** check box.
4. Click **OK**.

Disabling Authentication

If your deployment does not require authentication, you can disable it. You can disable authentication for each virtual server that does not require authentication.

> **Important:**
>
> **Important:** Citrix recommends disabling authentication with caution. If you are not using an external authentication server, create local users and groups to allow the appliance to authenticate users. Disabling authentication stops the use of authentication, authorization, and accounting features that control and monitor connections to the appliance. When users type a web address to connect to the appliance, the logon page does not appear.

To disable authentication

1. Navigate to **Configuration > Citrix Gateway > Virtual Servers**.
2. In the details pane, click a virtual server, and then click **Open**.
3. In the **Basic Settings** page, clear the **Enable Authentication** check box.

How authentication, authorization, and auditing works

September 14, 2021

Authentication, authorization, and auditing provides security for a distributed internet environment by allowing any client with the proper credentials to connect securely to protected application servers from anywhere on the Internet. This feature incorporates the three security features of authentication, authorization, and auditing. Authentication enables the Citrix ADC to verify the client's credentials, either locally or with a third-party authentication server, and allow only approved users to access protected servers. Authorization enables the ADC to verify which content on a protected server it allows each user to access. Auditing enables the ADC to keep a record of each user's activity on a protected server.

To understand how authentication, authorization, and auditing works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the ADC does the following:

- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through LDAP. For more information, see Determining Attributes in Your LDAP Directory.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.

- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

Configure authentication authorization and auditing policies

After you set up your users and groups, you next configure authentication policies, authorization policies, and audit policies to define which users are allowed to access your intranet, which resources each user or group is allowed to access, and what level of detail authentication, authorization, and auditing will preserve in the audit logs. An authentication policy defines the type of authentication to apply when a user attempts to log on. If external authentication is used, the policy also specifies the external authentication server. Authorization policies specify the network resources that users and groups can access after they log on. Auditing policies define the audit log type and location.

You must bind each policy to put it into effect. You bind authentication policies to authentication virtual servers, authorization policies to one or more user accounts or groups, and auditing policies both globally and to one or more user accounts or groups.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the Citrix ADC operating system, policy priorities work in reverse order: the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The authentication, authorization, and auditing feature implements only the first of each type of policy that a request matches, not any additional policies of that type that a request might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind the policies. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the Citrix ADC appliance, see the Citrix ADC product documentation.

Configure the No_Auth policy to bypass certain traffic

You can now configure No_Auth policy to bypass certain traffic from authentication when 401-based authentication is enabled on traffic management virtual server. For such traffic, you must bind a “No_Auth” policy.

To configure the No_Auth policy to bypass certain traffic by using the CLI

At the command prompt, type:

```
add authentication policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

Example:

```
add authentication policy ldap -rule ldapAct1 -action No_Auth
<!--NeedCopy-->
```

Basic components of authentication, authorization, and auditing configuration

September 14, 2021

The basic components of the authentication, authorization, and auditing configuration are as follows:

- **Authentication virtual server** - All authentication requests are redirected by the traffic management virtual server (load balancing or content switching) to the authentication virtual sever. This virtual server processes the associated authentication policies and accordingly provides access to the application. For details, see Authentication virtual server.
- **Authentication profiles** - An authentication profile specifies the authentication virtual server, the authentication host, the authentication domain, and an authentication level.

You can create one or more authentication profiles to specify different authentication settings and bind these authentication profiles to relevant traffic management servers based on your requirements. For details, see Authentication profiles.

- **Authentication policies** - When users log on to the Citrix ADC or Citrix Gateway appliance, they are authenticated according to a policy that you create. An authentication policy comprises of an expression and an action. Authentication policies use Citrix ADC expressions. For details, see Authentication policies.
- **Authorization policies** - When you configure an authorization policy, you can set it to allow or deny access to network resources in the internal network. For details, see Authorization policies.
- **Users and groups:** - After configuring the authentication, authorization, and auditing basic setup, you create users and groups. You first create a user account for each person who will authenticate via the Citrix ADC appliance. If you are using local authentication controlled by

the Citrix ADC appliance itself, you create local user accounts and assign passwords to each of those accounts. For details, see Users and groups.

Authentication virtual server

September 14, 2021

The traffic management virtual server (load balancing or content switching) redirects all authentication requests to the authentication virtual server. This virtual server processes the associated authentication policies and accordingly provides access to the application.

Note: You cannot bind traffic management policies to authentication, authorization, and auditing virtual servers.

Set up authentication virtual server

The steps involved in setting up an authentication virtual server are;

1. Enable the authentication, authorization, and auditing feature.

```
enable ns feature AAA
<!--NeedCopy-->
```

2. Configure an authentication virtual server. It must be of type SSL and make sure to bind the SSL certificate-key pair to the virtual server.

```
add authentication vserver <name> SSL <ipaddress> <port>

bind ssl certkey <auth-vserver-name> <certkey>
<!--NeedCopy-->
```

3. Specify the FQDN of the domain for the authentication virtual server.

```
set authentication vserver <name> -authenticationDomain <FQDN>
<!--NeedCopy-->
```

4. Associate the authentication virtual server to the relevant traffic management virtual server.

   Points to Note:

   - The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly. On the traffic management virtual server:
     - Enable authentication.

- Specify the FQDN of the authentication virtual server as the authentication host of the traffic management virtual server.
- [Optional] Specify the authentication domain on the traffic management virtual server.
- If you do not configure the authentication domain, the appliance assigns an FQDN that consists of the FQDN of the authentication virtual server without the host name portion. For example, if the domain name of the authentication virtual server is **tm.xyz.bar.com**, the appliance assigns **xyz.bar.com** as the authentication domain.
  * For load balancing:

```
set lb vserver <name> -authentication ON -
   authenticationhost <FQDN> [-authenticationdomain <
   authdomain>]
<!--NeedCopy-->
```

  * For content switching:

```
set cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

- If you have to set a domain wide cookie for an authentication domain, you must enable authentication profile on a load balancing virtual server.

5. Verify that both the virtual servers are UP and configured correctly.

```
show authentication vserver <name>
<!--NeedCopy-->
```

To set up an authentication virtual server by using the GUI

1. Enable the authentication, authorization, and auditing feature.

   Navigate to **System > Settings**, click **Configure Basic features**, and enable **Authentication, Authorization and Auditing**.

2. Configure the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and configure as required.

3. Configure the traffic management virtual server for authentication.

   - **For load balancing:**

     Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and configure the virtual server as required.

   - **For content switching:**

Navigate to **Traffic Management > Content Switching > Virtual Servers**, and configure the virtual server as required.

4. • Verify the authentication setup.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and check the details of the relevant authentication virtual server.

Configure the authentication virtual server

To configure authentication, authorization, and auditing, first configure an authentication virtual server to handle authentication traffic. Next, bind an SSL certificate-key pair to the virtual server to enable it to handle SSL connections.
For additional information about configuring SSL and creating a certificate-key pair, see SSL certificates.

Configure an authentication virtual server by using the CLI

To configure an authentication virtual server and verify the configuration, at the command prompt type the following commands in the same order:

```
add authentication vserver <name> ssl <ipaddress>

show authentication vserver <name>

bind ssl certkey <certkeyName>

show authentication vserver <name>

set authentication vserver <name>

show authentication vserver <name>
<!--NeedCopy-->
```

Example:

```
add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done

```



```
bind ssl certkey Auth-Vserver-2 Auth-Cert-1 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: UP Client Idle Timeout
    : 180 sec Down state flush: DISABLED Disable Primary Vserver On Down
     : DISABLED Authentication : ON Current AAA Users: 0 Done

set authentication vserver Auth-Vserver-2

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done
<!--NeedCopy-->
```

> **Note**
>
> The Authentication Domain parameter is deprecated. Use Authentication Profile for setting domain wide cookies.

Configure an authentication virtual server by using the GUI

1. Navigate to **Security > AAA - Application Traffic > Virtual Servers**.
2. In the details pane, do one of the following:
   - To create a new authentication virtual server, click **Add**.
   - To modify an existing authentication virtual server, select the virtual server, and then click **Edit**. The Configuration dialog opens with the Basic Settings area expanded.
3. Specify values for the parameters as follows (asterisk indicates a required parameter):
   - Name*—name (Cannot be changed for a previously created virtual server)
   - IP Address Type*—IP address type of the authentication virtual server
   - IP Address*—IP address of the authentication virtual server
   - Port*—TCP port on which the virtual server accepts connections.
   - Failed login timeout—failedLoginTimeout (Seconds allowed before login fails, and user must start login process again.)
   - Max login attempts—maxLoginAttempts (Number of login attempts allowed before user is locked out)

   > **Note:**
   >
   > The authentication virtual server uses only the SSL protocol and port 443, so those options

> are grayed out. Any options that are not mentioned can be ignored.

4. Click **Continue** to display the Certificates area.
5. In the **Certificates** area, configure any SSL certificates you want to use with this virtual server.
   - To configure a CA certificate, click the arrow on the right of CA Certificate to display the CA Cert Key dialog box, select the certificate you want to bind to this virtual server, and click **Save**.
   - To configure a server certificate, click the arrow on the right of Server Certificate, and follow the same process as for the CA certificate.
6. Click **Continue** to display the **Advanced Authentication Policies** area.
7. If you want to bind an advanced authentication policy to the virtual server, click the arrow on the right side of the line to display the **Authentication Policy** dialog box, choose the policy that you want to bind to the server, set the priority, and then click **OK**.
8. Click **Continue** to display the **Basic Authentication Policies** area.
9. If you want to create a basic authentication policy and bind it to the virtual server, click the plus sign to display the **Policies** dialog box, and follow the prompts to configure the policy and bind it to this virtual server.
10. Click **Continue** to display the 401-Based Virtual Servers area.
11. In the 401-Based Virtual Servers area, configure any load balancing or content switching virtual servers that you want to bind to this virtual server.
    - To bind a load balancing virtual server, click the arrow to the right of load balancing virtual server to display the Load Balancing Virtual Servers dialog box, and follow the prompts.
    - To bind a content switching virtual server, click the arrow to the right of content switching virtual server to display the Content Switching Virtual Servers dialog box, and follow the same process as to bind an LB virtual server.
12. If you want to create or configure a group, in the Groups area click the arrow to display the Groups dialog box, and follow the prompts.
13. Review your settings, and when you are finished, click **Done**. The dialog box closes. If you created a new authentication virtual server, it now appears in the **Configuration** window list.

Traffic management virtual server

After you have created and configured your authentication virtual server, you next create or configure a traffic management virtual server and associate your authentication virtual server with it. You can use either a load balancing or content switching virtual server for a traffic management virtual server. For more information about creating and configuring either type of virtual server, see the *Citrix Traffic Management Guide* at Traffic Management.

> **Note:**
>
> The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

You configure a traffic management virtual server for authentication, authorization, and auditing by enabling authentication and then assigning the FQDN of the authentication server to the traffic management virtual server. You can also configure the authentication domain on the traffic management virtual server currently. If you do not configure this option, the Citrix ADC appliance assigns the traffic management virtual server an FQDN that consists of the FQDN of the authentication virtual server without the host name portion. For example, if the domain name of the authentication virtual server is tm.xyz.bar.com, the appliance assigns xyz.bar.com. as the authentication domain.

To configure a traffic management virtual server by using the CLI

At the command prompt, type one of the following sets of commands:

```
set lb vserver <name>  -authentication ON -authenticationhost <FQDN> [-
    authenticationdomain <authdomain>]
show lb vserver <name>
set cs vserver <name>  -authentication ON -authenticationhost <FQDN> [-
    authenticationdomain <authdomain>]
show cs vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki
    .index.com Done

show lb vserver vs-cont-sw vs-cont-sw (0.0.0.0:0) - TCP Type: ADDRESS
    State: DOWN Last state change was at Wed Aug 19 10:03:15 2009 (+410
    ms) Time since last state change: 5 days, 20:00:40.290 Effective
    State: DOWN Client Idle Timeout: 9000 sec Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED No. of Bound Services : 0
     (Total) 0 (Active) Configured Method: LEASTCONNECTION Mode: IP
    Persistence: NONE Connection Failover: DISABLED Authentication: ON
    Host: mywiki.index.com
Done
<!--NeedCopy-->
```

To configure a traffic management virtual server by using the GUI

1. In the navigation pane, do one of the following.

- Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
- Navigate to **Traffic Management > Content Switching > Virtual Servers**
- In the details pane, select the virtual server on which you want to enable authentication, and then click **Edit**.
- In the Domain text box, type the authentication domain.
- In the **Advanced** menu on the right, select **Authentication**.
- Choose either **Form Based Authentication** or **401 Based Authentication**, and fill in the Authentication information.
  - For Form Based Authentication, enter the Authentication FQDN (the fully qualified domain name of the authentication server), the Authentication virtual server (the IP address of the authentication virtual server), and the Authentication Profile (the profile to use for authentication).
  - For 401 Based Authentication, enter the Authentication virtual server and the Authentication Profile only.
- Click **OK**. A message appears in the status bar, stating that the virtual server has been configured successfully.

Simplified login protocol support for authentication, authorization, and auditing

The login protocol between authentication, authorization, and auditing traffic management virtual servers and authentication, authorization, and auditing virtual servers is simplified to use internal mechanisms as opposed to sending the encrypted data through query parameters. Using this feature, the replay of requests is prevented.

Configure DNS

For the domain session cookie used in the authentication process to function correctly, you must configure DNS to assign both the authentication and the traffic management virtual servers to FQDNs in the same domain. For information about how to the configure DNS address records, see Domain Name System.

Verify authentication virtual server

After you configure authentication and traffic management virtual servers and before you create user accounts, you must verify that both virtual servers are configured correctly and are in the UP state.

Configure a noAuth authentication by using the CLI

At the command prompt, type the following command:

```
show authentication vserver <name>
<!--NeedCopy-->
```

Example:

```
show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Done
<!--NeedCopy-->
```

Configure a noAuth authentication by using the GUI

1. Navigate to **Security > Citrix ADC AAA - Application Traffic > Virtual Servers**.
   Note: From Citrix Gateway, navigate to **Citrix Gateway > Virtual Servers**.
2. Review the information in the **AAA Virtual Servers** pane to verify that your configuration is correct and your authentication virtual server is accepting traffic. You can select a specific virtual server to view detailed information in the details pane.

Authorization policies

September 14, 2021

When you configure an authorization policy, you can set it to allow or deny access to network resources in the internal network. For example, to allow users access to the 10.3.3.0 network, use the following expression:

CLIENT.IP.DST.IN_SUBNET(10.3.0.0/16)

Authorization policies are applied to users and groups. After a user is authenticated, Citrix Gateway performs a group authorization check by obtaining the user's group information from either an RADIUS, LDAP, or TACACS+ server. If group information is available for the user, Citrix Gateway checks the network resources allowed for the group.

To control which resources users can access, you must create authorization policies. If you do not need to create authorization policies, you can configure default global authorization.

If you create an expression within the authorization policy that denies access to a file path, you can only use the subdirectory path and not the root directory. For example, use fs.path contains "\\dir1\\dir2" instead of fs.path contains "\\rootdir\\dir1\\dir2". If you use the second version in this example, the policy fails.

After you configure the authorization policy, you then bind it to a user or group.

By default, authorization policies are validated first against policies that you bind to the virtual server and then against policies bound globally. If you bind a policy globally and want the global policy to take precedence over a policy that you bind to a user, group, or virtual server, you can change the priority number of the policy. Priority numbers start at zero. A lower priority number gives the policy higher precedence.

For example, if the global policy has a priority number of one and the user has a priority of two, the global authentication policy is applied first.

> **Important:**
>
> - Classic authorization policies are applied only on TCP traffic.
> - Advanced authorization policy can be applied on all types of traffic (TCP/UDP/ICMP/DNS).
>   - To apply policy on UDP/ICMP/DNS traffic, policies must be bound at type UDP_REQUEST, ICMP_REQUEST, and DNS_REQUEST respectively.
>   - While binding, if "type" is not explicitly mentioned or "type" is set to REQUEST, the behavior does not change from earlier builds, that is these policies are applied only to TCP traffic.
>   - The policies bound at UDP_REQUEST do not apply for DNS traffic. For DNS, policies must be explicitly bound to DNS_REQUEST TCP_DNS is similar to other TCP requests.

For more details on advanced authorization policies, see article https://support.citrix.com/article/CTX232237.

Configure and bind an authorization policy

Configure an authorization policy by using the GUI

1. Navigate to **Citrix Gateway > Policies > Authorization.**
2. In the details pane, click **Add**.
3. In **Name**, type a name for the policy.
4. In **Action**, select **Allow** or **Deny**.
5. In **Expression**, click **Expression Editor**.

6. To start to configure the expression, click **Select** and choose the necessary elements.
7. Click **Done** when your expression is complete.
8. Click **Create**.

Bind an authorization policy to a user by using the GUI

1. Navigate to **Citrix Gateway > User Administration.**
2. Click **AAA Users**.
3. In the details pane, select a user and then click **Edit**.
4. In **Advanced Settings**, click **Authorization Policies**.
5. In **Policy Binding** page, select a policy or create a policy.
6. In **Priority**, set the priority number.
7. In **Type**, select the request type and then click **OK**.

Bind an authorization policy to a group by using the GUI

1. Navigate to **Citrix Gateway > User Administration**.
2. Click **AAA Groups**.
3. In the details pane, select a group and then click **Edit**.
4. In **Advanced Settings**, click **Authorization Policies**.
5. In **Policy Binding** page, select a policy or create a policy.
6. In **Priority**, set the priority number.
7. In **Type**, select the request type and then click **OK**.

Authorization specifies the network resources to which users have access when they log on to Citrix Gateway. The default setting for authorization is to deny access to all network resources. Citrix recommends using the default global setting and then creating authorization policies to define the network resources users can access.

You configure authorization on Citrix Gateway by using an authorization policy and expressions. After you create an authorization policy, you can bind it to the users or groups that you configured on the appliance.

Default global authorization

To define the resources to which users have access on the internal network, you can configure default global authorization. You configure global authorization by allowing or denying access to network resources globally on the internal network.

Any global authorization action you create is applied to all users who do not already have an authorization policy associated with them, either directly or through a group. A user or group authorization policy always overrides the global authorization action. If the default authorization action is set to

Deny, you must apply authorization policies for all users or groups to make network resources accessible to those users or groups. This requirement helps to improve security.

To set default global authorization:

1. In the configuration utility, on the Configuration tab, in the navigation pane, expand Citrix Gateway and then click Global Settings.
2. In the details pane, under Settings, click Change global settings.
3. On the Security tab, next to Default Authorization Action, select Allow or Deny then and click OK.

Authentication profiles

September 14, 2021

When you want the same authentication settings to be used by multiple traffic management virtual servers, you can create an authentication profile which specifies the authentication virtual server, the authentication host, the authentication domain, and the authentication level.

This authentication profile can be associated with the relevant traffic management virtual servers.

Configure an authentication profile

Configure an authentication profile by using the CLI

- Create the authentication profile and set the required parameters.

  For example, to create a profile with an authentication virtual server named “authVS”.

```
1  add authentication authnProfile authProfile1 -authnVsName authVS
       -authenticationHost authnVS.example.com -authenticationDomain
       example.com -authenticationLevel
2  <!--NeedCopy-->
```

> **Note:**
>
> The authentication weight or level depends on the virtual server to which the traffic is bound. A session that is created by authenticating against the traffic management virtual server at a given level cannot be used to access the traffic management virtual server at a higher level.

- Bind the authentication profile to the relevant traffic management virtual servers.

  For example, to bind authProfile1 to a load balancing virtual server named “vserver1”.


```
set lb vserver vserver1 -authnProfile authProfile1
<!--NeedCopy-->
```

Configure an authentication profile by using the GUI

In the **Configuration** tab, navigate to **Security > AAA - Application Traffic > Authentication Profile**, and configure the authentication profile as required.

> **Note:**
>
> - You can create an authentication profile by using the Citrix Gateway wizard as well. The profile contains all the settings for the authentication policy. You configure the profile when you create the authentication policy.
> - With the Citrix Gateway wizard, you can use the chosen authentication type to configure authentication. If you want to configure other authentication policies after running the wizard, you can use the configuration utility. For more information about the Citrix Gateway wizard, see Configuring Settings by Using the Citrix Gateway Wizard].

Authentication policies

September 14, 2021

When users log on to the Citrix ADC or Citrix Gateway appliance, they are authenticated according to a policy that you create. An authentication policy comprises of an expression and an action. Authentication policies use Citrix ADC expressions.

After creating an authentication action and an authentication policy, bind it to an authentication virtual server and assign a priority to it. When binding it, also designate it as either a primary or a secondary policy. Primary policies are evaluated before secondary policies. In configurations that use both types of policy, primary policies are normally more specific policies while secondary policies are normally more general policies. It is intended to handle authentication for any user accounts that do not meet the more specific criteria. The policy defines the authentication type. A single authentication policy can be used for simple authentication needs and is typically bound at the global level. You can also use the default authentication type, which is local. If you configure local authentication, you must also configure users and groups on the appliance.

You can configure multiple authentication policies and bind them to create a detailed authentication procedure and virtual servers. For example, you can configure cascading and two-factor authentication by configuring multiple policies. You can also set the priority of the authentication policies to

determine which servers and the order in which the appliance checks user credentials. An authentication policy includes an expression and an action. For example, if you set the expression to True value, when users log on, the action evaluates user logon to true and then users have access to network resources.

After you create an authentication policy, you bind the policy at either the global level or to virtual servers. When you bind at least one authentication policy to a virtual server, any authentication policies that you bound to the global level are not used when users log on to the virtual server, unless the global authentication type has a higher precedence than the policy bound to the virtual server.

When a user logs on to the appliance, authentication is evaluated in the following order:

- The virtual server is checked for any bound authentication policies.
- If authentication policies are not bound to the virtual server, the appliance checks for global authentication policies.
- If an authentication policy is not bound to a virtual server or globally, the user is authenticated through the default authentication type.

If you configure LDAP and RADIUS authentication policies and want to bind the policies globally for two-factor authentication, you can select the policy in the configuration utility and then select if the policy is the primary or secondary authentication type. You can also configure a group extraction policy.

> **Note:**
>
> The Citrix ADC or the Citrix Gateway appliance encodes only UTF-8 characters for authentication, and it is not compatible with servers that use ISO-8859-1 characters.

Create an authentication policy

Create an authentication policy by using the GUI

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication**, and then select the type of policy that you want to create.
   For Citrix Gateway, navigate to **Citrix Gateway > Policies > Authentication**.
2. In the details pane, on the **Policies** tab, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the action, and then click **Edit**.
3. In the Create Authentication Policy or Configure Authentication Policy dialog, type or select the values for the parameters.
   - **Name** — policy name (Cannot be changed for a previously configured action)
   - **Authentication Type** — `authtype`

- **Server** — `authVsName`
- **Expression** — rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)

4. Click **Create** or **OK**. The policy that you created appears in the Policies page.
5. Click the **Servers** tab, and in the details pane do one of the following:
   - To use an existing server, select it, and then click.
   - To create a server, click Add, and follow the instructions.
6. If you want to designate this policy as a secondary authentication policy, on the Authentication tab, click Secondary. If you want to designate this policy as a primary authentication policy, skip this step.
7. Click **Insert Policy**.
8. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
9. In the **Priority** column to the left, modify the default priority to ensure that the policy is evaluated in the proper order.
10. Click **OK**. A message appears in the status bar, stating that the policy has been configured successfully.

Modify an authentication policy by using the GUI

You can modify configured authentication policies and profiles, such as the IP address of the authentication server or the expression.

1. In the configuration utility, on the Configuration tab, expand **Citrix Gateway > Policies > Authentication**.
   Note: You can also configure the policy from **Security > AAA - Application Traffic > Policies > Authentication**, and then select the type of policy that you want to modify.
2. In the navigation pane, under Authentication, select an authentication type.
3. In the details pane, on the Servers tab, select a server and then click Open.

Remove an authentication policy by using the GUI

If you changed or removed an authentication server from your network, remove the corresponding authentication policy from Citrix Gateway.

1. In the configuration utility, on the Configuration tab, expand **Citrix Gateway > Policies \ > Authentication**.

Note: To configure from ADC, navigate **Security > AAA - Application Traffic > Policies > Authentication**, and then select the type of policy that you want to remove.

2. In the navigation pane, under Authentication, select an authentication type.
3. In the details pane, on the Policies tab, select a policy and then click Remove.

Create an authentication policy by using the CLI

At the command prompt, type the following commands:

```
add authentication negotiatePolicy <name> <rule> <reqAction>

show authentication localPolicy <name>

bind authentication vserver <name> -policy <policyname> [-priority <
    priority>][-secondary]]

show authentication vserver <name>
<!--NeedCopy-->
```

Example:

```
    > add authentication localPolicy Authn-Pol-1 ns_true
      Done
    > show authentication localPolicy
    1)      Name: Authn-Pol-1       Rule: ns_true          Request
        action: LOCAL   Done
    > bind authentication vserver Auth-Vserver-2 -policy Authn-Pol-1
    Done
    > show authentication vserver Auth-Vserver-2
        Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT State: UP
            Client Idle
        Timeout: 180 sec Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
        Authentication Domain: myCompany.employee.com
    1)  Primary authentication policy name:  Authn-Pol-1 Priority: 0
        Done
<!--NeedCopy-->
```

Modify an existing authentication policy by using the CLI

At the command prompt, type the following commands to modify an existing authentication policy:


```
set authentication localPolicy <name> <rule> [-reqaction <action>]<!--
    NeedCopy-->
```

```
Example

<!--NeedCopy-->
```

set authentication localPolicy Authn-Pol-1 'ns_true'

```
### Remove an authentication policy by using the CLI

At the command prompt, type the following command to remove an
    authentication policy:

<!--NeedCopy-->
```

rm authentication localPolicy

```
Example

<!--NeedCopy-->
```

rm authentication localPolicy Authn-Pol-1

```
### Bind an authentication policy

After you configure the authentication policies, you bind the policy
    either globally or to a virtual server. You can use either the
    configuration utility to bind an authentication policy.

To bind an authentication policy globally by using the configuration
    utility:

1.  In the configuration utility, on the Configuration tab, expand **
    Citrix Gateway \> Policies \ > Authentication**.
     Note: To configure from ADC, navigate **Security > AAA -
        Application Traffic > Policies > Authentication**
1.  Click an authentication type.
1.  In the details pane, on the Policies, tab, click a server and then
    in Action, click Global Bindings.
1.  On the Primary or Secondary tab, under Details, click Insert Policy
    .
1.  Under Policy Name, select the policy and then click OK.

```



```
    **Note:** When you select the policy, Citrix Gateway sets the
       expression to True value automatically.

To unbind a global authentication policy by using the configuration
   utility:

1.  In the configuration utility, on the Configuration tab, expand **
   Citrix Gateway \> Policies \ > Authentication**.
    Note: To configure from ADC, navigate **Security > AAA -
       Application Traffic > Policies > Authentication**
1.  On the Policies tab, in Action, click Global Bindings.
1.  In the Bind/Unbind Authentication Policies to Global dialog box, on
    the Primary or Secondary tab, in Policy Name, select the policy,
   click Unbind Policy, and then click OK.

## Add an authentication action

### Add an authentication action by using the command line interface

If you do not use LOCAL authentication, you need to add an explicit
   authentication action. At the command prompt, type the following
   command:

<!--NeedCopy-->
```

add authentication tacacsAction -serverip [-serverPort][-authTimeout][...]

```
Example

<!--NeedCopy-->
```

add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812 -authtimeout 15 -tacacsSecret “minotaur” -authorization OFF -accounting ON -auditFailedCmds OFF -defaultAuthenticationGroup “users”

```
### Configure an authentication action by using the command line
   interface

To configure an existing authentication action, at the command prompt,
   type the following command:

<!--NeedCopy-->
```

set authentication tacacsAction -serverip [-serverPort][-authTimeout][...]


```
Example

<!--NeedCopy-->
```

set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812 -authtimeout 15 -tacacsSecret “minotaur” -authorization OFF -accounting ON -auditFailedCmds OFF -defaultAuthenticationGroup “users”

```
### Remove an authentication action by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

<!--NeedCopy-->
```

rm authentication radiusAction

```
Example

<!--NeedCopy-->
```

rm authentication tacacsaction Authn-Act-1

```
## The noAuth authentication

Citrix ADC appliance supports noAuth authentication capability that enables the customer to configure a defaultAuthenticationGroup parameter in the `noAuthAction` command, when a user performs this policy. The administrator can check for the presence of this group in user's group to determine the user's navigation through the noAuth policy.

### To configure a noAuth authentication by using the command line interface

At the command prompt, type;

<!--NeedCopy-->
```

add authentication noAuthAction [-defaultAuthenticationGroup]

```
**Example:**

<!--NeedCopy-->
```


add authentication noAuthAction noauthact –defaultAuthenticationGroup mynoauthgroup

```
## Default global authentication types

When you installed Citrix Gateway and ran the Citrix Gateway wizard, you configured authentication within the wizard. This authentication policy is bound automatically to the Citrix Gateway global level. The authentication type you configure within the Citrix Gateway wizard is the default authentication type. You can change the default authorization type by running the Citrix Gateway wizard again or you can modify the global authentication settings in the configuration utility.

If you need to add other authentication types, you can configure authentication policies on Citrix Gateway and bind the policies to Citrix Gateway by using the configuration utility. When you configure authentication globally, you define the type of authentication, configure the settings, and set the maximum number of users that can be authenticated.

After configuring and binding the policy, you can set the priority to define which authentication type takes precedence. For example, you configure LDAP and RADIUS authentication policies. If the LDAP policy has a priority number of 10 and the RADIUS policy has a priority number of 15, the LDAP policy takes precedence, regardless of where you bind each policy. This is called cascading authentication.

You can select to deliver logon pages from the Citrix Gateway in-memory cache or from the HTTP server running on Citrix Gateway. If you choose to deliver the logon page from the in-memory cache, the delivery of the logon page from Citrix Gateway is faster than from the HTTP server. Choosing to deliver the logon page from the in-memory cache reduces the wait time when many users log on at the same time. You can only configure the delivery of logon pages from the cache as part of a global authentication policy.

You can also configure the network address translation (NAT) IP address that is a specific IP address for authentication. This IP address is unique for authentication and is not the Citrix Gateway subnet, mapped, or virtual IP addresses. This is an optional setting.

**Note:**
>
>You cannot use the Citrix Gateway wizard to configure SAML
```



```
    authentication.
16
17 You can use the Quick Configuration wizard to configure LDAP, RADIUS,
    and client certificate authentication. When you run the wizard, you
    can select from an existing LDAP or RADIUS server configured on
    Citrix Gateway. You can also configure the settings for LDAP or
    RADIUS. If you use two-factor authentication, Citrix recommends
    using LDAP as the primary authentication type.
18
19 ### Configure default global authentication types
20
21 1.  In the configuration utility, on the Configuration tab, in the
    navigation pane, expand Citrix Gateway and then click Global
    Settings.
22 1.  In the details pane, under Settings, click Change authentication
    settings.
23 1.  In Maximum Number of Users, type the number of users who can be
    authenticated by using this authentication type.
24 1.  In NAT IP address, type the unique IP address for authentication.
25 1.  Select Enable static caching to deliver logon pages faster.
26 1.  Select Enable Enhanced Authentication Feedback to provide a message
     to users if authentication fails. The message users receive include
     password errors, account disabled or locked, or the user is not
    found, to name a few.
27 1.  In Default Authentication Type, select the authentication type.
28 1.  Configure the settings for your authentication type and then click
    OK.
29 <!--NeedCopy-->
```

Users and groups

September 14, 2021

After configuring the authentication, authorization, and auditing basic setup, you create users and groups. You first create a user account for each person who authenticates via the Citrix ADC appliance. If you are using local authentication controlled by the Citrix ADC appliance itself, you create local user accounts and assign passwords to each of those accounts.

You also create user accounts on the Citrix ADC appliance if you are using an external authentication server. In this case, however, each user account must exactly match an account for that user on the external authentication server, and you do not assign passwords to the user accounts that you create on the Citrix ADC. The external authentication server manages the passwords for users that authenticate

with the external authentication server.

If you are using an external authentication server, you can still create local user accounts on the Citrix ADC appliance if, for example, you want to allow temporary users (such as visitors) to log in but do not want to create entries for those users on the authentication server. You assign a password to each local user account, just as you would if you were using local authentication for all user accounts.

Each user account must be bound to policies for authentication and authorization. To simplify this task, you can create one or more groups and assign user accounts to them. You can then bind policies to groups instead of individual user accounts.

Configure policies with groups

After you configure groups, you can use the **Group** dialog box to apply policies and settings that specify user access. If you are using local authentication, you create users and add them to groups that are configured on Citrix Gateway. The users then inherit the settings for that group.

You can configure the following policies or settings for a group of users in the **Group** dialog box:

- Users
- Authorization policies
- Auditing policies
- Session policies
- Traffic policies
- Bookmarks
- Intranet applications
- Intranet IP addresses

In your configuration, you might have users that belong to more than one group. In addition, each group might have one or more bound session policies, with different parameters configured. Users that belong to more than one group inherit the session policies assigned to all the groups to which the user belongs. To ensure which session policy evaluation takes precedence over the other, you must set the priority of the session policy.

For example, you have group1 that is bound with a session policy configured with the home page www.homepage1.com. Group2 is bound with a session policy configured with home page www.homepage2.com. When these policies are bound to respective groups without a priority number or with a same priority number, the home page that appears to users who belong to both the groups depends on which policy is processed first. By setting a lower priority number, which gives higher precedence, for the session policy with home page www.homepage1.com, you can ensure that users who belong to both the groups receive the home page www.homepage1.com.

If session policies do not have a priority number assigned or have the same priority number, precedence is evaluated in the following order:

- User
- Group
- Virtual server
- Global

If policies are bound to the same level, without a priority number or if the policies have the same priority number, the order of evaluation is per the policy bind order. Policies that are bound first to a level receive precedence over policies bound later.

If we have a user bound to multiple groups with each group having IIP bound, the user can get free IP from any of the bound groups.

Create users and groups

Configure authentication, authorization, and auditing local users by using the GUI

1. Navigate to **Security > AAA - Application Traffic > Users**
   From Citrix Gateway, expand **Citrix Gateway > User Administration**, and then click **AAA Users**.
2. In the details pane, do one of the following:
   - To create a new user account, click **Add**.
   - To modify an existing user account, select the user account, and then click **Open**.
3. In the **Create AAA User** dialog box, in the **User Name** text box, type a name for the user.
4. If creating a locally authenticated user account, clear the **External Authentication** check box and provide a local password that the user uses to log on.
5. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the user has been configured successfully.

Configure authentication, authorization, and auditing local groups and add users to them by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Groups**
   From Citrix Gateway, expand **Citrix Gateway > User Administration**, and then click **AAA Groups**.
2. In the details pane, do one of the following:
   - To create a new group, click **Add**.
   - To modify an existing group, select the group, and then click **Edit**.
3. If you are creating a new group, in the **Create AAA Group** dialog box, in the **Group Name** text box, type a name for the group.

4. In the **Advanced** area to the right, click **AAA Users**.
   - To add a user to the group, select the user, and then click **Add**.
   - To remove a user from the group, select the user, and then click **Remove**.
   - To create a new user account and add it to the group, click the **Plus** icon, and then follow the instructions in "To configure authentication, authorization, and auditing local users by using the configuration utility."
5. Click **Create** or **OK**. The group that you created appears in the **AAA Groups** page.

Delete a group by using the GUI

You can also delete user groups from Citrix Gateway.

1. Navigate to **Security > AAA - Application Traffic > Groups**
   From Citrix Gateway, expandCitrix **Gateway > User Administration,** and then click **AAA Groups**.
   In the details pane, select the group, and then click Remove.

Configure authentication, authorization, and auditing local users by using the CLI

At the command prompt, type the following commands:

```
add aaa group <groupname>

bind aaa group <groupname> -username <username>
<!--NeedCopy-->
```

Example:

```
add aaa group group-2

bind aaa group group-2 -username user-2
<!--NeedCopy-->
```

Remove users from an authentication, authorization, and auditing group by using the command line interface

At the command prompt, unbind users from the group by typing the following command once for each user account that is bound to the group:

```
unbind aaa group <groupname> -username <username><!--NeedCopy-->
```



```
**Example:**

<!--NeedCopy-->
```

unbind aaa group group-hr -username user-hr-1

```
### Remove an authentication, authorization, and auditing group by
    using the command line interface

First remove all users from the group. Then, at the command prompt,
    type the following command to remove a Citrix ADC AAA group and
    verify the configuration:

<!--NeedCopy-->
```

rm aaa group

```
**Example:**

<!--NeedCopy-->
```

rm aaa group group-hr

```
> **Note**
>
>You cannot add a user name with domain if the user name is already
    added without domain. If the user name with domain is added first
    followed by the same user name without domain, then the Citrix ADC
    appliance adds the user name to the user list.

The following example shows adding a user name with domain is not
    permitted if the same user name is added without domain.

<!--NeedCopy-->
```

> add aaa user u47985
> Done
> show aaa users
> 1) UserName: u47985
> Done
> add aaa user u47985@domain.com

> ERROR: User already exists
> ‘‘‘

The following example shows if the user name with domain is added first followed by the same user name without domain, then the Citrix ADC appliance adds the user name to the user list.

```
> add aaa user u47985@domain.com
Done
> add aaa user u47985
Done
> sh aaa user
1)   UserName: u47985@domain.com
2)   UserName: u47985
```

‘‘‘

Authentication methods

September 14, 2021

The Citrix ADC appliance can authenticate users with local user accounts or by using an external authentication server. The appliance supports the following authentication types:

- **LOCAL:** Authenticates to the Citrix ADC appliance by using a password, without reference to an external authentication server. User data is stored locally on the Citrix ADC appliance.
- **RADIUS:** Authenticate to an external RADIUS server.
- **LDAP:** Authenticates to an external LDAP authentication server.
- **TACACS:** Authenticates to an external Terminal Access Controller Access-Control System (TACACS) authentication server.
- **CERT:** Authenticates to the Citrix ADC appliance by using a client certificate, without reference to an external authentication server.
- **NEGOTIATE:** Authenticates to a Kerberos authentication server. If there is an error in Kerberos authentication, Citrix ADC uses NTLM authentication.
- **SAML:** Authenticates to a server that supports the Security Assertion Markup Language (SAML).
- **SAML IDP:** Configures the Citrix ADC to serve as a Security Assertion Markup Language (SAML) Identity Provider (IdP).
- **WEB:** Authenticates to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that the user authentication was successful.

- **Native OTP:** Citrix ADC appliance supports one-time passwords (OTPs) without having to use a third-party server.
- **Push notification:** Citrix Gateway supports push notifications for OTP. Users do not have to manually enter the OTP received on their registered devices to log in to Citrix Gateway. Admins can configure Citrix Gateway such that login notifications are sent to users' registered devices using push notification services.
- **Email OTP:** The Email OTP method enables you to authenticate using the one-time password (OTP) that is sent to the registered email address. When you try to authenticate on any service, the server sends an OTP to the registered email address of the user.
- **reCaptcha authentication** - Citrix Gateway supports a new first class action 'captchaAction' that simplifies reCaptcha configuration. As reCaptcha is a first class action, it can be a factor of its own. You can inject reCaptcha anywhere in the nFactor flow.
- **nFactor authentication:** Multifactor authentication enhances the security of an application by requiring users to provide multiple proofs of identify to gain access. The Citrix ADC appliance provides an extensible and flexible approach to configuring multifactor authentication. This approach is called nFactor authentication.
- **OAuth authentication:** OAuth authentication authorizes and authenticates users to services that are hosted on applications such as Google, Facebook, and Twitter.

nFactor authentication

September 14, 2021

> **Important**
>
> - nFactor authentication is supported from NetScaler 11.0 Build 62.x onwards.
> - For nFactor authentication to work with Citrix ADC, an Advanced license or a Premium license is required.
> - Starting from release 13.0 build 67.x, nFactor authentication is supported with Standard license only for Gateway/VPN virtual server. For more information about nFactor authentication with Citrix Gateway, see nFactor for Gateway Authentication.
> - nFactor authentication is not supported for Linux client.

Multifactor authentication enhances the security of an application by requiring users to provide multiple proofs of identify to gain access. The Citrix ADC appliance provides an extensible and flexible approach to configuring multifactor authentication. This approach is called *nFactor authentication*.

How nFactor authentication works

Each authentication factor performs the following tasks:

- Collects credentials from the user. Citrix ADC supported authentication mechanisms include LDAP, RADIUS, SAML assertion, Client Certificate, OAuth OpenID Connect, Kerberos, and so on.
- Evaluates the supplied credentials to decide whether the authentication succeeded, failed or the actions like Group extraction, Attribute extraction is to be performed.
- Based on the evaluation results, access is either granted, denied, or a next factor is selected.
- Repeat these steps, until there are no more next factors to evaluate.

With nFactor authentication you can:

- Configure any number of authentication factors.
- Base the selection of the next factor on the result of executing the previous factor.
- Customize the login interface. For example, you can customize the label names, error messages, and help text.
- Extract user group information without doing authentication.
- Configure pass-through for an authentication factor. This means that no explicit login interaction is required for that factor.
- Configure the order in which different types of authentication are applied. Any of the authentication mechanisms that are supported on the Citrix ADC appliance can be configured as any factor of the nFactor authentication setup. These factors are executed in the order in which they are configured.
- Configure the Citrix ADC to proceed to an authentication factor that must be executed when authentication fails. To do so, you configure another authentication policy with the exact same condition, but with the next highest priority and with the action set to “NO_AUTH”. You must configure the next factor, which must specify the alternative authentication mechanism to apply.

Encryption of Citrix Gateway login information for nFactor authentication

Citrix Gateway with nFactor authentication can encrypt the login request fields submitted by a client (browser or SSO apps) during authentication process. The encrypted login request fields provide an extra layer of security to protect the user’s sensitive data from being disclosed.

Compatible browsers

The following table lists the browsers along with version details that support login encryption.

Browsers	Version
Chrome	78 and above
Firefox	69 and above
Internet Explorer	11
Edge	42 and above
Safari	11.0 and above
Opera	66

Compatible clients

The following section lists the clients along with version details that support encryption of Citrix Gateway login information.

- Citrix Workspace app in Mac supports encryption only when OS version is 10.14.x and above.
- Citrix SSO app in Mac supports encryption only when OS version is 10.14.x and above.
- Windows SSO app does not have restrictions on the compatibility.
- Password encryption in Citrix Workspace app for Windows clients is supported only in Internet Explorer 11 version.

To enable the login encryption by using the CLI

At the command prompt, type:

```
set aaa parameter [-loginEncryption (ENABLED | DISABLED)]
```

> **Note**
>
> The loginEncryption parameter is DISABLED by default. You must ENABLE it.

To enable the login encryption by using the GUI

1. Navigate to **Security > AAA – Application Traffic**, click **Change authentication AAA settings** under **Authentication Settings** section.
2. On the **Configure AAA Parameter** page, scroll down to the **Login Encryption** option, and enable it.

nFactor concepts, entities, and terminology

September 14, 2021

This topic captures some of the major entities involved in nFactor authentication and their significance.

Login schema

nFactor decouples the 'view', the user interface, with the 'model' that is the runtime handling. nFactor's view is defined by login schema'. Login schema is an entity that defines what user sees and specifies how to extract the data from user.

For defining view, login schema points to a file on disk that defines the logon form. This file should be according to the specification of "Citrix Common Forms Protocol." This file is essentially an XML definition of the logon form.

In addition to the XML file, login schema contains advanced policy expressions to glean user name and password from the user's login request. These expressions are optional, and can be omitted if user name and password from user arrive with expected form variable names.

Login schema also defines, whether current set of credentials should be used as default SingleSignOn credentials.

Policy label

A policy label is a collection of policies. It is a construct not alien to Citrix ADC's policy infrastructure. Policy label defines an authentication factor. That is, it contains all the policies necessary to determine whether credentials from user are satisfied. All the policies in a policy label can be assumed as homogenous. Policy label for authentication cannot take policies of different type, say rewrite. To put in a different way, all the policies in a policy label validate same password/credential from user, mostly. The result of policies in a policyLabel follows logical OR condition. Hence, if the authentication specified by first policy succeeds, other policies following it are skipped.

Policy label can be created by executing the following CLI command:

```
1 add authentication policy label mylabel  - loginSchema <>
2 <!--NeedCopy-->
```

A policy label takes login schema as the property. Login schema defines the view for that policy label. If login schema is not specified, an implicit login schema, LSCHEMA_INT, is associated with that policy label. Login schema decides whether a policy label becomes a passthrough or not.

Virtual server label

In Citrix ADC's advanced policy infrastructure, a virtual server is also an implicit policy label. That's because virtual server can also be bound with more than one policy. However, a virtual server is special because it is the entry point for client traffic and can take policies of a different type. Each of the policies it put under its own label within the virtual server. Hence, virtual server is a conglomeration of labels.

Next factor

Whenever a policy is bound to a virtual server or a policy label, it can be specified with next factor. Next factor determines what should be done if a given authentication succeeds. If there is no next factor, that concludes authentication process for that user.

Each policy bound to a virtual server or policy label can have a different next factor. This allows for ultimate flexibility where in every policy's success can define a new path for user's authentication. Administrator can take advantage of this fact and craft clever fallback factors for users who do not meet certain policies.

No-Auth policy

nFactor introduces a special kind of built-in policy called NO_AUTHN. NO_AUTHN policy always returns success as authentication result. No-auth policy can be created by executing the following CLI command:

```
1 add authentication policy noauthpolicy  -rule <> -action NO_AUTHN
2 <!--NeedCopy-->
```

As per the command, no-authentication policy takes a rule that can be any advanced policy expression. Authentication result is always success from NO_AUTHN.

A no-auth policy in itself does not seem to add value. However, when used along with passthrough policy labels, it offers great flexibility to make logical decisions to drive user authentication flow. NO_AUTHN policy and passthrough factors offer a new dimension to nFactor's flexibility.

Note: Check the examples that depict the usage of no-auth and passthrough in subsequent sections.

Passthrough factor/label

Once the user has passed the authentication at virtual server (for first factor), subsequent authentications happen at policy labels or user defined (secondary) factors.
Every policy label/factor is associated with a login schema entity to display view for that factor. This allows for customizing views based on the path user would have taken to arrive at a given factor.

There are specialized kinds of policy labels which do not point explicitly to a login schema. Specialized policy labels point to a login schema that does not actually point to the XML file for the view. These policy labels/factors are called 'passthrough' factors.

Passthrough factors can be created by executing the following CLI commands:

Example 1:

```
1 add authentication policylabel example1
2 <!--NeedCopy-->
```

Example 2:

```
1 add loginschema passthrough_schema  -authenticationSchema noschema
2
3 add authentication policylabel example2  -loginschema
     passthrough_schema
4 <!--NeedCopy-->
```

Passthrough factor implies that authentication, authorization, and auditing subsystem should not go back to user to get credential set for that factor. Instead, it is a hint for authentication, authorization, and auditing to continue with already obtained credentials. This is useful in cases where user intervention is not desired. For example,

- When user is presented two password fields. After the first factor, second factor does not need user intervention
- When authentication of a type (say certificate) is done, and administrator needs to extract groups for that user.

Passthrough factor can be used with NO_AUTH policy to make conditional jumps.

nFactor authentication flow

Authentication always begins at virtual server in nFactor. Virtual server defines the first factor for the user. The first form that the user sees is served by the virtual server. The logon form that user sees can be customized at virtual server using login schema policies. If there are no login schema policies, a single user name and password field are displayed to the user.

If more than one password fields must be displayed to the user on customized form, login schema policies must be used. They allow for displaying different forms based on the configured rules (such as intranet user versus external user, service provider A versus service provider B).

Once user credentials are posted, authentication begins at authentication virtual server, the first factor. Because authentication virtual server can be configured with multiple policies, each of them is evaluated in a sequence. At any given point, if an authentication policy succeeds, the next factor

specified against it is taken. If there is no next factor, the authentication process ends. If next factor exists, it is checked if that factor is a passthrough factor or a regular factor. If that is passthrough, authentication policies on that factor are evaluated without user intervention. Otherwise, login schema associated with that factor is displayed to the user.

Example of using passthrough factor and no-auth policies to make logical decisions

Administrator would like to decide nextFactor based on groups.

- Add authentication policylabel group check
- Add authentication policy admin group –rule http.req.user.is_member_of(“Administrators”) –action NO_AUTHN
- Add authentication policy nonadmins –rule true –action NO_AUTHN
- Bind authentication policy label group check –policy admingroup –pri 1 –nextFactor factor-for-admin
- Bind authentication policy label groupcheck –policy nonadmins –pri 10 –nextfactor factor-for-others
- Add authentication policy first_factor_policy –rule <> -action <>
- Bind authentication vserver <> -policy first_factor_policy –priority 10 –nextFactor groupcheck

Configuring nFactor authentication

September 14, 2021

You can configure multiple authentication factors using nFactor configuration rather than just two factors. nFactor configuration is supported only in Citrix ADC Advanced and Premium editions.

Methods to configure nFactor

You can configure nFactor authentication by one of the following methods:

- **nFactor Visualizer:** nFactor visualizer enables you to easily link factors or policy labels together in a single pane and also change the linking of the factors in the same pane. You can create an nFactor flow using the visualizer and bind that flow to a Citrix ADC AAA virtual server. For details about nFactor Visualizer and an example nFactor configuration using visualizer, see nFactor Visualizer for simplified configuration.
- **Citrix ADC GUI:** For details, see section **Configuration elements involved in nFactor configuration**.

- **Citrix ADC CLI:** For a sample snippet on nFactor configuration using the Citrix ADC CLI, see Sample snippet on nFactor configuration by using the Citrix ADC CLI.

> **Important:** This topic contains details about configuring nFactor by using the Citrix ADC GUI.

Configuration elements involved in nFactor configuration

The following elements are involved in configuring nFactor. For detailed steps, refer to the appropriate sections in this topic.

Configuration element	Tasks to be performed
AAA virtual server	Create a AAA virtual server
	Bind portal theme to AAA virtual server
	Enable client certificate authentication
Login schema	Configure a login schema profile
	Create and bind a login schema policy
Advanced authentication policies	Create advanced authentication policies
	Bind first factor advanced authentication policy to Citrix ADC AAA virtual server
	Use extracted LDAP groups to select the next authentication Factor
Authentication policy label	Create authentication policy label
	Bind authentication policy label
nFactor for Citrix Gateway	Create authentication profile to link a Citrix ADC AAA virtual server with Citrix Gateway virtual server
	Configure SSL parameters and CA certificate for Citrix Gateway
	Configure Citrix Gateway traffic policy for nFactor single sign-on to StoreFront

How nFactor works

When a user connects to Citrix ADC AAA or Citrix Gateway virtual server, the sequence of events that occur are as follows:

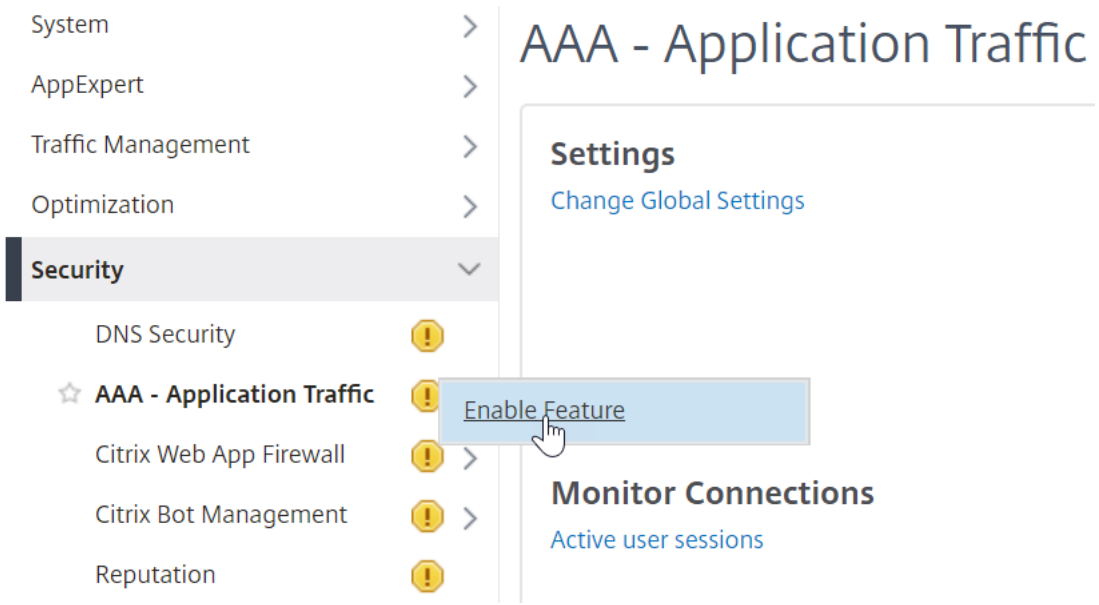
1. If forms-based authentication is used, the login schema bound to the Citrix ADC AAA virtual server is displayed.
2. Advanced authentication policies bound to the Citrix ADC AAA virtual server are evaluated.
    - If the advanced authentication policy succeeds, and if next factor (authentication policy label) is configured, next factor is evaluated. If Next Factor is not configured, then authentication is complete and successful.
    - If the advanced authentication policy fails, and if Goto Expression is set to Next, then next bound advanced authentication policy is evaluated. If none of the advanced authentication policies succeed, then authentication fails.
3. If the next factor authentication policy label has a Login Schema bound to it, it is displayed to the user.
4. The advanced authentication policies bound to the next factor authentication policy label is evaluated.
    - If the Advanced authentication policy succeeds, and if next factor (authentication policy label) is configured, next factor is evaluated.
    - If Next Factor is not configured, then authentication is complete and successful.
5. If the Advanced authentication policy fails, and if Goto Expression is Next, then the next bound advanced authentication policy is evaluated.
6. If none of the advanced authentication policies succeeds, then authentication fails.

AAA virtual server

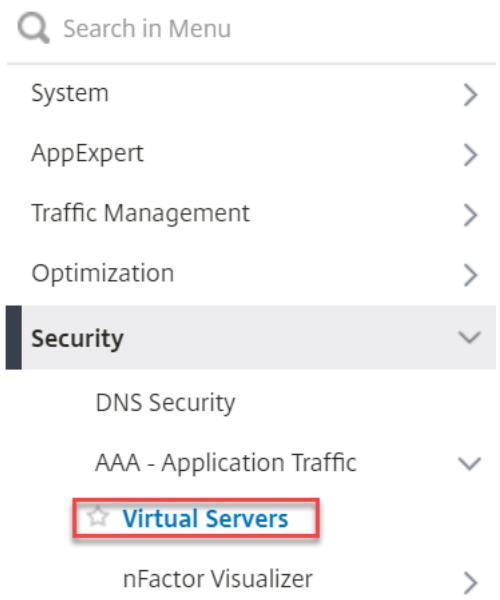
To use nFactor with Citrix Gateway, you first configure it on a AAA Virtual Server. Then you later link the AAA virtual server to the Citrix Gateway virtual server.

Create AAA Virtual Server

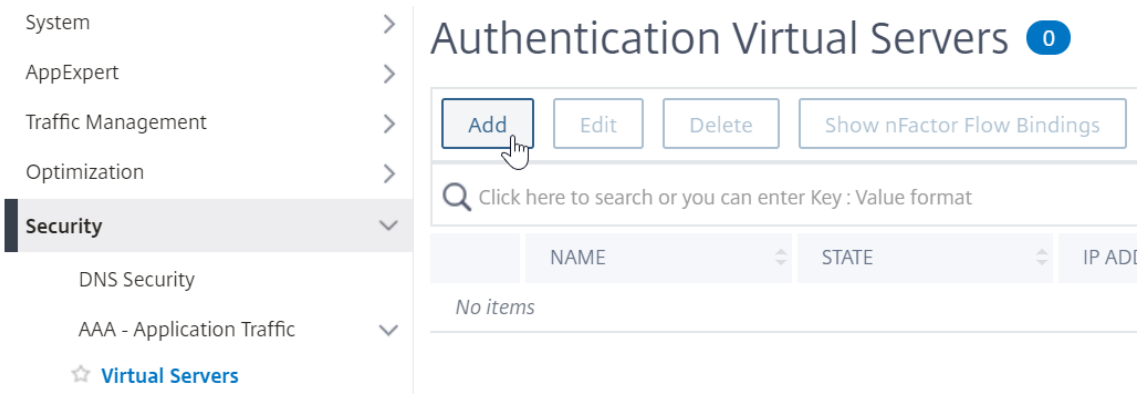
1. If AAA feature is not already enabled, navigate to, **Security > AAA – Application Traffic**, and right click to enable feature.



2. Navigate to **Configuration > Security > AAA - Application Traffic > Virtual Servers**.

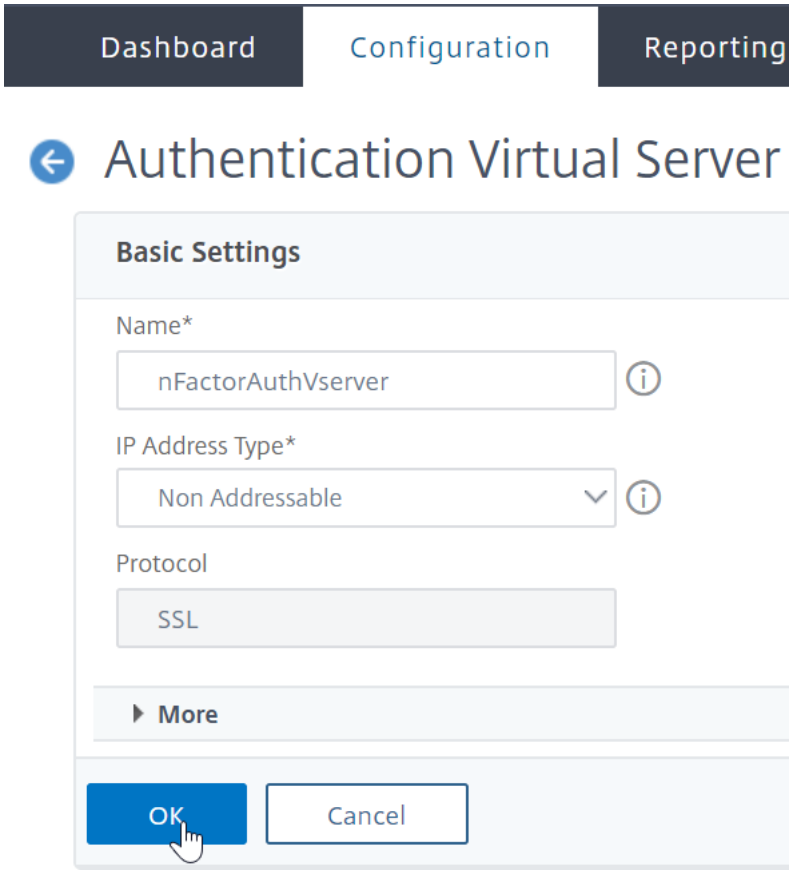


3. Click **Add** to create an authentication virtual server.

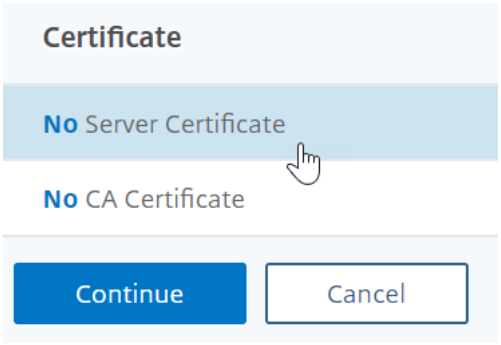


4. Enter the following information and click **OK**.

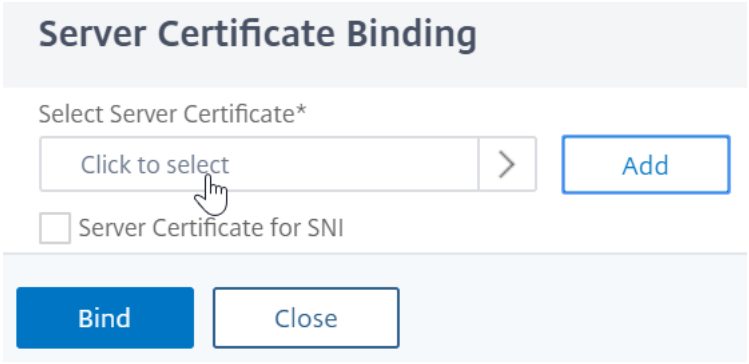
Parameter name	Parameter Description
Name	Name for the AAA virtual server.
IP address Type	Change the IP address Type to **Non Addressable** if this virtual server is used only for Citrix Gateway.



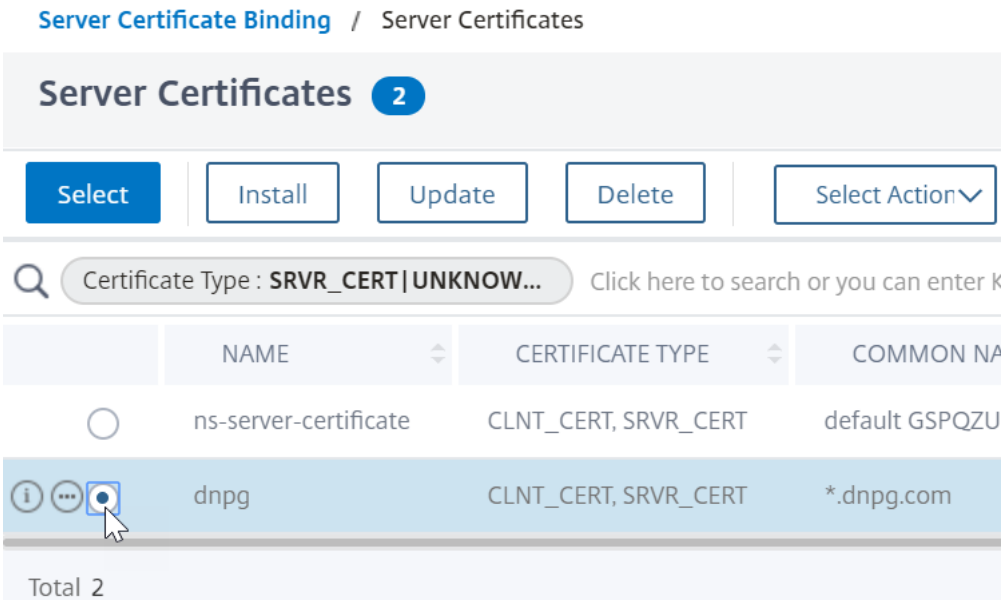
5. Under Certificate, select **No Server Certificate**.



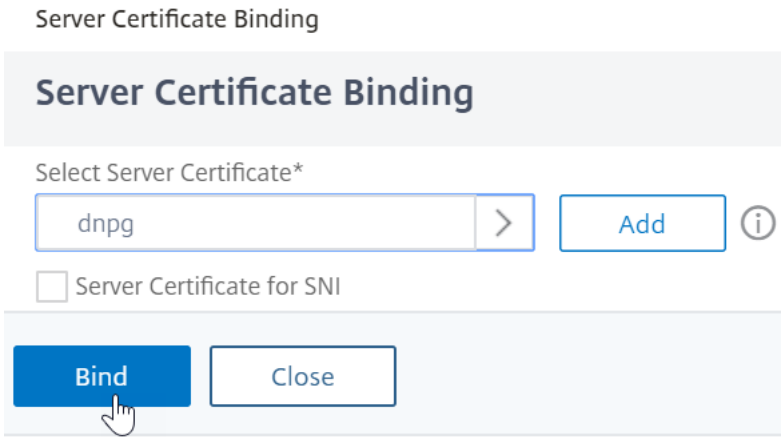
6. Click the text, **Click to select** to select the server certificate.



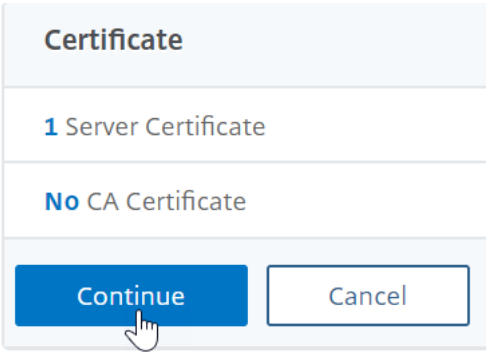
7. Click the radio button next to a certificate for the AAA Virtual Server, and click **Select**.The chosen certificate doesn't matter because this server is not directly accessible.



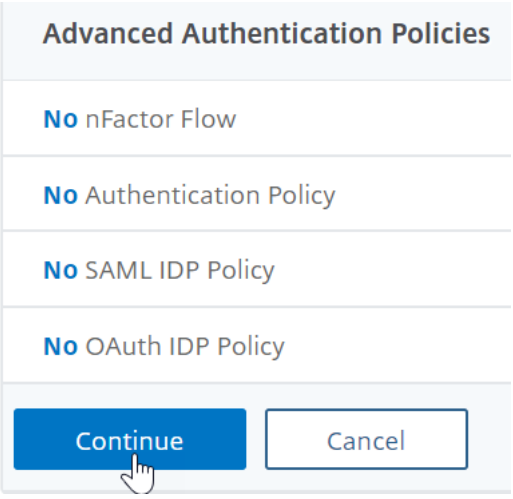
8. Click **Bind**.



9. Click **Continue** to close the **Certificate** section.

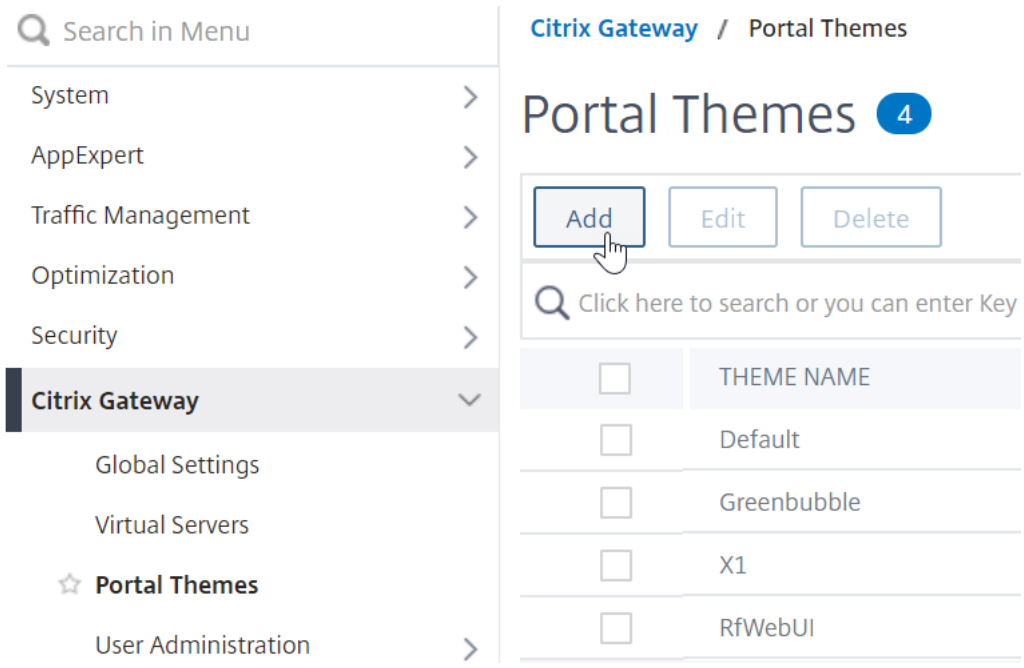


10. Click **Continue**.

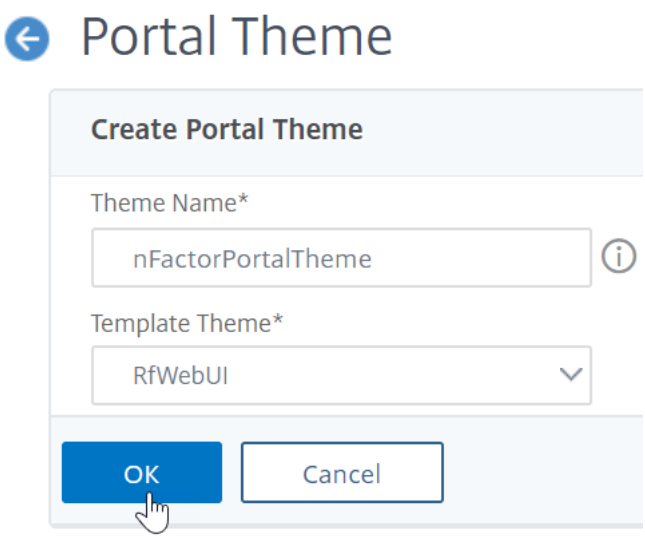


Bind portal theme to AAA virtual server

1. Navigate to **Citrix Gateway > Portal Themes**, and add a theme. You create the theme under Citrix Gateway, and then later bind it to the AAA virtual server.

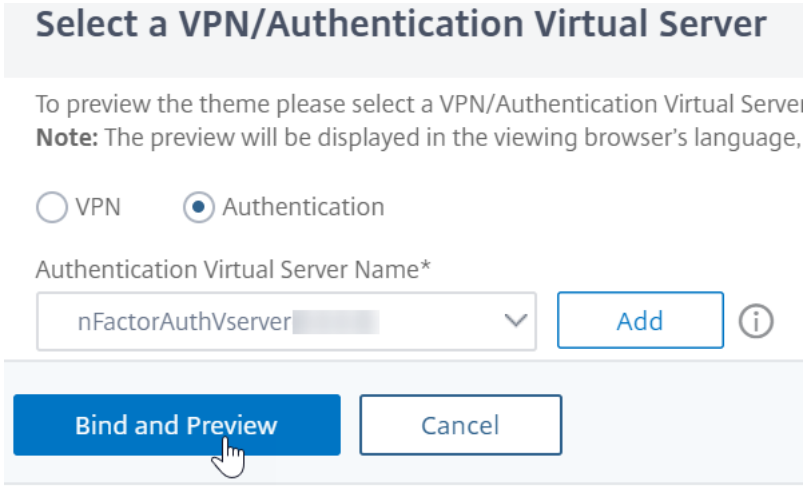


2. Create a theme based on the RfWebUI template theme.



3. After adjusting the theme as desired, at the top of the portal theme editing page, click **Click to Bind and View Configured Theme**.

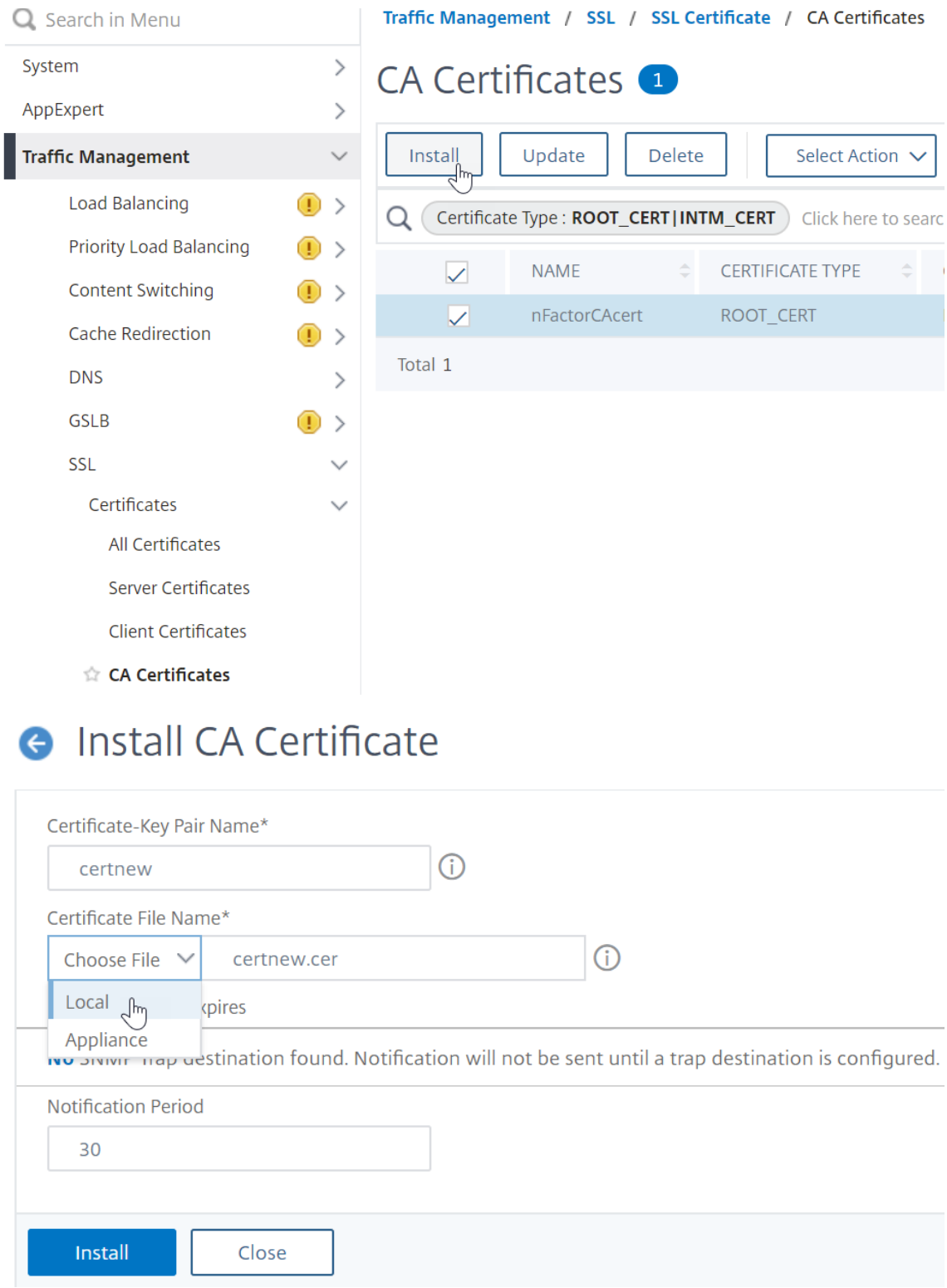
4. Change the selection to Authentication. From the **Authentication Virtual Server Name** drop-down menu, select the AAA Virtual Server, and click **Bind and Preview** and close the preview window.



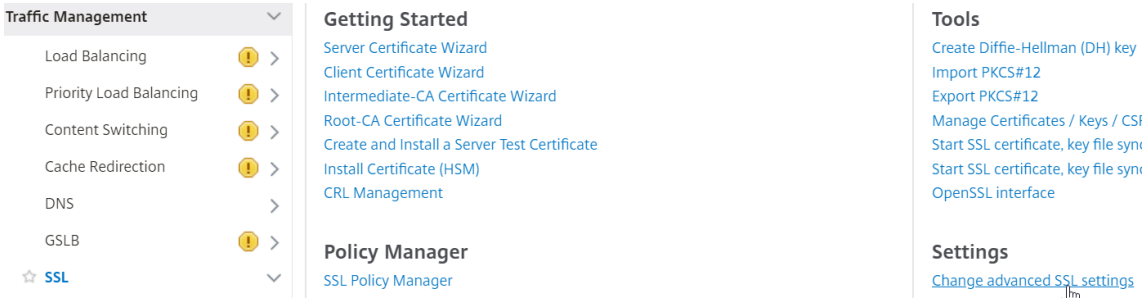
Enable client certificate authentication

If one of your authentication Factors is client certificate, then you must perform some SSL configuration on the AAA Virtual Server:

1. Navigate to **Traffic Management > SSL > Certificates > CA Certificates**, and install the root certificate for the issuer of the client certificates. Root certificates do not have a key file.



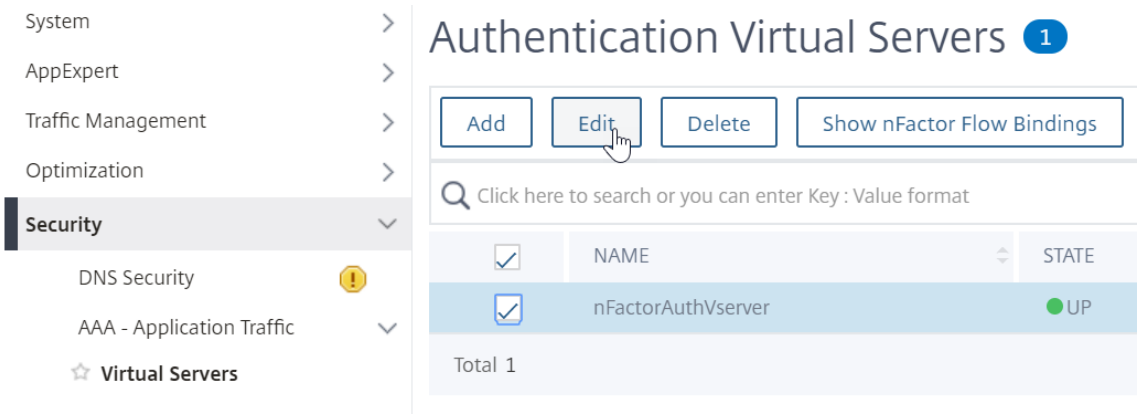
2. Navigate to **Traffic Management > SSL > Change advanced SSL settings**.



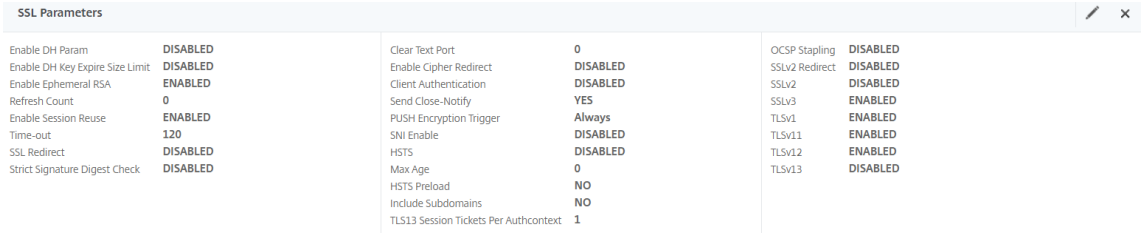
a. Scroll down to check whether **Default Profile** is **ENABLED**. If yes, then you must use an SSL Profile to enable Client Certificate Authentication. Otherwise, you can enable Client Certificate Authentication directly on the AAA Virtual Server in the SSL Parameters section.

3. If default SSL Profiles are not enabled:

a. Navigate to **Security > AAA - Application > Virtual Servers**, and edit an existing AAA virtual server.

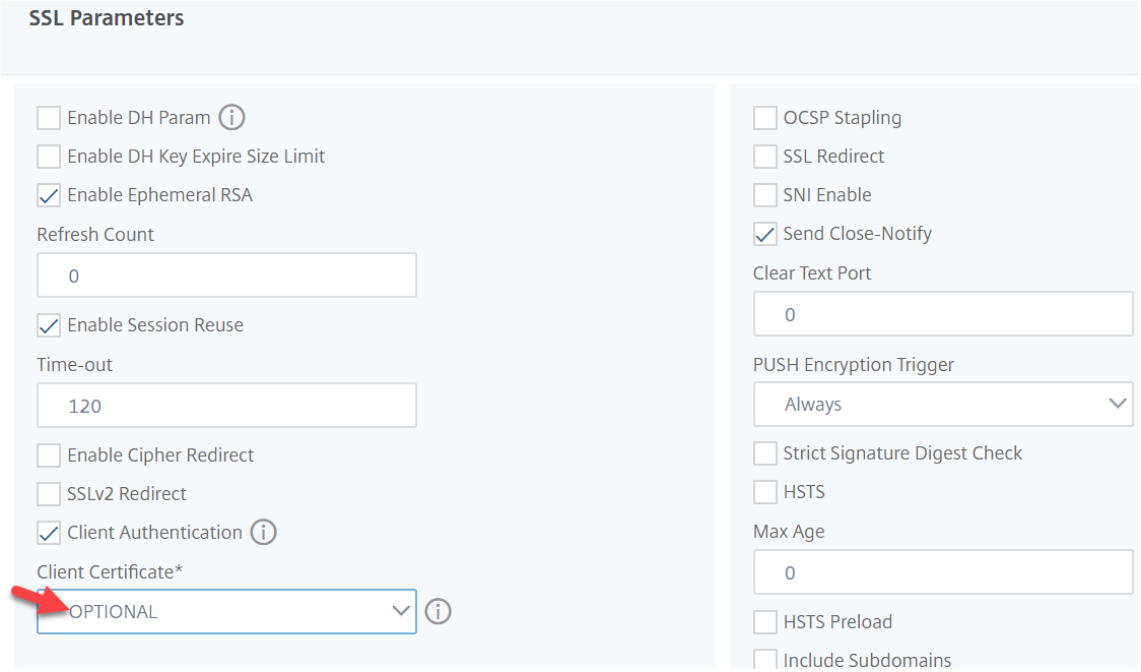


b. On the left, in the **SSL Parameters** section, click the pencil icon.



c. Check the box next to **Client Authentication**.

d. Make sure **Optional** is selected in the **Client Certificate** drop-down menu, and click **OK**.



4. If Default SSL Profiles are enabled, then create a new SSL Profile with Client Authentication enabled:

   a. On the left menu, expand System, and click Profiles.

   b. On the top right, switch to the SSL Profile tab.

   c. Right-click the ns_default_ssl_profile_frontend profile, and click Add. This copies settings from the default profile.

   d. Give the Profile a name. The purpose of this profile is to enable Client Certificates.

   e. Scroll down and find the Client Authentication checkbox. Check the box.

   f. Change the Client Certificate drop-down to OPTIONAL.

   g. Copying the default SSL Profile does not copy the SSL Ciphers so you'll have to redo them.

   h. Click Done when done creating the SSL Profile.

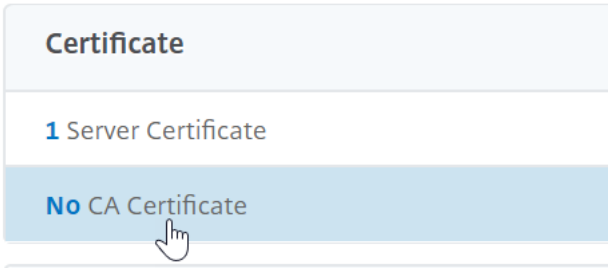
   i. Navigate to Security > AAA – Application Traffic > Virtual Servers, and edit a AAA vServer.

   j. Scroll down to the SSL Profile section and click the pencil.

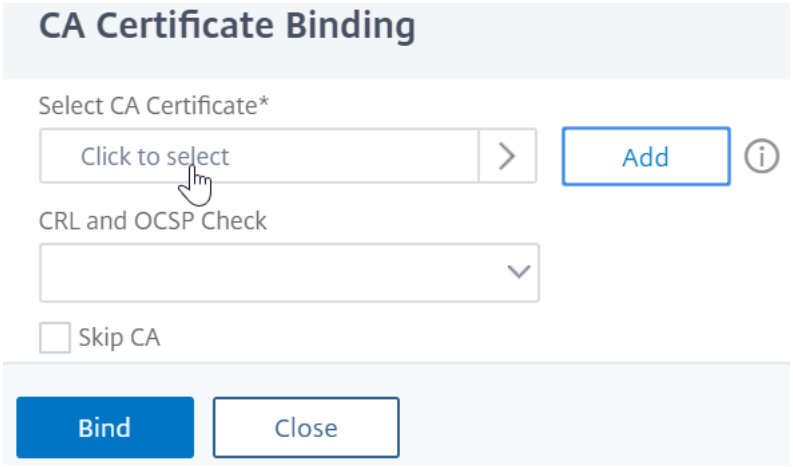
   k. Change the SSL Profile drop-down to the profile that has Client Certificates enabled. Click OK.

   l. Scroll down this article until you reach the instructions to bind the CA certificate.

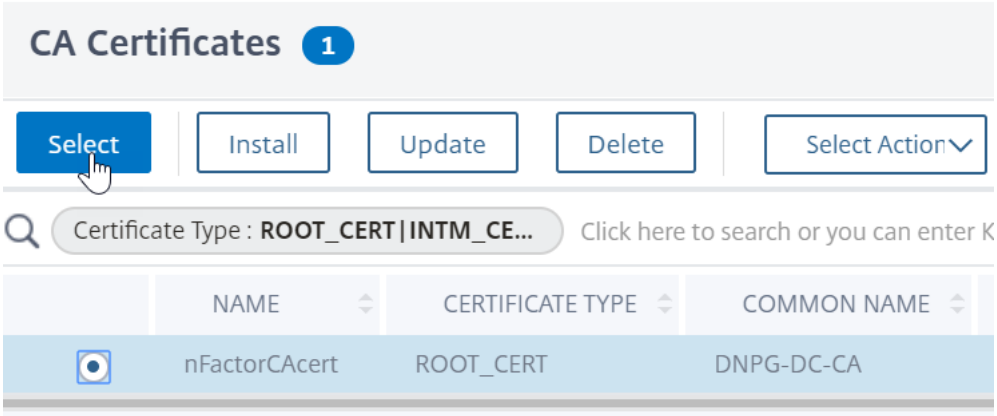
5. On the left, in the **Certificates** section, click where it says **No CA Certificate**.



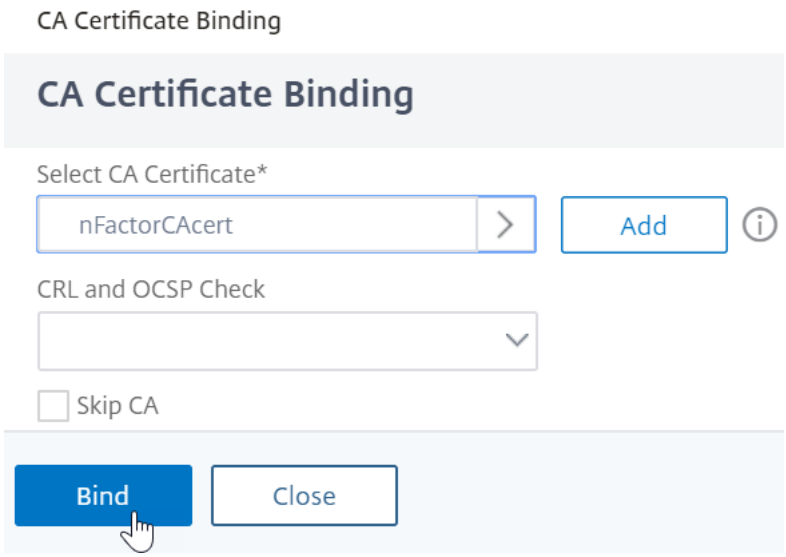
6. Click the text, **Click to select**.



7. Click the radio button next to the root certificate for the issuer of the client certificates, and click **Select**.



8. Click **Bind**.



Login schema XML file

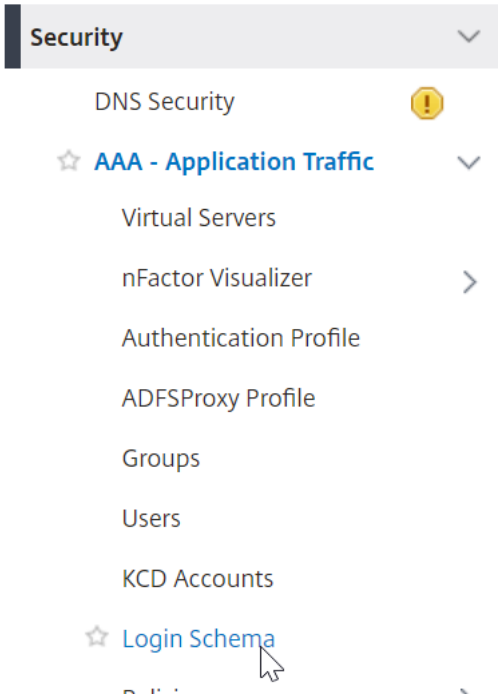
Login Schema is an XML file providing the structure of forms-based authentication logon pages.

nFactor implies multiple authentication Factors that are chained together. Each Factor can have different Login Schema pages/files. In some authentication scenarios, users could be presented with multiple logon screens.

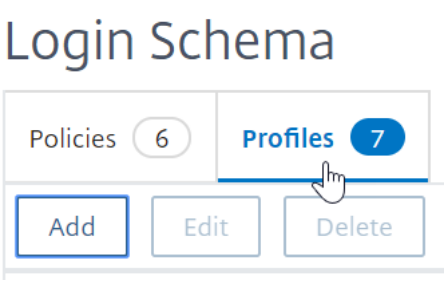
Configure a login schema profile

To configure a Login Schema Profile:

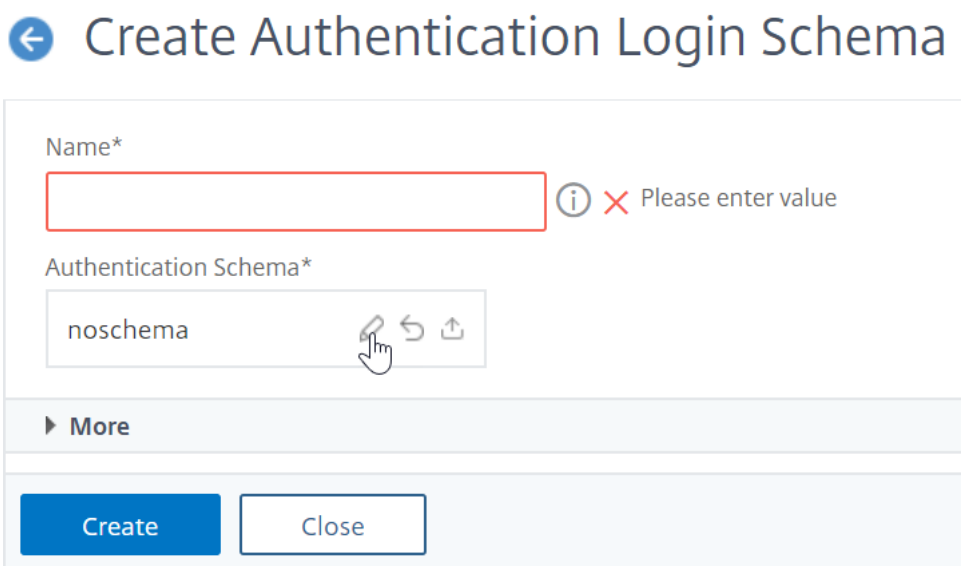
1. Create or Edit a Login Schema .XML file based on your nFactor design.
2. Navigate to **Security > AAA - Application Traffic > Login Schema**.



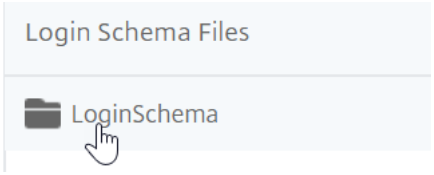
3. On the right, switch to the **Profiles** tab, and click **Add**.



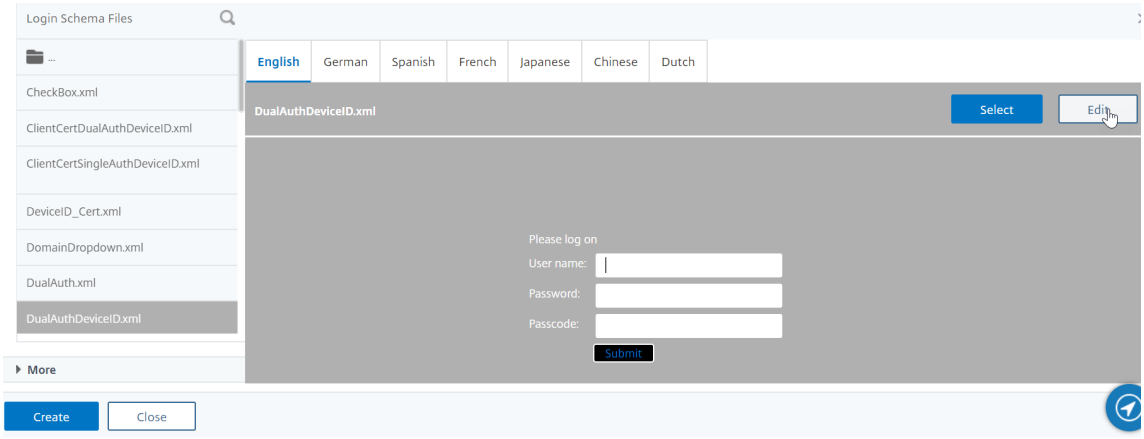
4. In the **Authentication Schema** field, click the pencil icon.



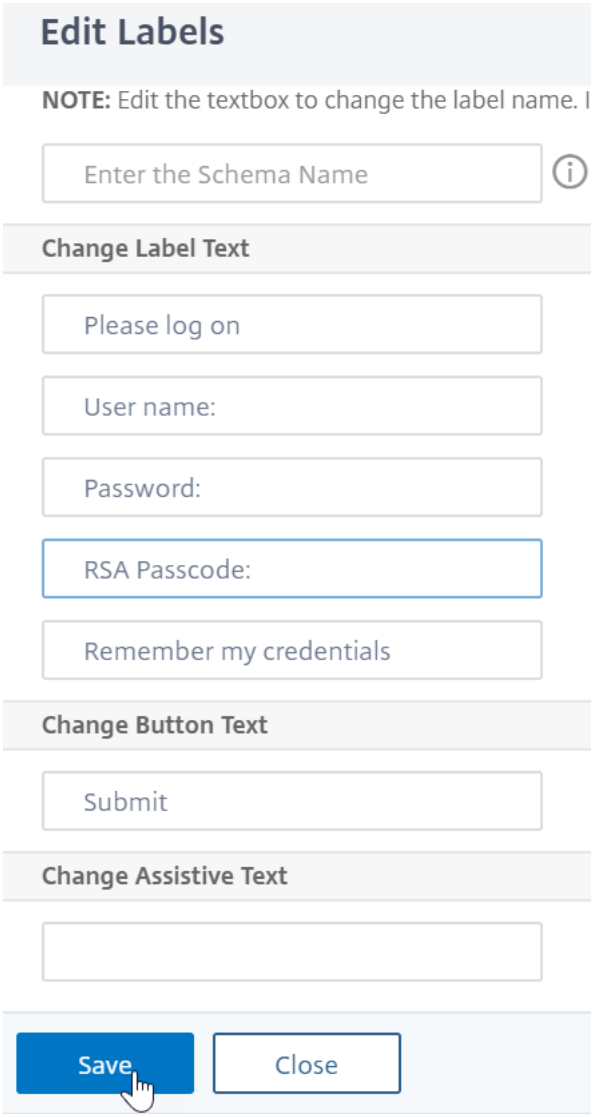
5. Click the LoginSchema folder to see the files in it.



6. Select one of the files. You can see a preview on the right. The labels can be changed by clicking the **Edit** button on the top right.

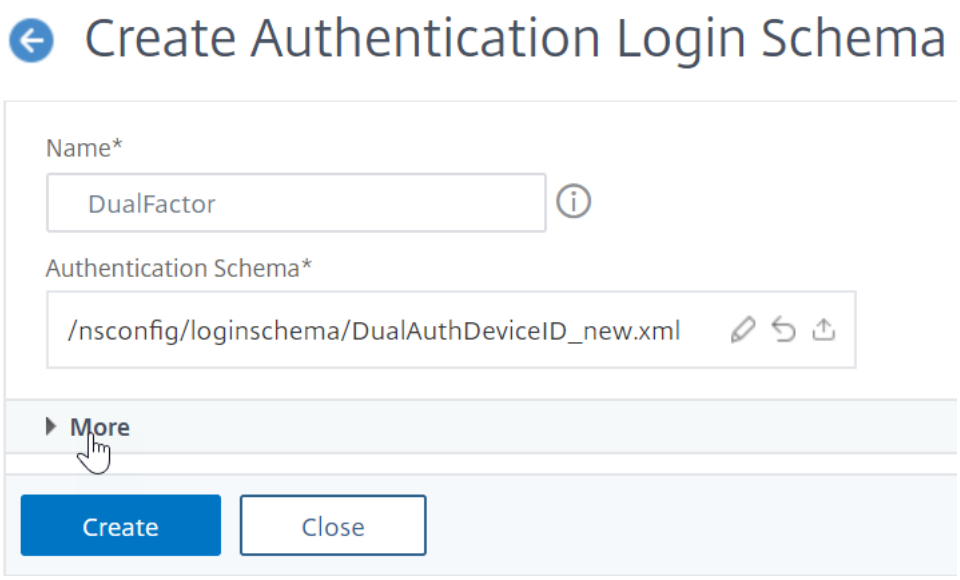


7. When you Save the changes, a new file is created under /nsconfig/LoginSchema.



8. On the top right, click **Select**.

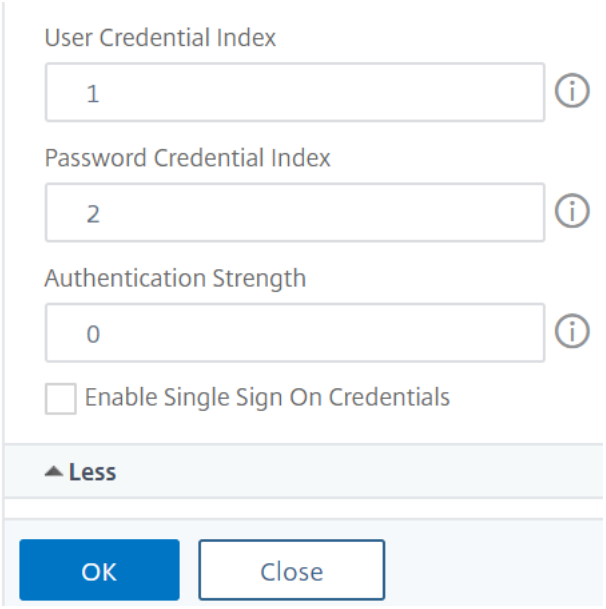
9. Give the Login Schema a name, and click **More**.



10. You might need to use the username and the password entered in the login schema for Single Sign-on (SSO) to a backend service, for example StoreFront.

    You can use the credentials entered in the login schema as your Single Sign-On credentials by using any of the following methods.

    - Click **More** at the bottom of the **Create Authentication Login Schema** page and select **Enable Single Sign On Credentials**.
    - Click **More** at the bottom of the **Create Authentication Login Schema** page and enter unique values for the user credential index and password credential index. These values can be between 1 and 16. Later you reference these index values in a traffic policy/profile by using the expression AAA.USER.ATTRIBUTE(#).



11. Click **OK** to create the login schema profile.

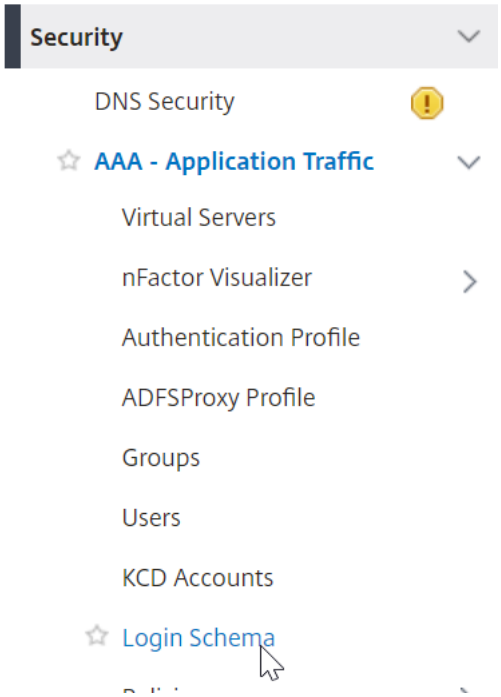
> **Note:** If you edit the login schema file (.xml) later, for changes to be reflected you need to edit the login schema profile and select the login schema (.xml) file again.

Create and bind a login schema policy

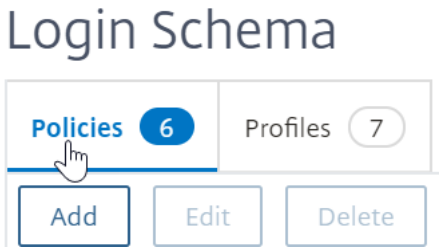
To bind a login schema profile to a AAA vServer, you must first create a login schema policy. Login schema policies are not required when binding the login schema profile to an authentication policy label, as detailed later.

To create and bind a login schema policy:

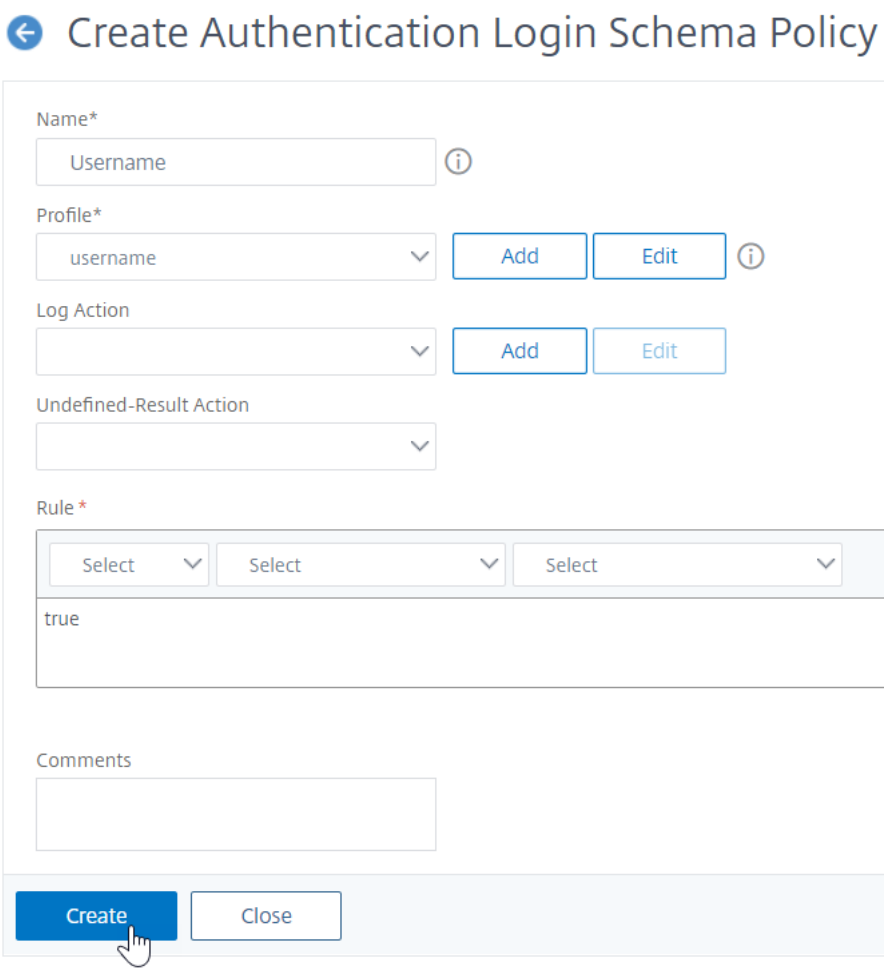
1. Navigate to **Security > AAA - Application Traffic > Login Schema**.



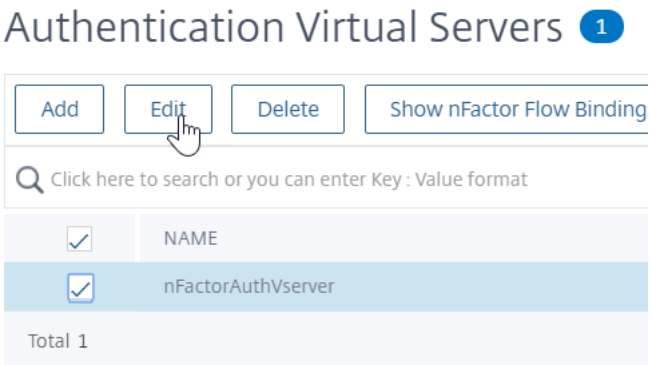
2. On the **Policies** tab, click **Add**.



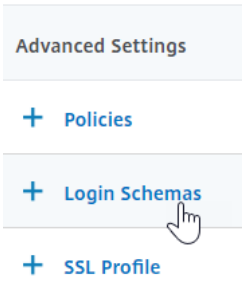
3. Use the **Profile** drop-down menu to select the Login Schema Profile you already created.
4. Enter a Default Syntax expression (e.g. true) in the **Rule** box, and click **Create**.



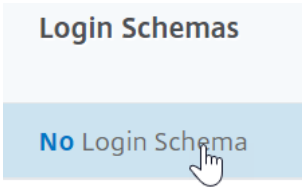
5. On the left, navigate to **Security > AAA - Application Traffic > Virtual Servers**, and edit an existing AAA Virtual Server.



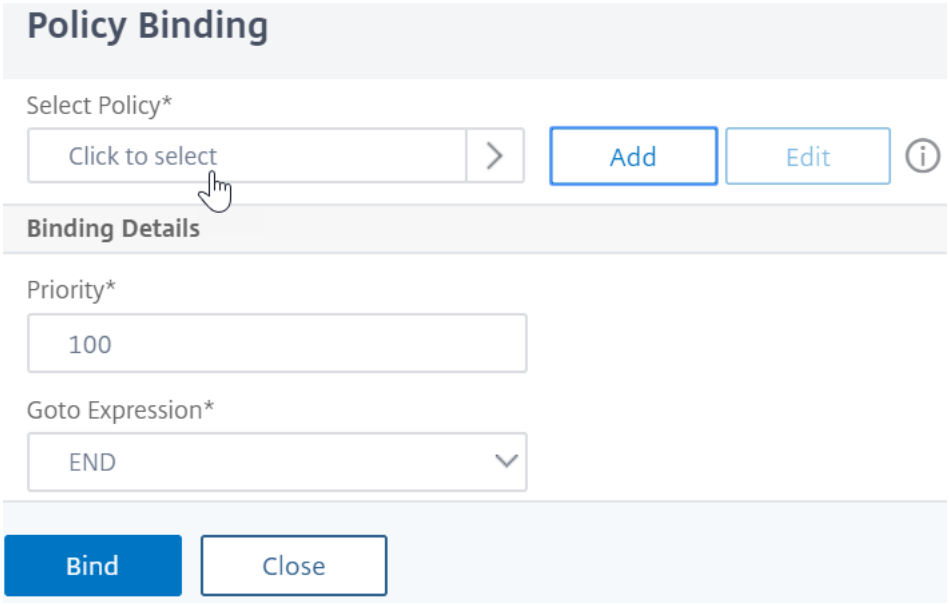
6. In the Advanced Settings column, click **Login Schemas**.



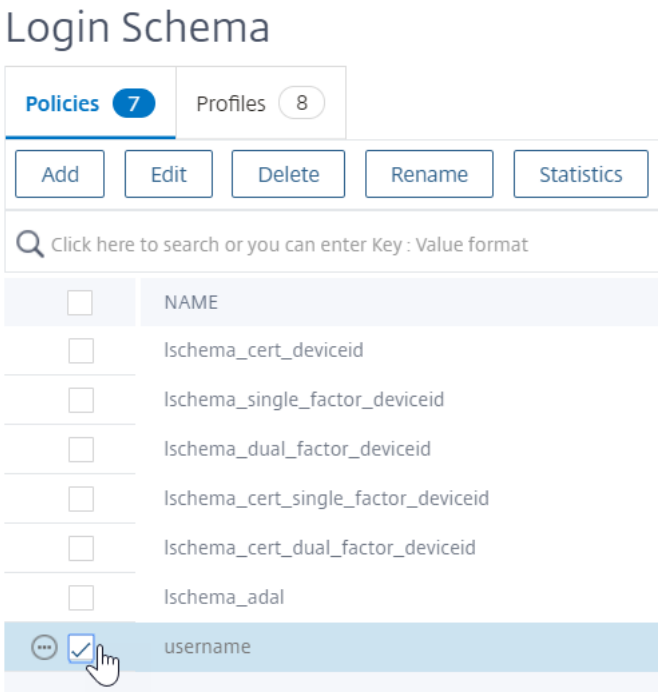
7. In the Login Schemas section, click the text **No Login Schema**.



8. Click the text, **Click to select**.



9. Click the radio button next to the login schema policy, and click **Select**. Only login schema policies appear in this list. Login schema profiles (without a policy) do not appear.



10. Click **Bind**.

Advanced authentication polices

Authentication policies are a combination of policy expression, and policy action. If the expression is true, then evaluate the authentication action.

Create advanced authentication policies

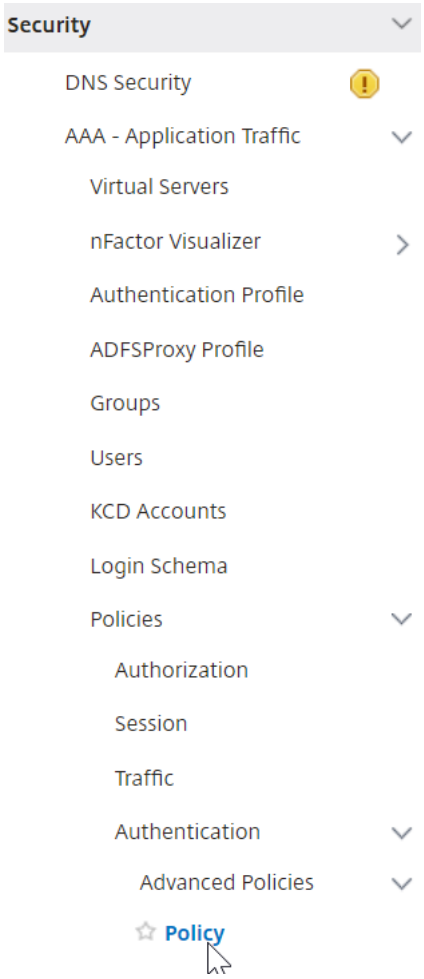
Authentication policies are a combination of policy expression and policy action. If the expression is true, then evaluate the authentication action.

You will need authentication actions/servers (e.g. LDAP, RADIUS, CERT, SAML, etc.)
When creating an advanced authentication policy, there's a plus (Add) icon that lets you create authentication actions/servers.

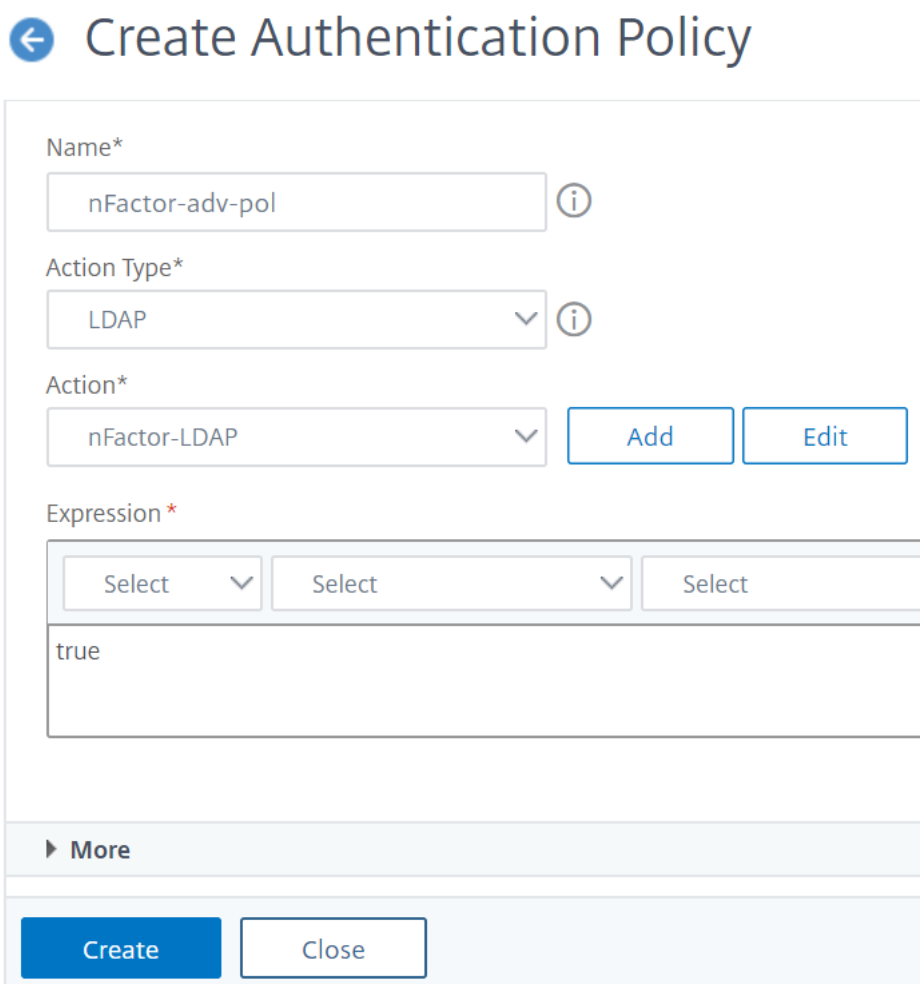
Or you can create authentication actions (aervers) prior to creating the advanced authentication policy. The authentication servers are located under **Authentication > Dashboard**. On the right, click Add and select a Server Type. The instructions for creating these Authentication Servers is not detailed here. See the Authentication – NetScaler 12 / Citrix ADC 12.1 procedures.

To create an Advanced Authentication Policy:

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**



2. In the details pane do one of the following:
   - To create a policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Edit**.
3. In the **Create Authentication Policy** or **Configure Authentication Policy** dialog box, type or select values for the parameters.



- **Name** - The policy name. Cannot be changed for a previously configured policy.
- **Action Type** - The policy type: Cert, Negotiate, LDAP, RADIUS, SAML, SAMLIDP, TACACS, or WEBAUTH.
- **Action** - The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the plus and create an action of the proper type.
- **Log Action** - The audit action to associate with the policy. You can choose an existing audit action, or click the plus and create an action.
  You don't have any Actions configured, or to create an action, click **Add** and complete the steps.
- **Expression** - The rule that selects connections to which you want to apply the action that you specified. The rule can be simple ("true" selects all traffic) or complex. You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open Add Expression dialog box and using the drop-down lists in it to construct your expression.)
- **Comment** - You can type a comment that describes the type of traffic that this authentica-

tion policy applies to. Optional.

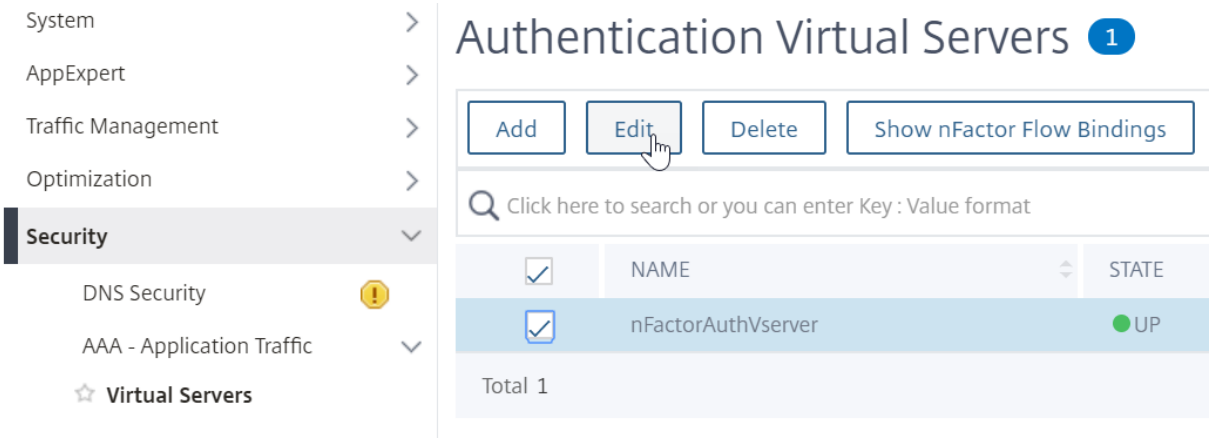
4. Click **Create** and then click **Close**. If you created a policy, that policy appears in the Authentication Policies and Servers page.

You must create additional advanced authentication policies as required based on your nFactor design.

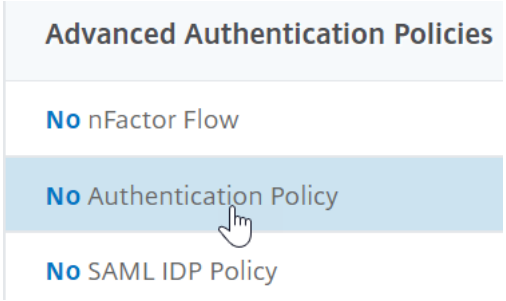
Bind first factor advanced authentication policy to Citrix ADC AAA

You can directly bind advanced authentication policies for the first Factor the Citrix ADC AAA virtual server. For the next factors, you must bind the advanced authentication policies to the authentication policy labels.

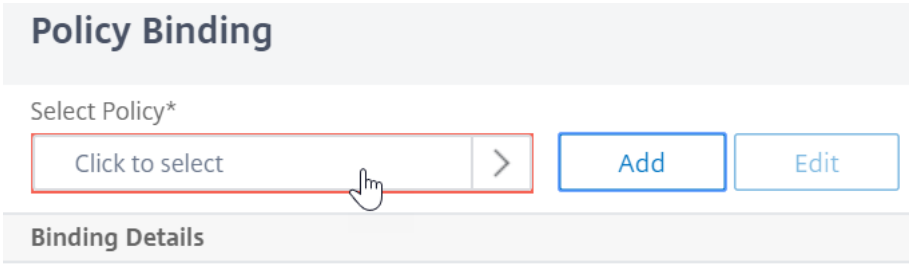
1. Navigate to **Security > AAA - Application Traffic > Virtual Servers**. Edit an existing virtual server.



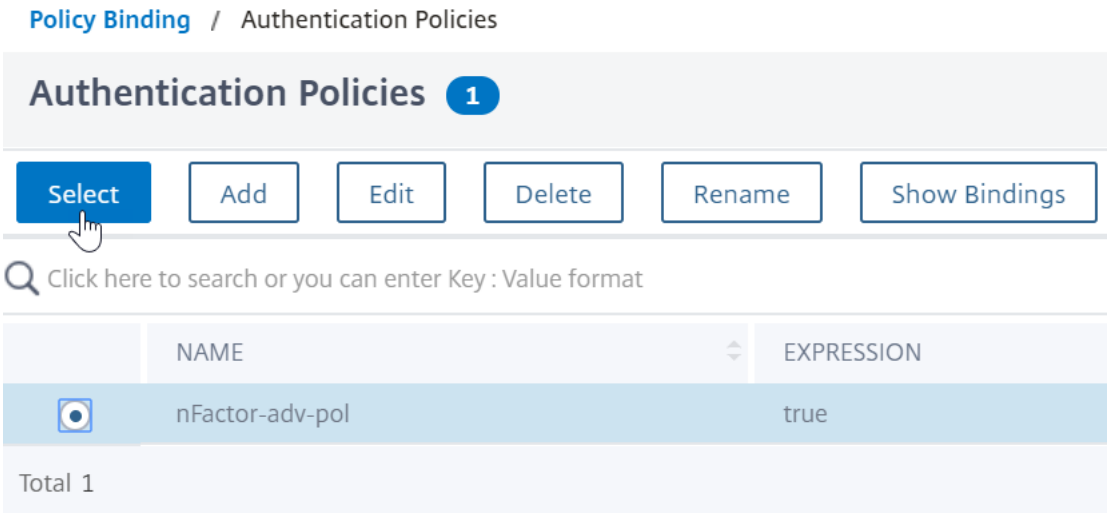
1. On the left, in the Advanced Authentication Policies section, click **No Authentication Policy**.



2. In **Select Policy**, click the text, **Click to select**.

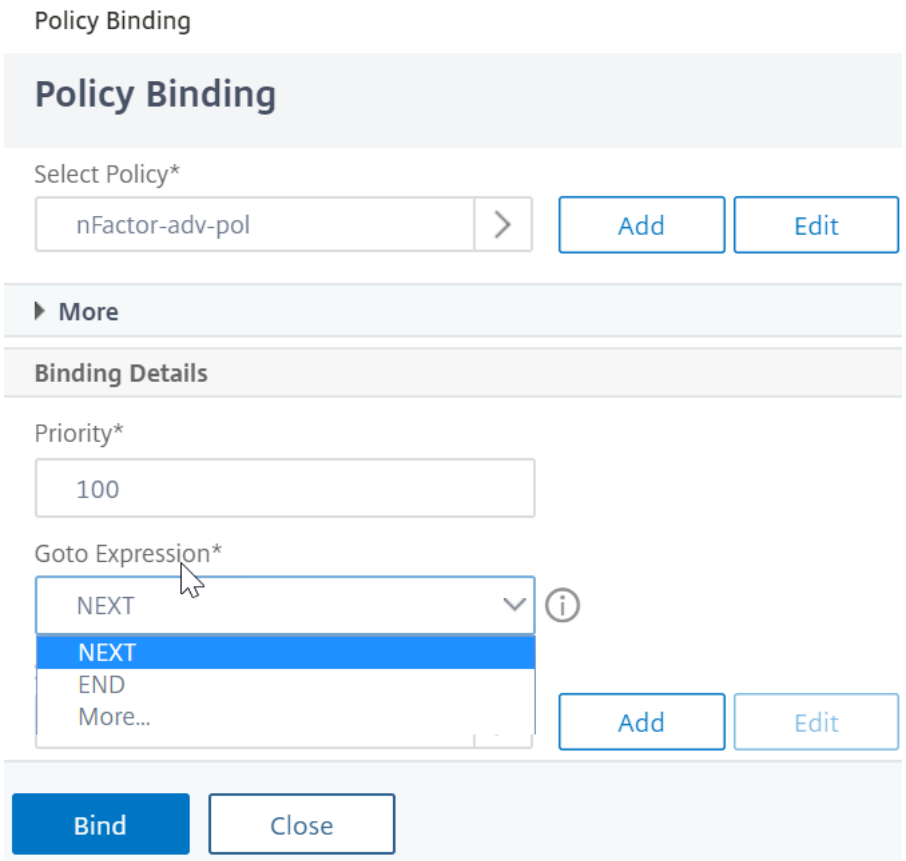


3. Click the radio button next to the **Advanced Authentication Policy**, and click **Select**.

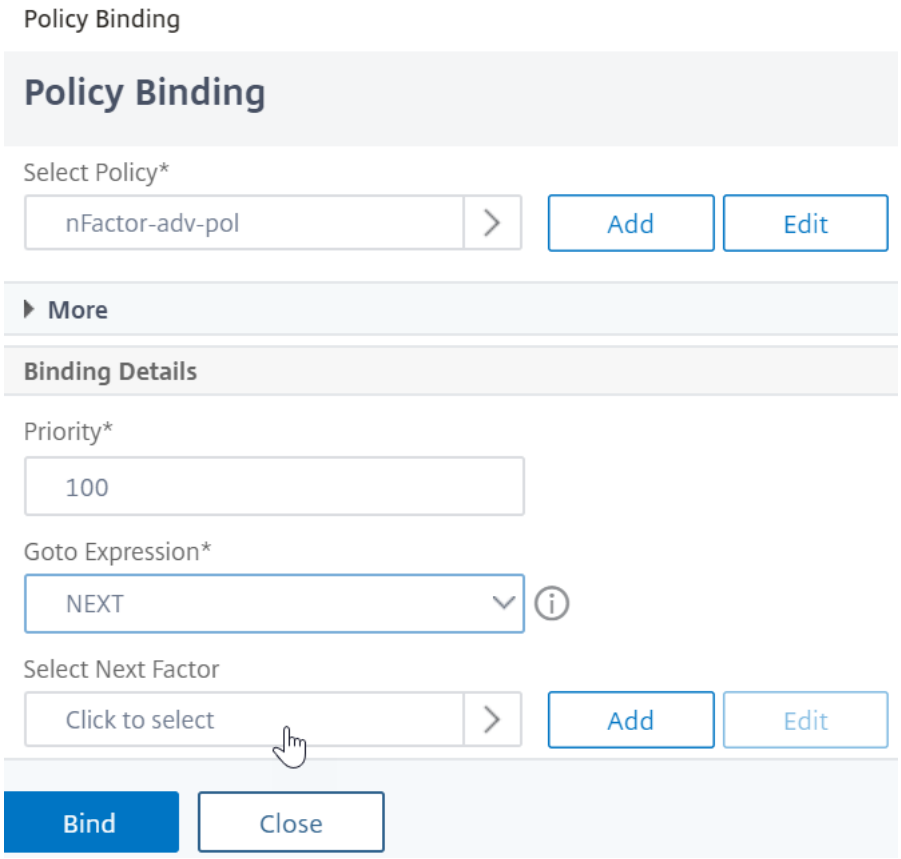


4. In the Binding Details section, the **Goto Expression** determines what happens next if this advanced authentication policy fails.

   - If **Goto Expression** is set to **NEXT**, then the next advanced authentication policy bound to this Citrix ADC AAA Virtual Server is evaluated.
   - If **Goto Expression** is set to **END**, or if there are no more advanced authentication policies bound to this Citrix ADC AAA Virtual Server, then authentication is completed and marked as failed.



5. In **Select Next Factor**, you can select can point to an authentication policy label. The next factor is evaluated only if the advanced authentication policy succeeds. Finally, click **Bind**.



Use extracted LDAP groups to select the next authentication Factor

You can use extracted LDAP groups to select the next authentication factor without actually authentication with LDAP.

1. When creating or editing an LDAP server or LDAP action, clear the **Authentication** check box.
2. In **Other Settings**, select appropriate values in **Group Attribute** and **Sub Attribute Name**.

Authenticate the policy label

When you bind an advanced authentication policy to the Citrix ADC AAA Virtual Server and have selected a next factor, the next factor is evaluated only if the advanced authentication policy. The next factor that is evaluated is an authentication policy label.

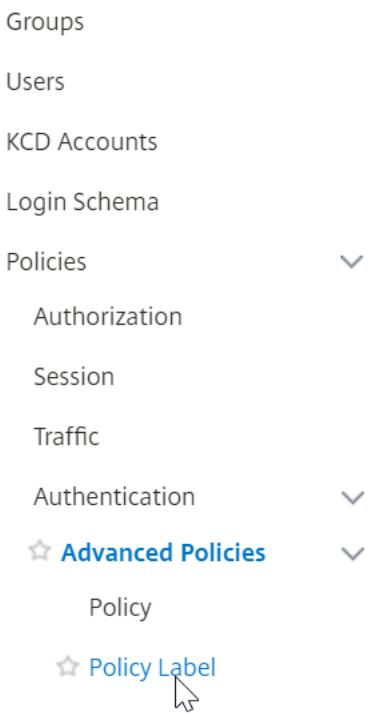
The authentication policy label specifies a collection of authentication policies for a particular factor. Each policy label corresponds to a single factor. It also specifies the login form that must be presented to the user. The authentication policy label must be bound as the next factor of an authentication policy or of another authentication policy label.

Note: Every factor does not need a login schema. Login schema profile is required only if you are binding a login schema to an Authentication Policy Label.

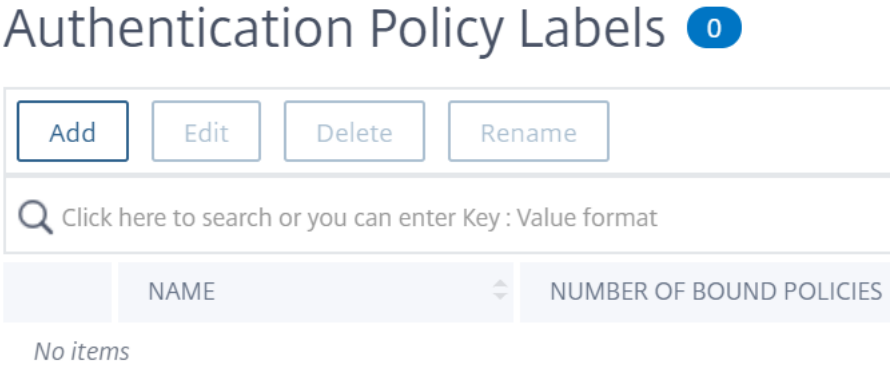
Create authentication policy label

A policy label specifies the authentication policies for a particular factor. Each policy label corresponds to a single factor. The policy label specifies the login form that must be presented to the user. The policy label must be bound as the next factor of an authentication policy or of another authentication policy label. Typically, a policy label includes authentication policies for a specific authentication mechanism. However, you can also have a policy label that has authentication policies for different authentication mechanisms.

1. Navigate to **Security > AAA – Application Traffic > Policies > Authentication > Advanced Policies > Policy Label**.



2. Click the **Add** button.

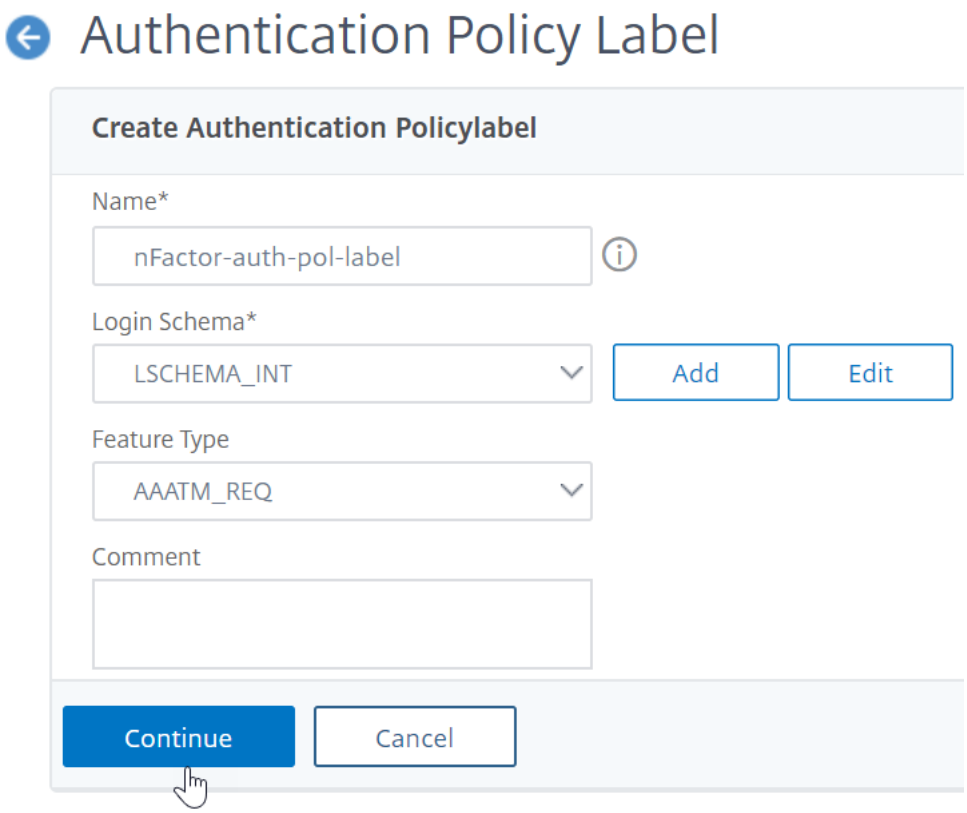


3. Complete the following fields to Create Authentication Policy Label:

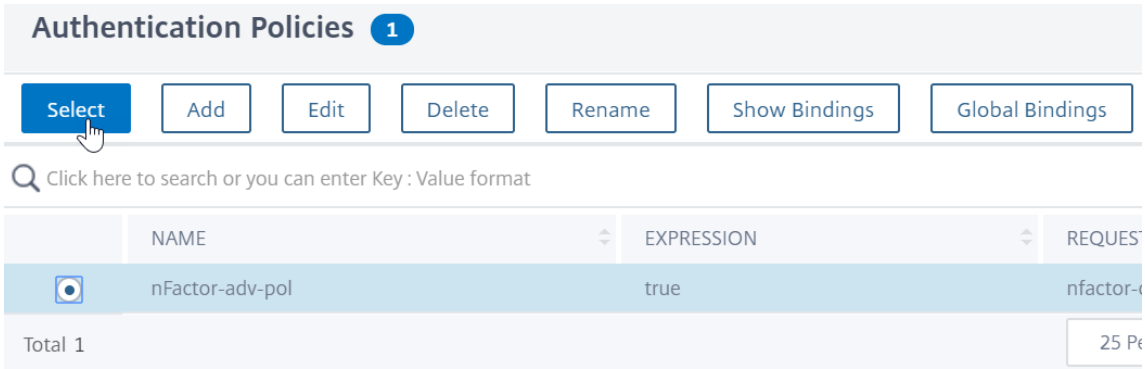
   a) Enter the **Name** for the new authentication policy label.

b) Select the **Login Schema** associated with authentication policy label. IF you do not want to display anything to the user, you can select a login schema profile that is set to noschema (LSCHEMA_INT).

c) Click **Continue**.



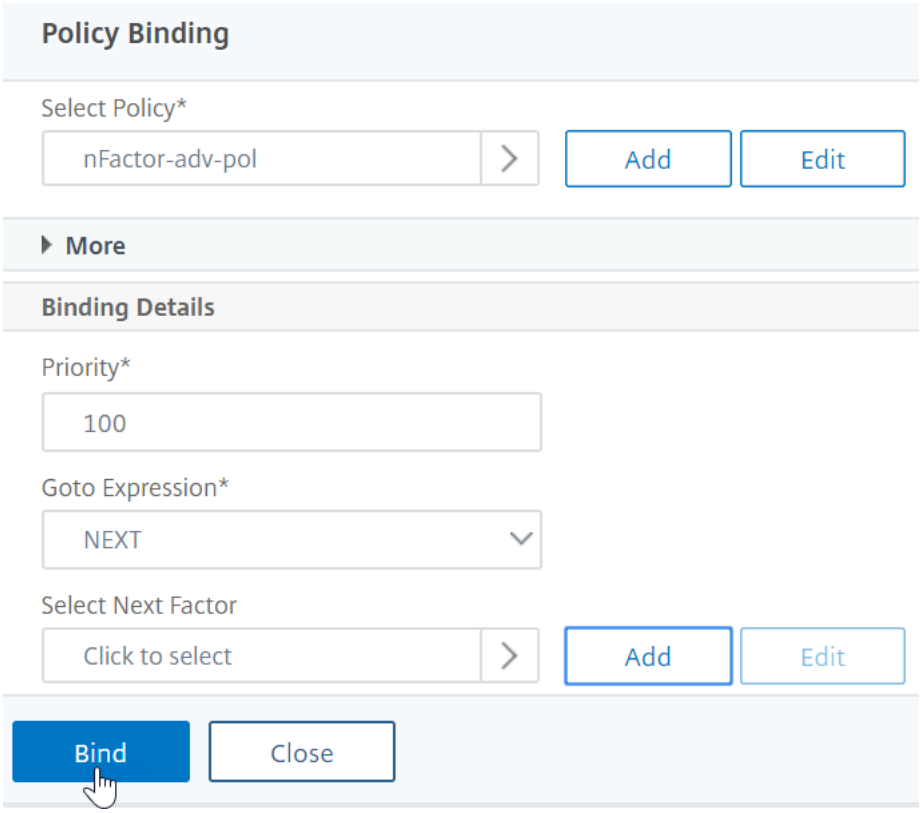
4. In **Policy Binding** section, click where it says **Click to select**.
5. Select the authentication policy that evaluates this factor.



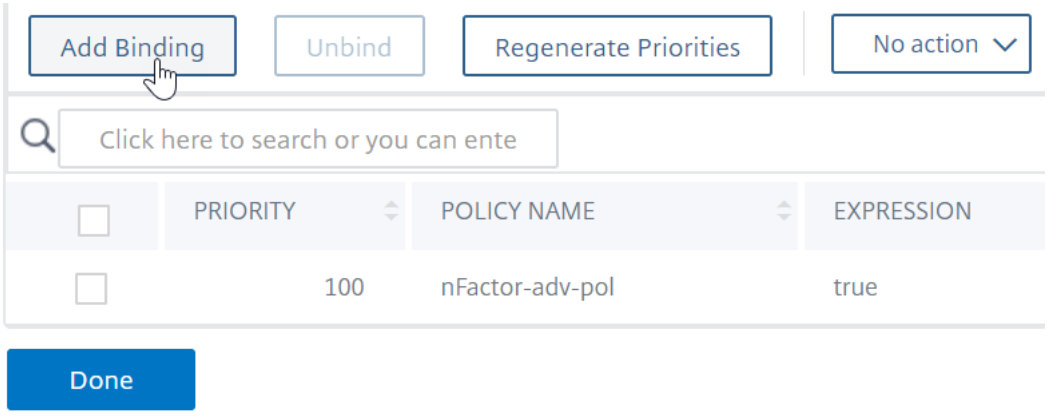
6. Complete the following fields:

   a) Enter the **Priority** of the policy binding.

   b) In **Goto Expression** select **NEXT** if you want to bind more advanced authentication policies to this factor or select **END**.



7. In **Select Next Factor**, if you want to add another factor, click to select and bind the next authentication policy label (next factor).
   If you do not select the next factor, and if this advanced authentication policy succeeds, then authentication is successful and complete.
8. Click **Bind**.
9. You can click **Add Binding** to add more advanced authentication policies to this policy label (factor). Click **Done** upon completion.



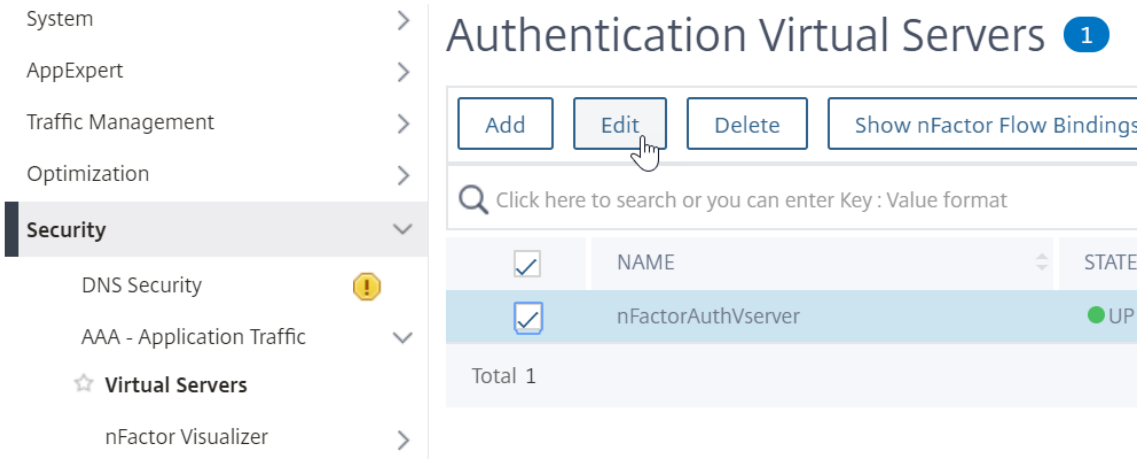
Bind authentication policy label

After you create the policy label, you bind it to an existing advanced authentication policy binding to chain factors together.

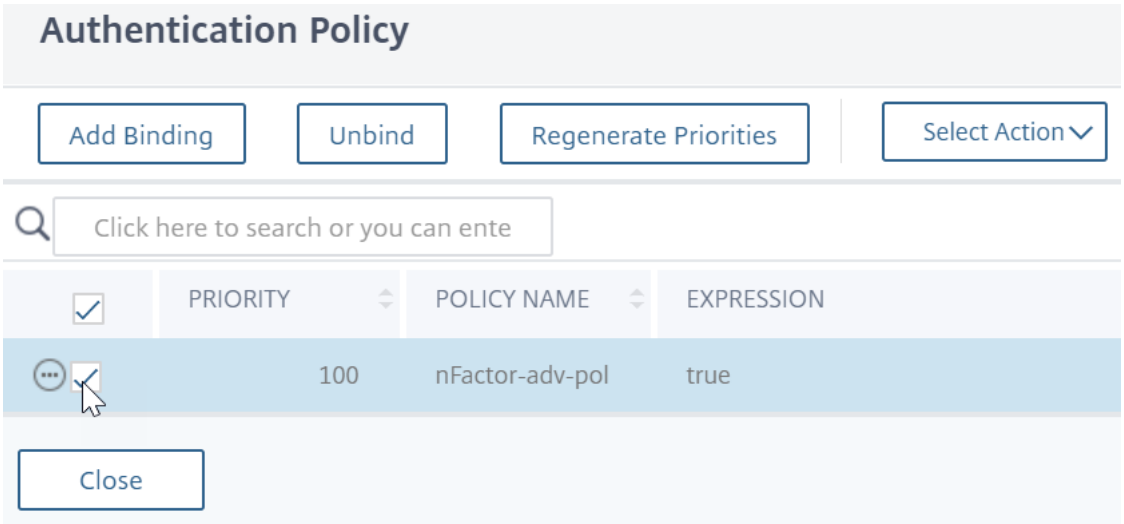
You can select the next factor when editing an existing Citrix ADC AAA virtual server that has an advanced authentication policy bound or when editing a different policy label to include next factor.

To edit an existing Citrix ADC AAA virtual server that has an advanced authentication policy already bound to it

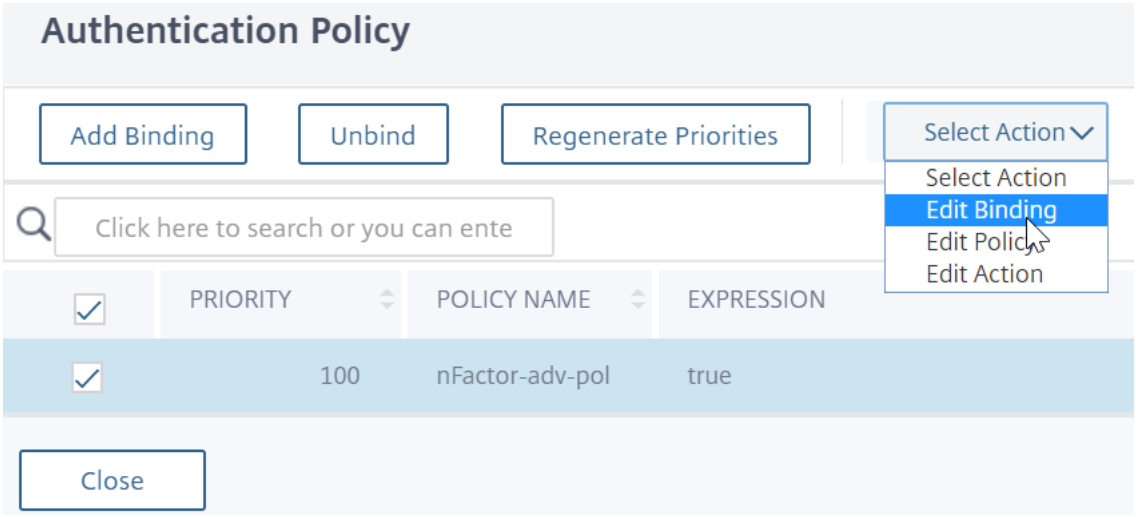
1. Navigate to **Security > AAA – Application Traffic > Virtual Servers.** Select the virtual server and click **Edit**.



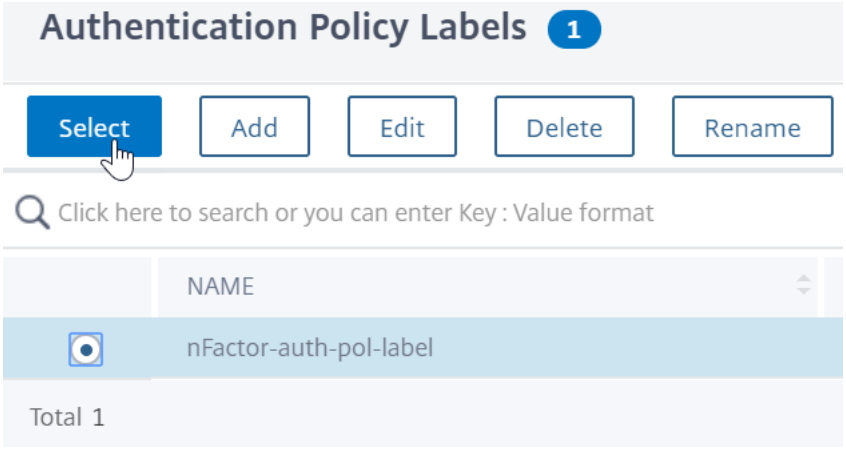
2. On the left, in the **Advanced Authentication Policies** section, click an existing authentication policy binding.



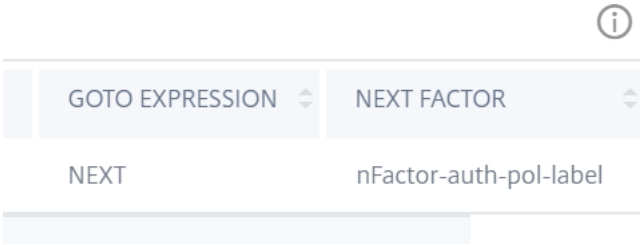
3. In **Select Action**, click **Edit Binding**.



4. In **Select Next Factor**, click, and select an existing authentication policy label (next factor).

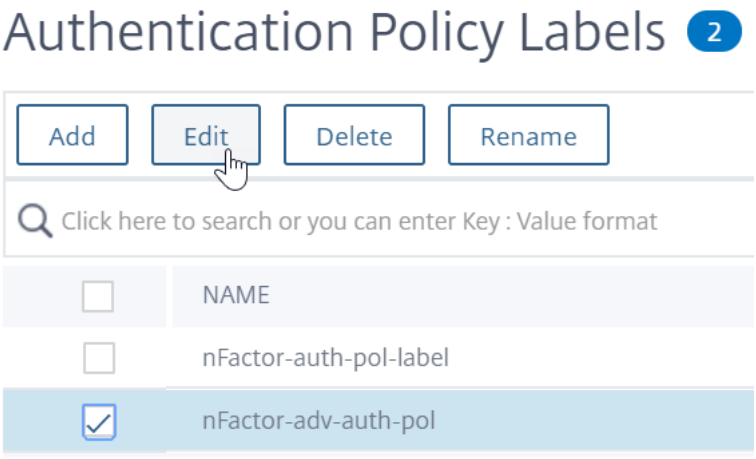


5. Click **Bind**. You can see the next factor on the extreme right.

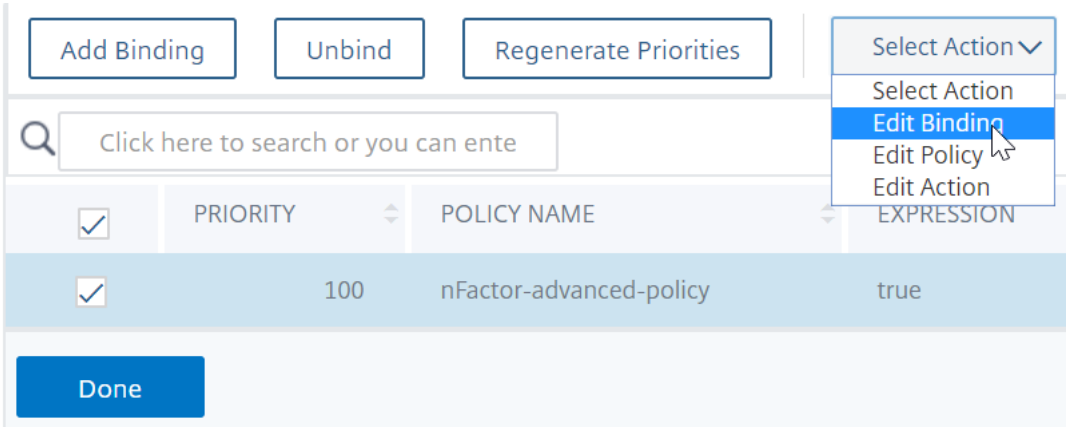


To add a policy label next factor to a different policy label

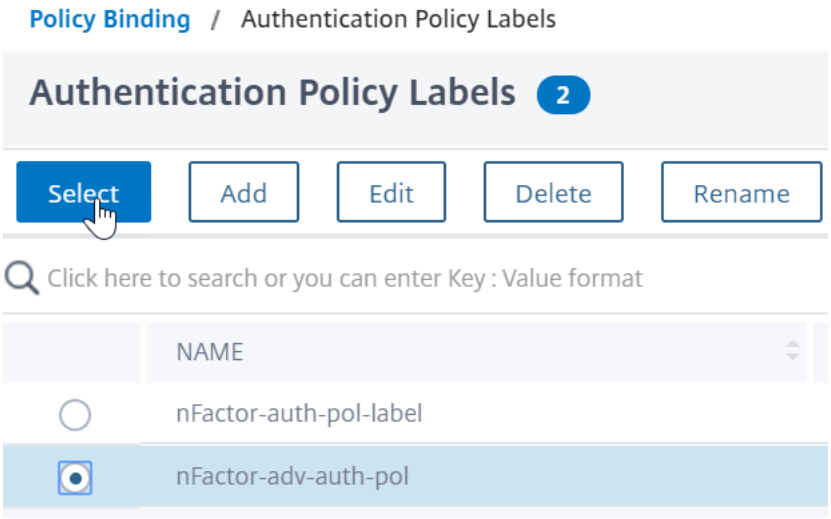
1. Navigate to **Security > AAA – Application Traffic > Policies > Authentication > Advanced Policies > PolicyLabel**. Select a different policy label and click **Edit.**



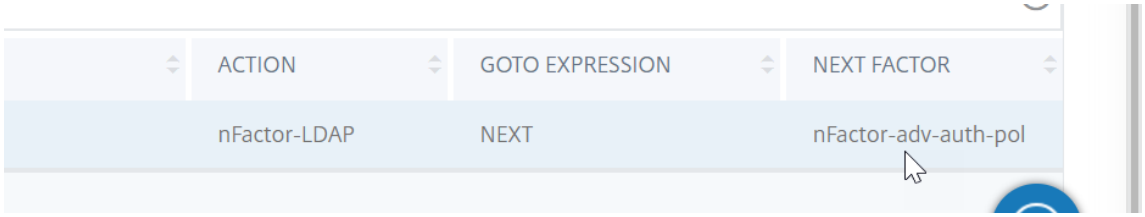
2. In **Select Action**, click **Edit Binding**.



3. In **Binding Details > Select Next Factor**, click to select the next factor.
4. Choose the policy label for the next factor and click the **Select** button.



5. Click **Bind.** You can see the next factor on the right.

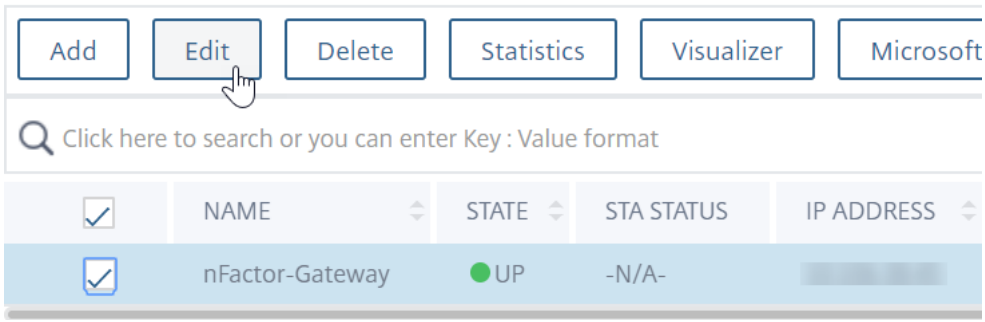


nFactor for Citrix Gateway

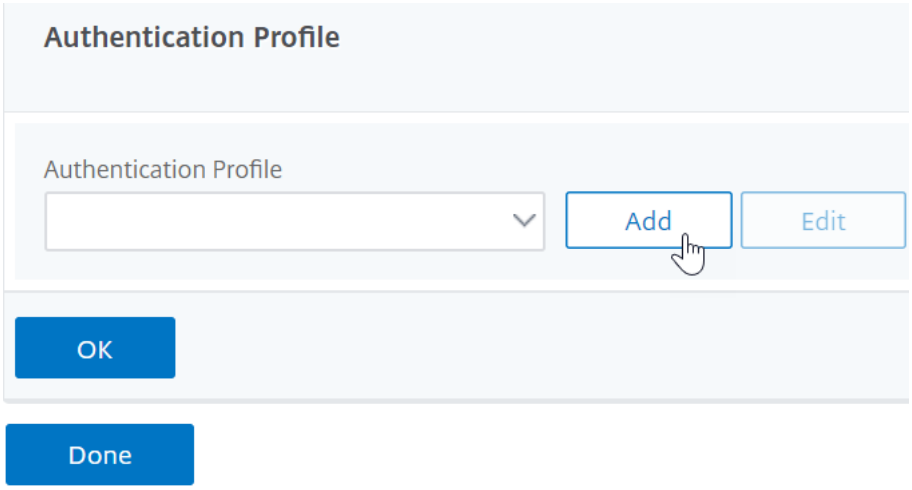
To enable nFactor on Citrix Gateway, an authentication profile must be linked to a Citrix ADC AAA virtual server.

Create authentication profile to link a Citrix ADC AAA virtual server with Citrix Gateway virtual server

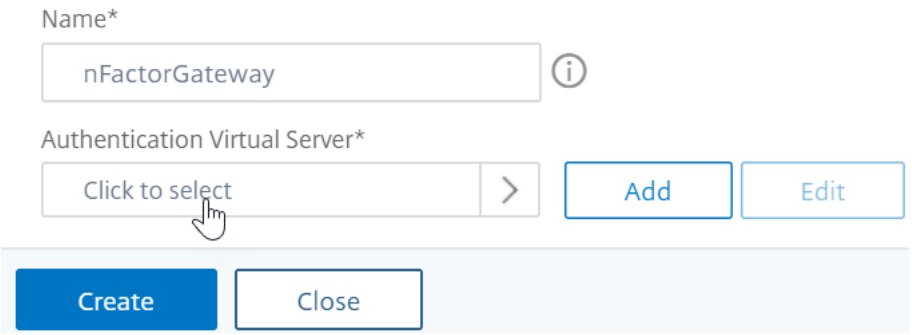
1. Navigate to **Citrix Gateway > Virtual Servers** and select and existing gateway virtual server to edit.



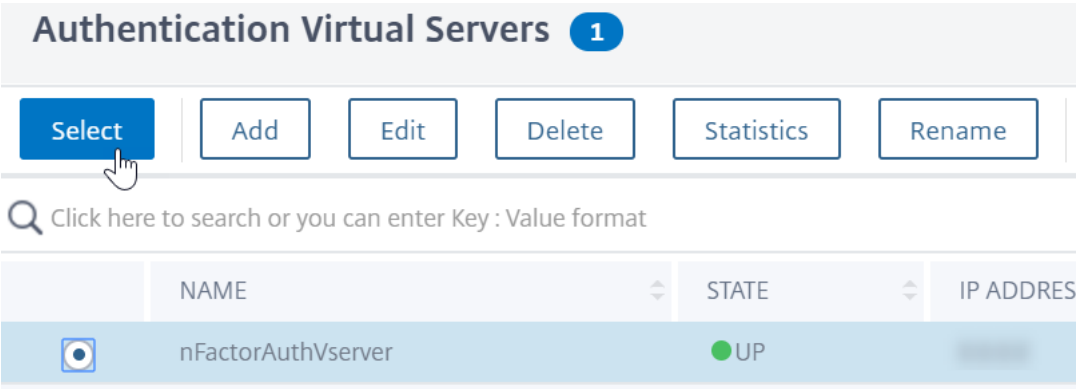
2. In **Advanced Settings**, click **Authentication Profile**.
3. Click **Add** under **Authentication Profile**



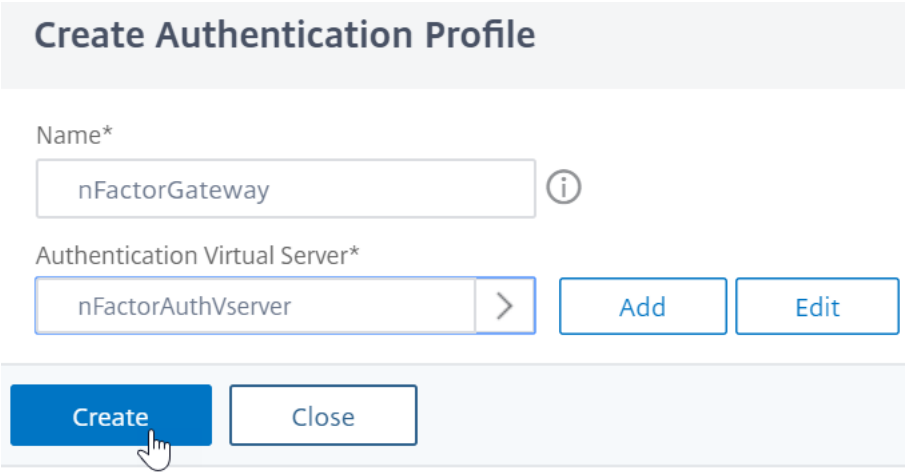
4. Enter the name for the authentication profile and click where it says **Click to select**.



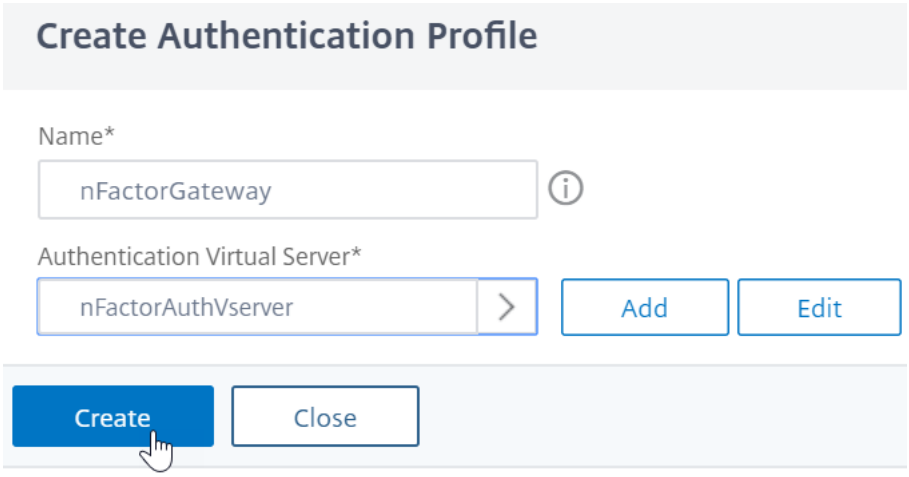
5. In **Authentication Virtual Server**, select an existing server that has login schema, advanced authentication policy, and authentication policy labels configured. You can also create an authentication virtual server. The Citrix ADC AAA virtual server does not need an IP address. Click **Select**.



6. Click **Create**.



7. Click **OK** to close the Authentication Profile section.



Note: If you have configured one of the factors as client certificates, then must configure SSL parameters and CA certificate.

After you have completed linking the authentication profile to a AAA virtual server, and when you browse to your Citrix Gateway, you can view the nFactor authentication screens.

Configure SSL parameters and CA certificate

If one of the authentication factors is a certificate, then you must perform some SSL configuration on the Citrix Gateway virtual server.

1. Navigate to **Traffic Management > SSL > Certificates > CA Certificates**, and install the root certificate for the issuer of the client certificates. Certificate Authority certificates do not need key files.

   If default SSL Profiles are enabled, then you should have already created an SSL Profile that has Client Authentication enabled.

2. Navigate to **Citrix Gateway > Virtual Servers**, and edit an existing Citrix Gateway virtual server that is enabled for nFactor.
   - If default SSL Profiles are enabled, click the edit icon.
   - In the SSL Profile list, select the SSL Profile that has Client Authentication enabled and set to OPTIONAL.
   - If default SSL Profiles are not enabled, click the edit icon.
   - Check the Client Authentication check box.
   - Ensure Client Certificate is set to Optional
3. Click OK.
4. In Certificates section, click **No CA Certificate.**

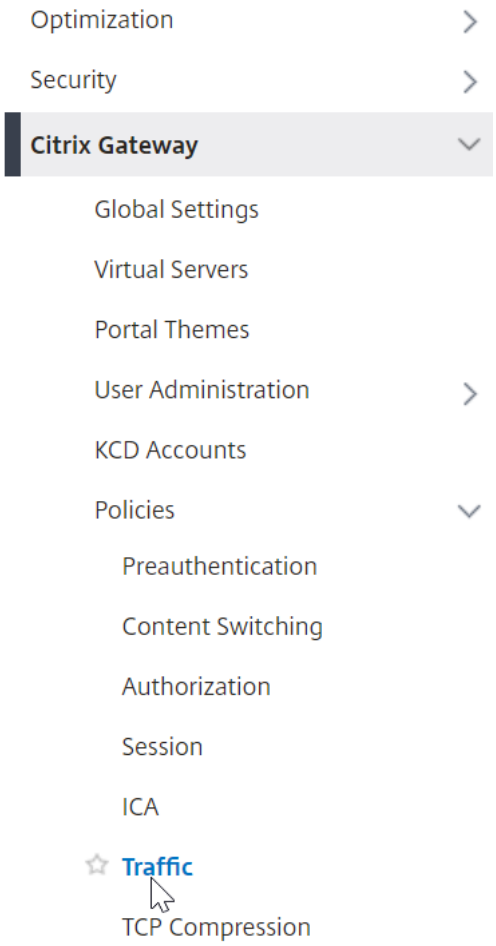
5. In Select CA Certificate, click to select and select the root certificate for the issuer of the client certificates.
6. Click Bind.

Note: You might have to also bind any Intermediate CA Certificates that issued the client certificates.

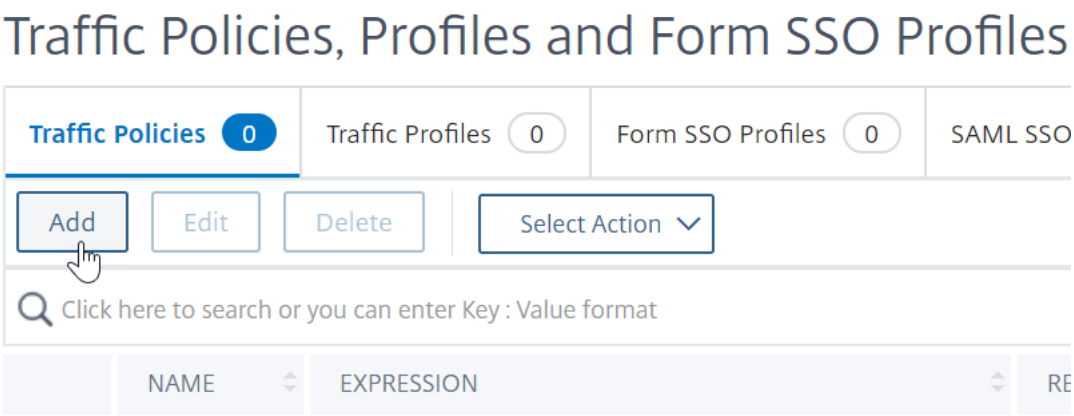
Configure Citrix Gateway traffic policy for nFactor single sign-on to StoreFront

For single sign-on to StoreFront, nFactor defaults to using the last entered password. If LDAP is not the last entered password, then you must create a traffic policy/profile to override the default nFactor behavior.

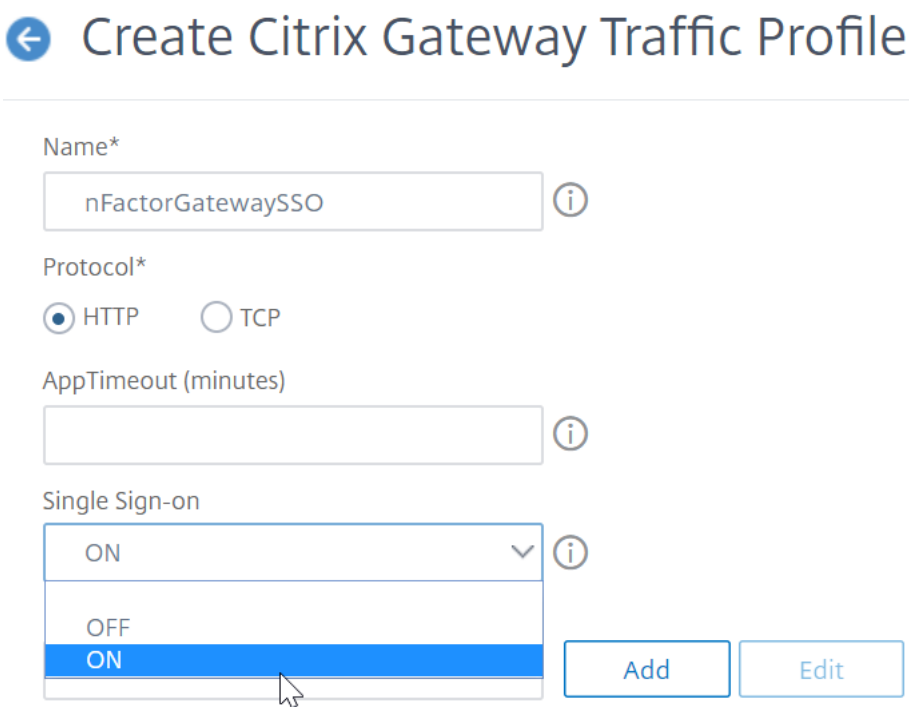
1. Navigate to **Citrix Gateway > Policies > Traffic**.



2. In **Traffic Profiles** tab, click **Add**.

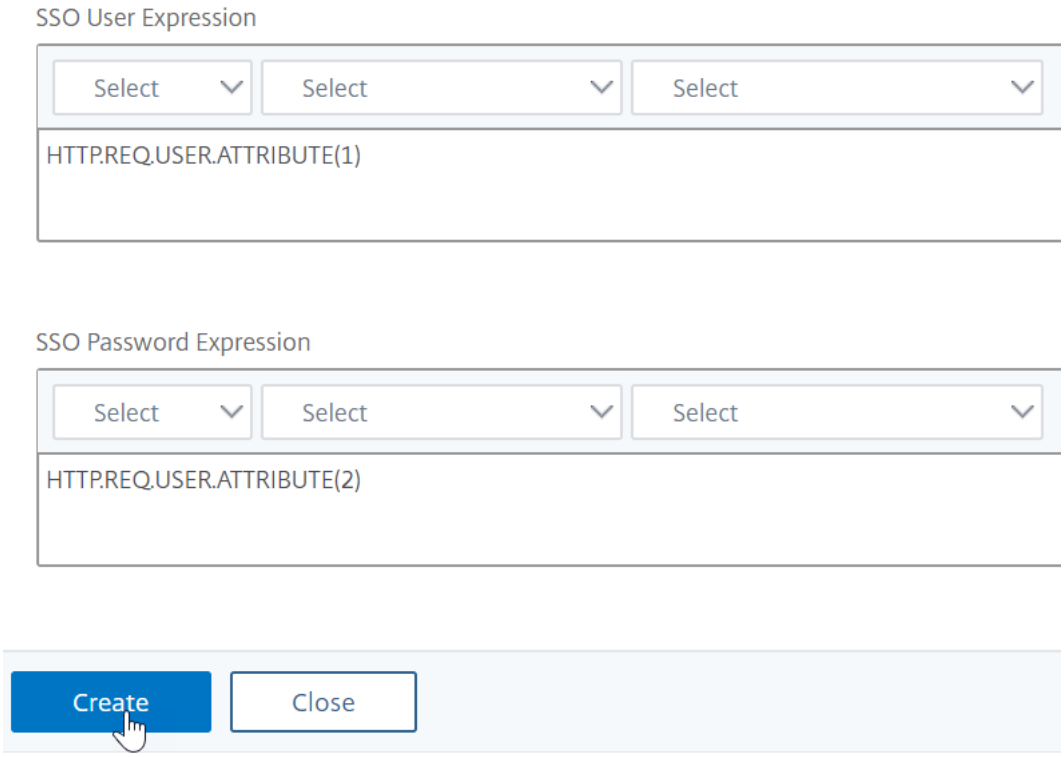


3. Enter a name for the traffic profile. Select the **HTTP** protocol.
   In **Single Sign-on**, select **ON**.



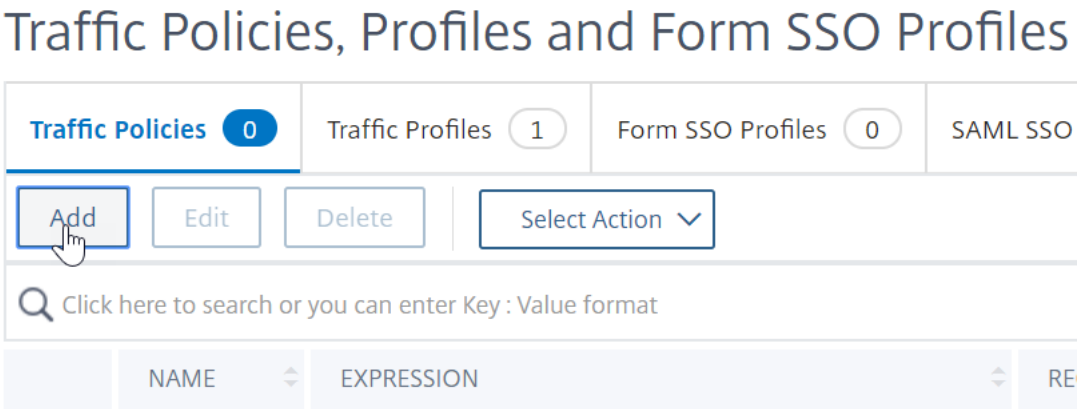
4. In the **SSO Expression**, enter a AAA.USER.ATTRIBUTE(#) expression that matches the indexes specified in the login schema and click **Create**.

> **Note**
> AAA.USER expression is now implemented to replace the deprecated HTTP.REQ.USER expressions.



5. Click **Traffic Policies** tab, and click **Add**.

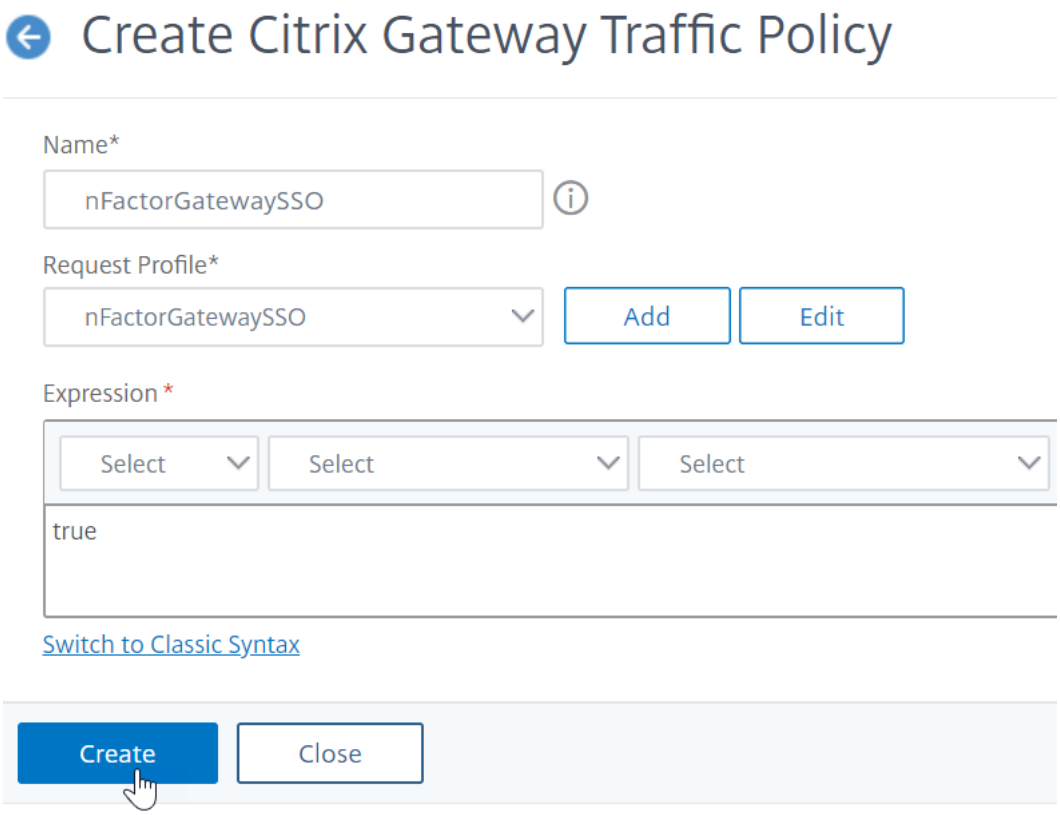
   Enter a name for the policy.
   Select the traffic profile created in the previous step.
   In **Expression**, enter an advanced expression, for example true.
   Click **Create**.



6. Navigate to **Citrix Gateway > Citrix Gateway Virtual Server**.

   - Select and existing virtual server and click **Edit**.
   - In the **Policies** section, click the **+** sign.
   - In **Choose Policy**, select **Traffic**.
   - In **Choose Type**, select **Request**.

- Select the traffic policy that you have created and then click **Bind**.



Sample snippet on nFactor configuration by using the Citrix ADC CLI

To understand the step-wise configurations for nFactor authentication, let us consider a two-factor authentication deployment where the first factor is LDAP authentication and the second factor is RADIUS authentication.

This sample deployment requires the user to log in to both factors using a single login form. Therefore, we define a single login form that accepts two passwords. The first password is used for LDAP authentication and the other for RADIUS authentication.
Here are the configurations that are performed:

1. Configure the load balancing virtual server for authentication

   add lb vserver lbvs89 HTTP 1.136.19.55 80 -AuthenticationHost auth56.aaatm.com -Authentication ON

2. Configure the authentication virtual server.

   add authentication vserver auth56 SSL 10.106.30.223 443 -AuthenticationDomain aaatm.com

3. Configure the login schema for the login form and bind it to a login schema policy.

add authentication loginSchema login1 -authenticationSchema login-2passwd.xml -userCredentialIndex 1 -passwordCredentialIndex 2

> **Note**
>
> You might need to use the username and one of the passwords entered in the login schema for Single Sign-on (SSO) to a backend service, for example StoreFront. You can reference these index values in the traffic action by using the expression AAA.USER.ATTRIBUTE(#). The values can be between 1 and 16.
>
> Alternatively, you can use the credentials entered in the login schema as your Single Sign-On credentials by using the following command.
>
> **add authentication loginSchema** login1 -authenticationSchema login-2passwd.xml -SSOCredentials YES

add authentication loginSchemaPolicy login1 -rule true -action login1

4. Configure a login schema for the pass-through and bind it to a policy label

   add authentication loginSchema login2 -authenticationSchema noschema

   add authentication policylabel label1 -loginSchema login2

5. Configure the LDAP and RADIUS policies.

   add authentication ldapAction ldapAct1 -serverIP 10.17.103.28 -ldapBase “dc=aaatm, dc=com” -ldapBindDn administrator@aaatm.com -ldapBindDnPassword 81qw1b99ui971mn1289op1abc125 -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName samAccountName -groupAttrName memberOf -subAttributeName CN

   add authentication Policy ldap -rule true -action ldapAct1

   add authentication radiusAction radius -serverIP 10.101.14.3 -radKey n231d9a8cao8671or4a9ace940d8623 -encrypted -encryptmethod ENCMTHD_3 -radNASip ENABLED -radNASid NS28.50 -radAttributeType 11 -ipAttributeType 8

   add authentication Policy radius -rule true -action radius

6. Bind the login schema policy to the authentication virtual server

   bind authentication vserver auth56 -policy login1 -priority 1 -gotoPriorityExpression END

7. Bind the LDAP policy (first factor) to the authentication virtual server.

   bind authentication vserver auth56 -policy ldap -priority 1 -nextFactor label1 -gotoPriorityExpression next

8. Bind the RADIUS policy (second factor) to the authentication policy label.

   bind authentication policylabel label1 -policyName radius -priority 2 -gotoPriorityExpression end

nFactor Visualizer for simplified configuration

September 14, 2021

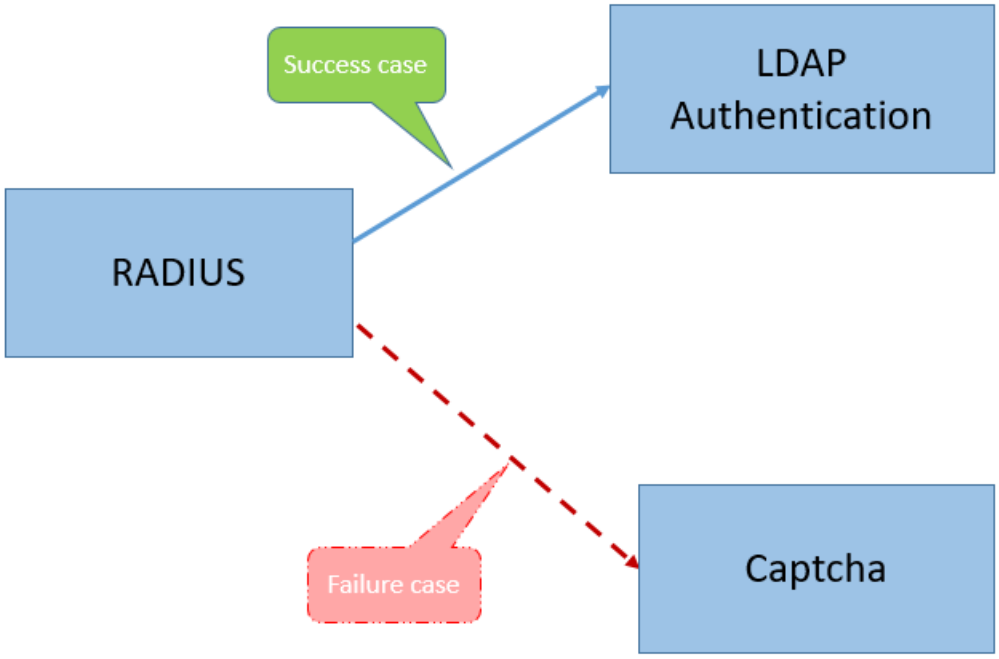
Starting from Citrix ADC release 13.0 build 36.27, nFactor configuration through the GUI is simplified by using the nFactor Visualizer. The nFactor Visualizer helps admins add multiple factors without losing track of each factor. The group of factors that are built in the flow are displayed in one place. Admins can add authentication success and failure paths separately. After creating the flow, admins have to bind the nFactor flow to an authentication virtual server.

> **Note**
>
> - All factors created by an admin in the nFactor flow are retained for any future use.
> - From Citrix ADC feature release 13.0 build 64.35 and above, using the nFactor visualizer, you can start the nFactor flow with a decision block.

Previously, nFactor configuration was cumbersome wherein the admins had to visit many pages to configure it. If a change was required, the admins had to revisit the configured sections each time. Also, there was no option to view the complete configuration in one place.

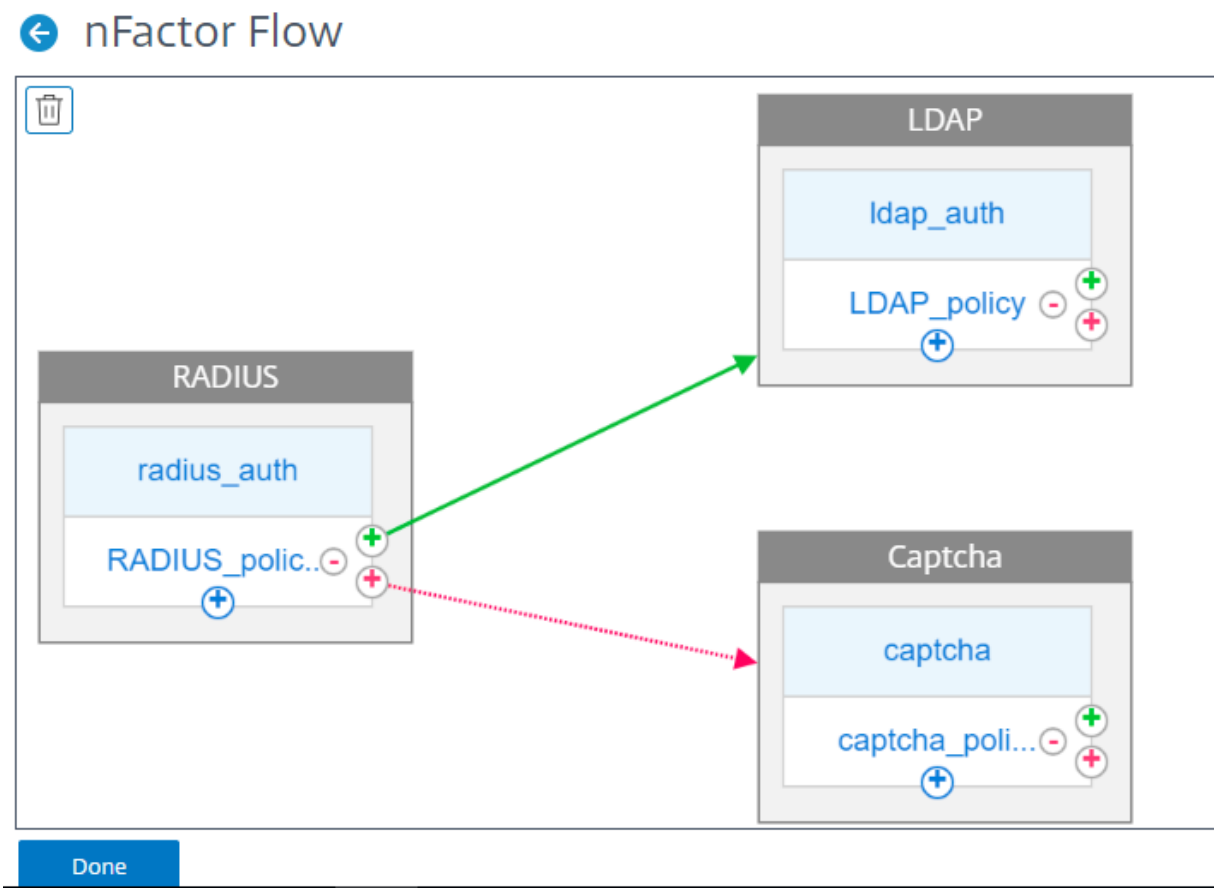
Use Case 1: RADIUS followed by LDAP authentication, else fallback to Captcha through nFactor Visualizer

Achieve RADIUS authentication as the first-level authentication followed by LDAP authentication. In case RADIUS fails, authentication must fall back to Captcha.



To achieve this use case, you can use the nFactor Visualizer. The Visualizer provides various controls that can be used to add this flow and the related items.

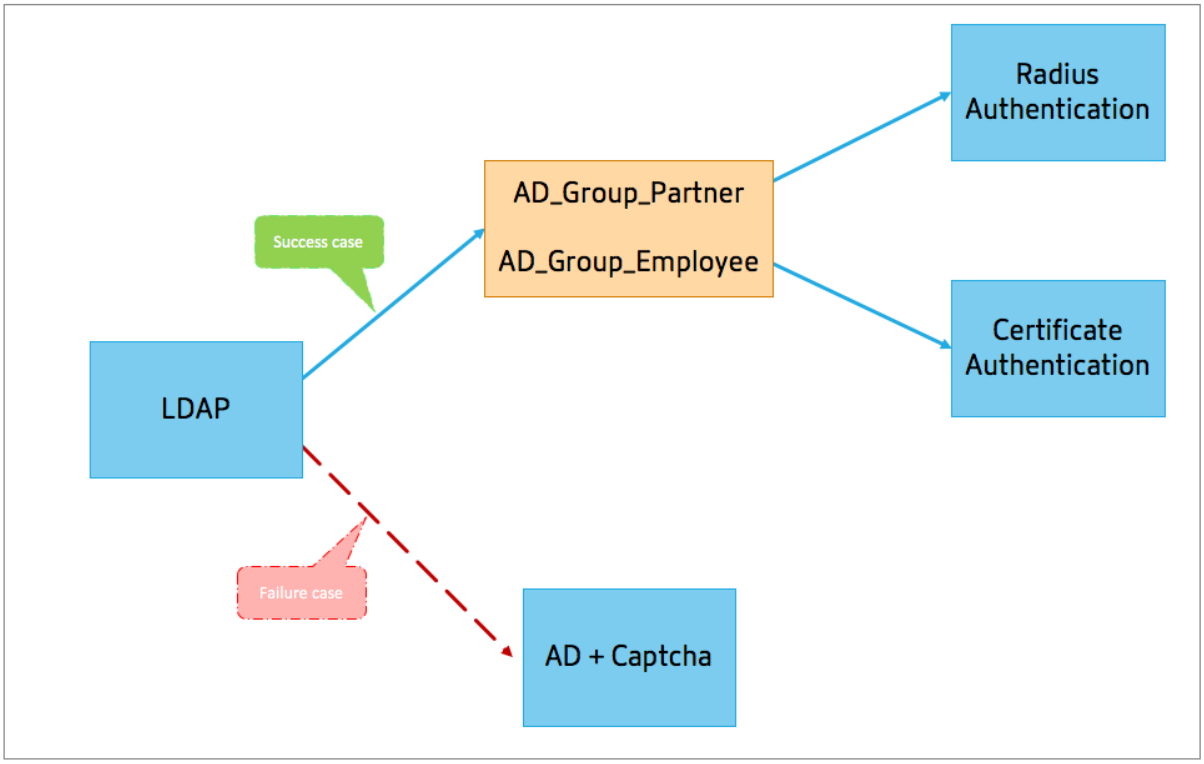
The following figure displays the nFactor flow created for the previous mentioned use case by using the visualizer.



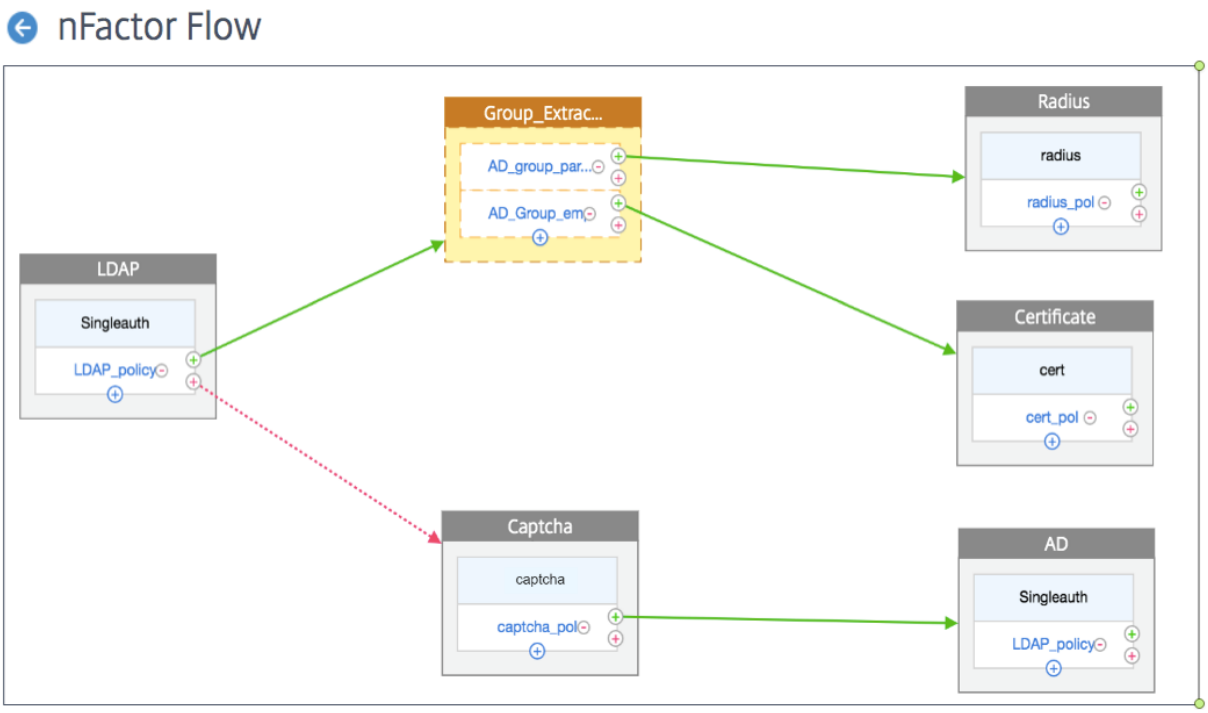
- **RADIUS.** You configure RADIUS as the first factor. You add a login schema and a policy. In this example, radius_auth and RADIUS_policy are the login schema and policy that is added. For the RADIUS_Policy, you can add another factor for the success case. In this example, an LDAP factor block is added for the success case. For the failure case, you can add a Captcha factor.
- **LDAP.** You configure LDAP authentication as the second factor. You add a login schema and a policy. In this example, ldap_auth and LDAP_policy are the login schema and policy that is added.
- **Captcha.** For the RADIUS policy failure case, you create a Captcha factor. In this example, captcha and captcha_policy are the login schema and policy that is added.

Use Case 2: LDAP followed by RADIUS/Certificate authentication with Captcha based on LDAP Group Membership through nFactor Visualizer

Achieve RADIUS authentication as the first-level authentication followed by LDAP authentication. In case RADIUS fails, authentication must fall back to Captcha.



The following figure displays the nFactor flow created for the previous mentioned use case by using the visualizer.



- **LDAP.** You configure LDAP as the first factor. You add a login schema and a policy. In this example SingleAuth and LDAP_Policy are the login schema and policy that is added. For the

LDAP_Policy, you can add another factor for the success case. In this example, a decision block is added for the success case. For the failure case, you can add Captcha followed by AD factor.

- **Group Extraction LDAP.** Is the decision block added for the LDAP success case. The decision block is used as a branch out factor to branch out the users based on the policy rules. Visualizer allows configuring only a NO_AUTHN policy for the decision block.

  In this example, Group_Extraction_LDAP is the decision block. You add two policies (`AD_Group_Partner and AD_Group_Employee`) to this decision block. As explained in the use cases, all requests routed through AD_Group_Partner policy use RADIUS authentication. Therefore, you connect the success case of this policy to the next factor that is RADIUS factor. Similarly, all requests routed through AD_Group_Employee policy use certification authentication. Therefore, you connect the success case of this policy to the next factor that is the certification authentication factor.

  - **RADIUS.** For the AD_Group_Partner policy success case, you create the RADIUS authentication factor.
  - **Certificate.** For the AD_Group_Employee policy success case, you create the certificate authentication factor.

- **Captcha.** For the LDAP policy failure case, you create two next factors, Captcha and AD factor.

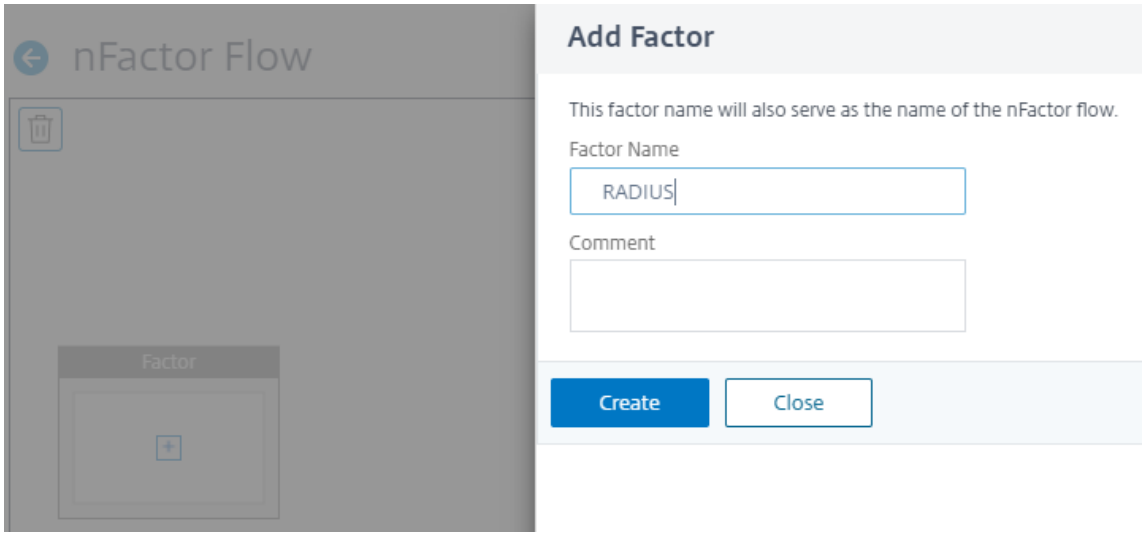
> **Note**
>
> - If you have a use case to branch out as a first thing, then you can either create two flows and bind separately or create one flow with the first one as branch out, and bind it to the virtual server.
> - If you have multiple blocks, and to view the entire flow in the nFactor Flow screen, click the visualizer and drag the flow to the extreme left.
> - Citrix recommends modifying the nFactor flows using the nFactor Flows page only.

To configure nFactor by using the nFactor Visualizer

> **Note**
>
> The following nFactor configuration is a simple example that helps you accomplish the Use Case 1 scenario configurations.

1. Navigate to **Security > AAA – Application Traffic > nFactor Visualizer > nFactor Flows**.
2. Click **Add**.
3. On the **nFactor Flows** page, click **+** to add a first factor for the flow. The first factor also serves as an identifier for this nFactor flow.

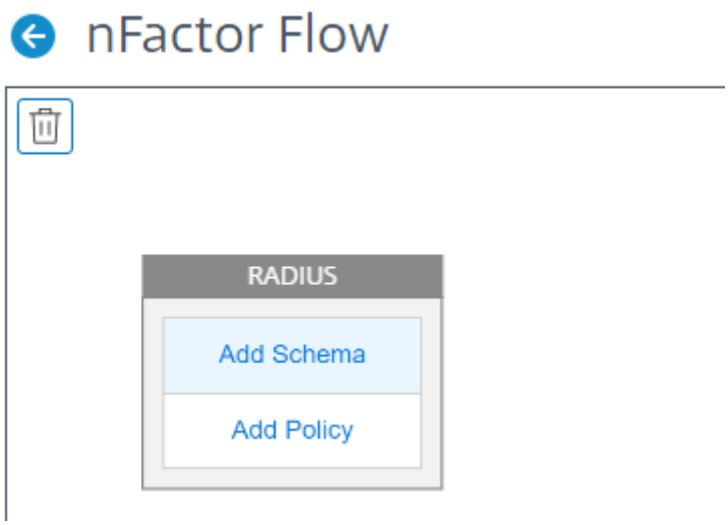
4. Enter the factor name and click **Create**.



The factor name appears on the factor block in the nFactor Flow page.

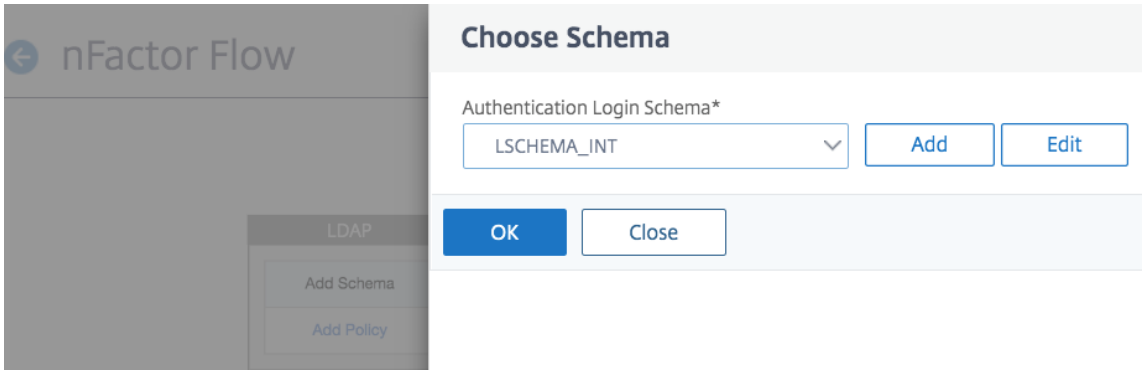
> **Note**
>
> Citrix recommends that you must not use policy label names such as, `__root` and `__<flow_name>` as suffix and `_db_` as prefix. It is used as the factor names that are created in the nFactor flow.

5. Once the RADIUS factor is created, the Add Schema and Add Policy must be created.

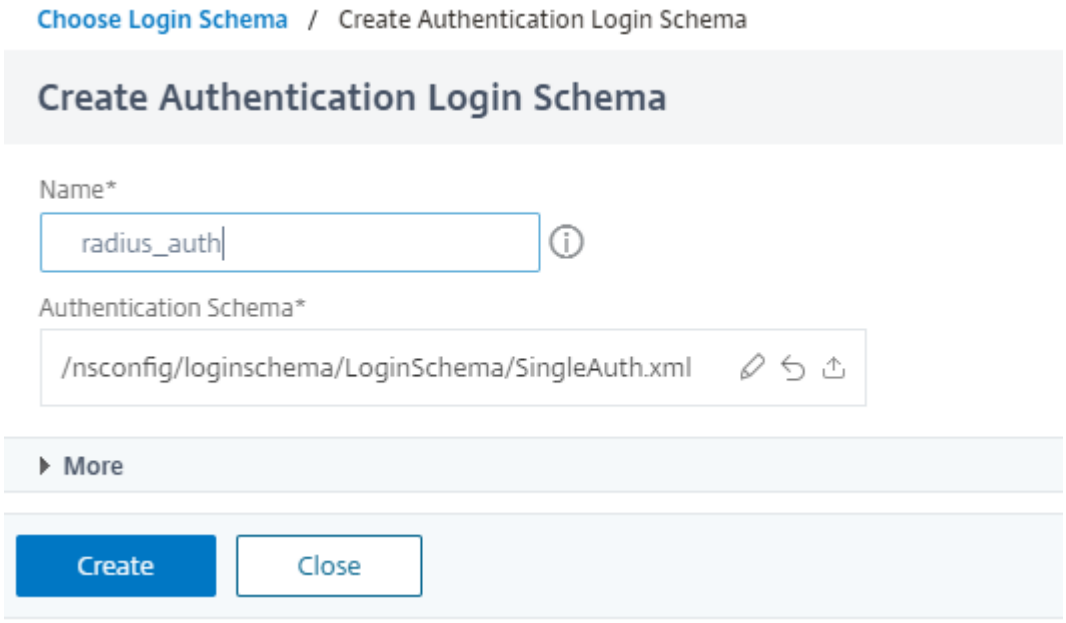


> **Note**
>
> For more information, see nFactor concepts, entities, and terminology

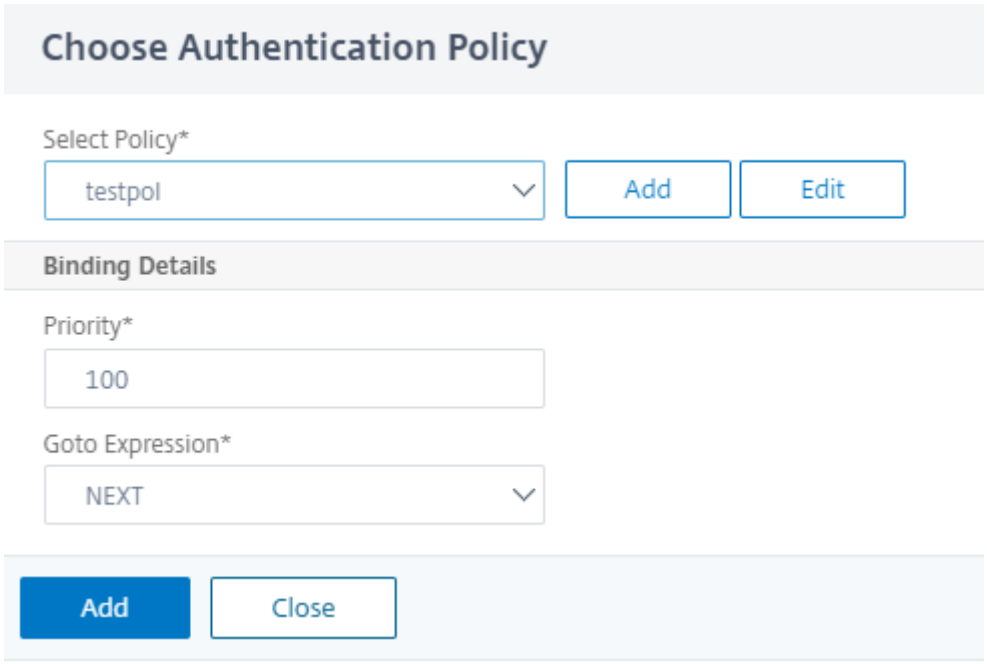
6. Click **Add Schema**. You can either add a new login schema or select an existing login schema from the **Authentication Login Schema** list.



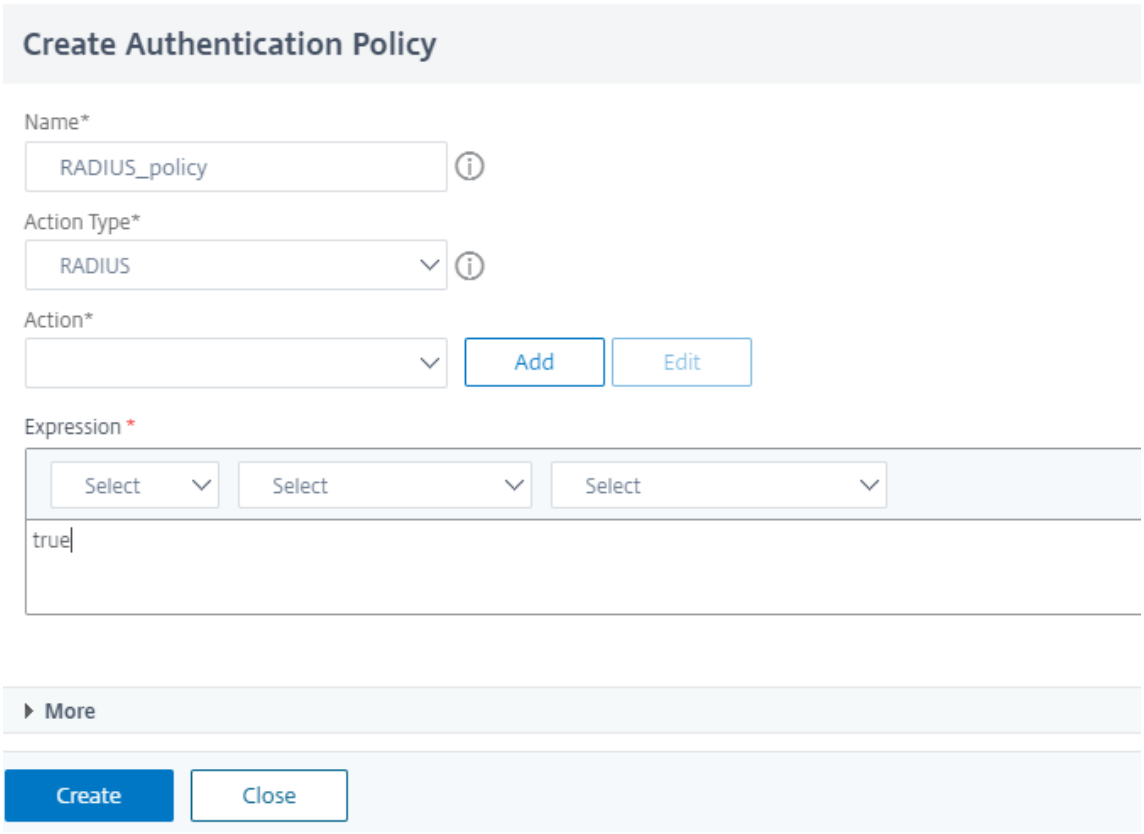
7. To create a login schema, click **Add** and in the **Create Authentication Login Schema** page, enter the name for the schema. Click **Edit** (pencil icon) to select the **Login Schema Files** from the list.



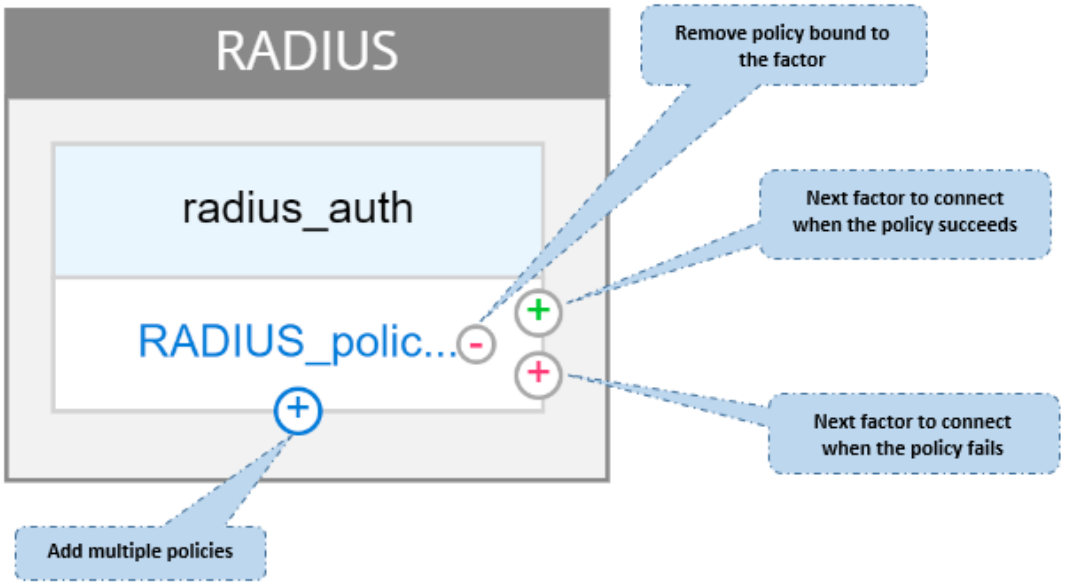
8. Click **Add Policy**. You can create an authentication policy or select an existing authentication policy.



9. To create a new policy, click **Add** and in the **Create Authentication Policy** page, enter the name for the policy and click **Create**.



10. After you add a login schema and policy to the factor, the login schema and policy appear on the factor in the Visualizer as displayed in the following figure. For any given factor, you can add multiple policies and define the next factor for the success and failure of each policy. You can also remove the policies that are part of the factor.



11. After you create the flow, you can then bind the nFactor flow to an authentication virtual server.

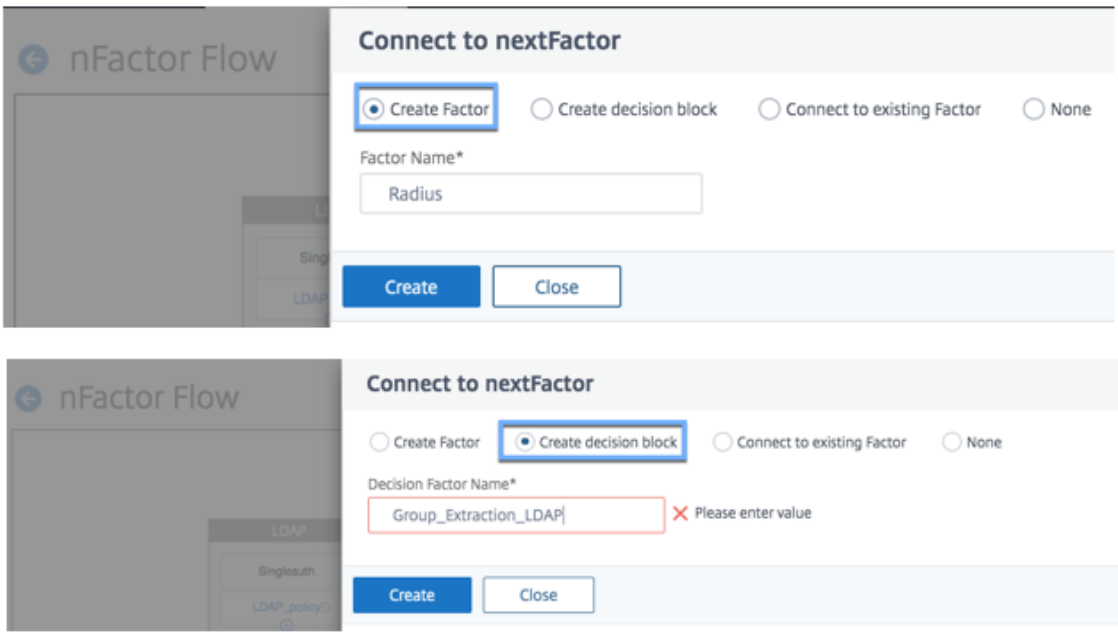
Adding the next factor

To add the next factor, you can select one of the following options as per your requirement:

- **Create Factor**. Create a factor. Each factor that is created in a flow is exclusive to that flow.
- **Create a decision block.** Create a decision block to serve as a branch-out factor. You cannot add a login schema to the decision block. Visualizer allows configuring only a NO_AUTHN policy for the decision block.

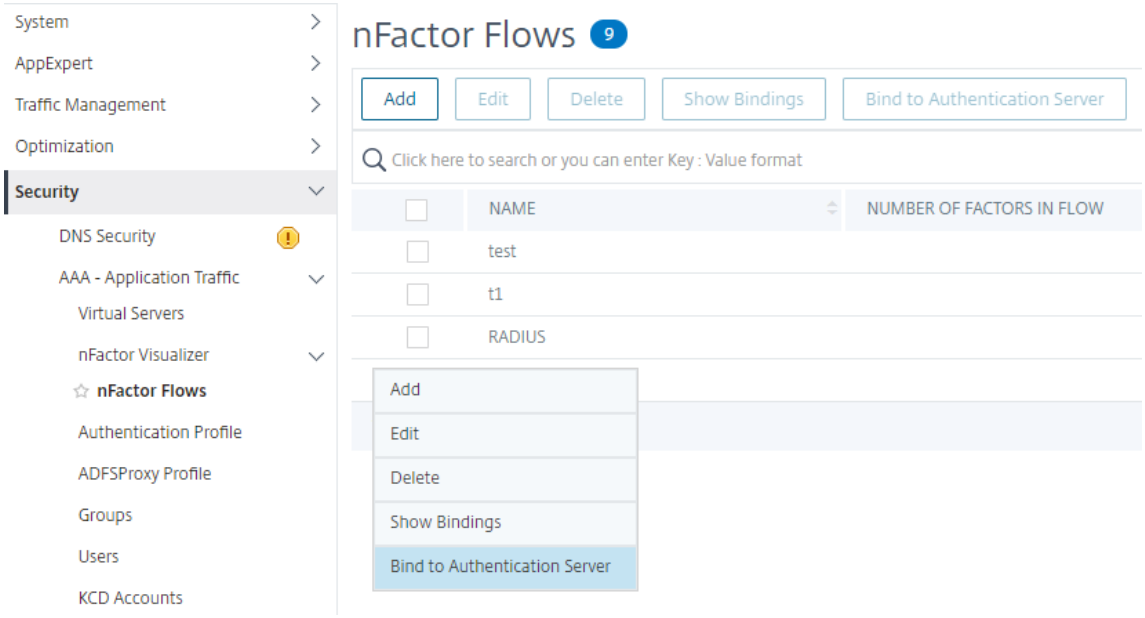
  > **Note**
  >
  > You can only add or edit the decision block through the Citrix ADC GUI. There is no option to configure the decision block from the CLI command.

- **Connect to an existing Factor**. Select an existing factor as your next factor. All the factors that appear in the existing list are created exclusively for that flow.
- **None.** Remove an existing connection.

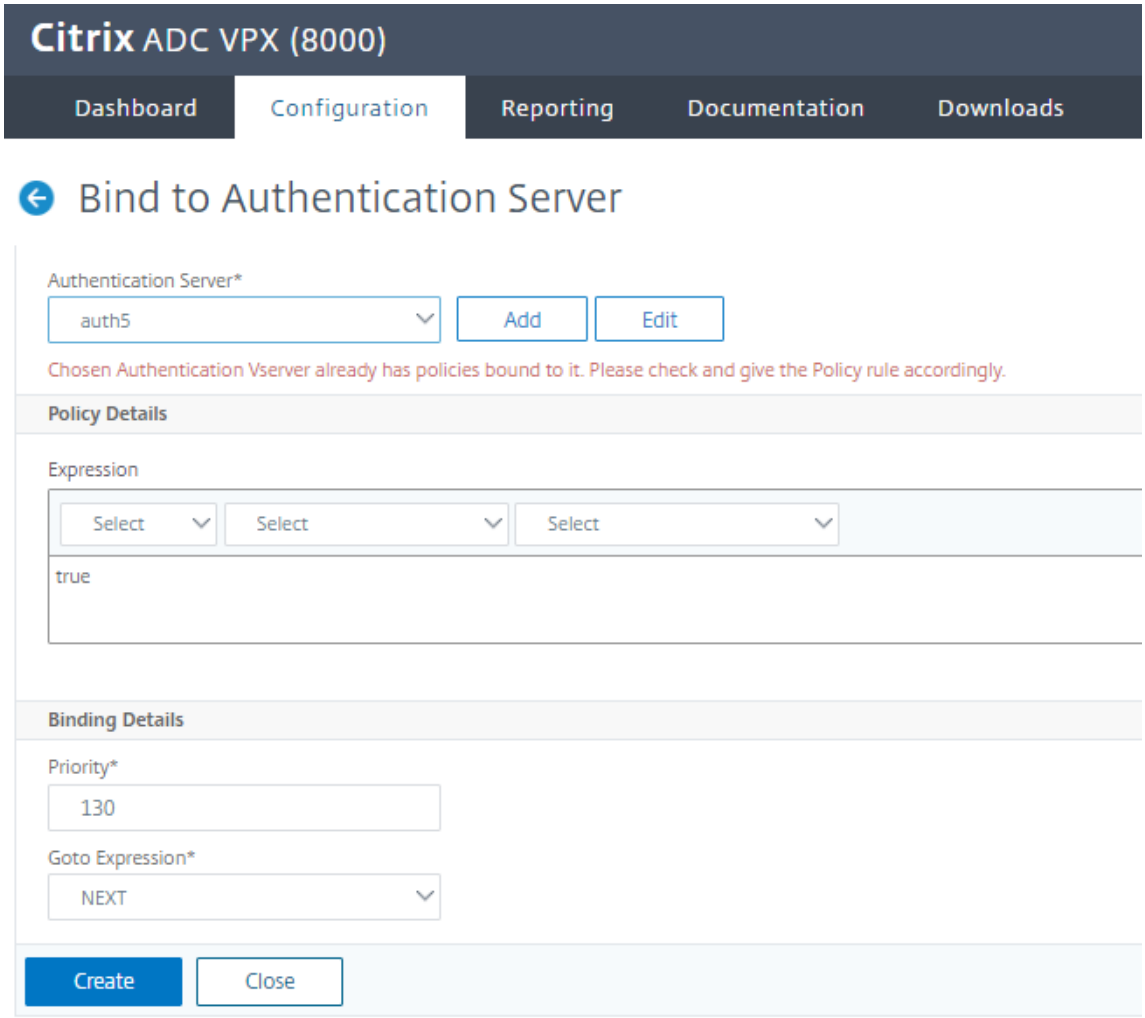


To bind the nFactor flow to authentication server

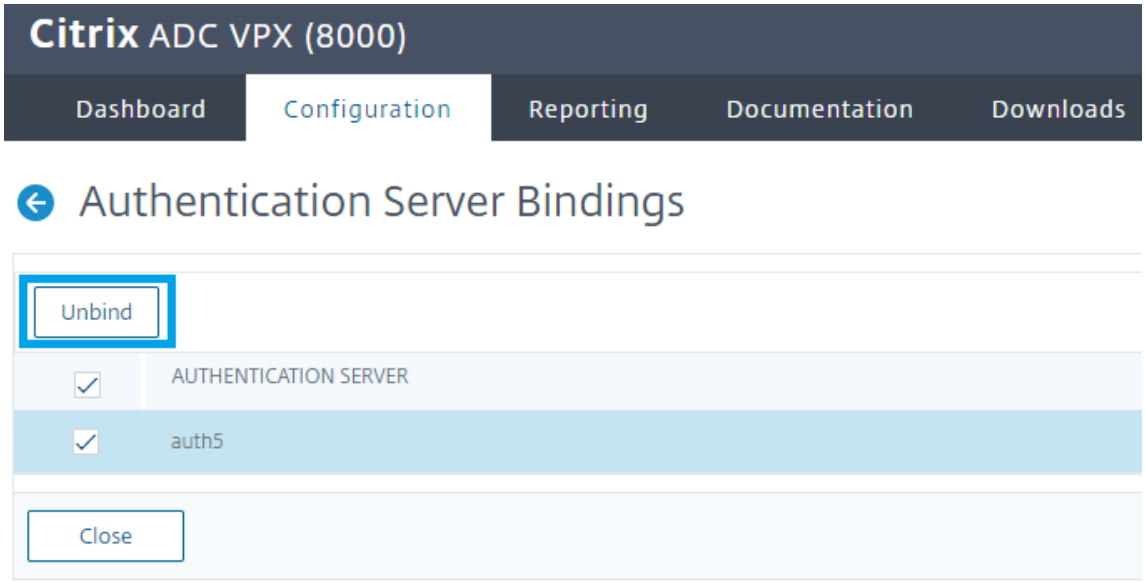
1. On the **nFactor Flows** page, select an nFactor flow that you prefer to bind to an authentication virtual server.
2. Click the hamburger icon to select **Bind to Authentication Server** option or in the details pane, click **Bind to Authentication Server**.



3. On the **Bind to Authentication Server** page, you can perform the following actions:
   - To add a **Authentication Virtual Server**, click **Add**.
   - To select an existing authentication server from the list, click **Authentication Server** field.



4. Click **Show Bindings** from the hamburger icon to view the bindings.
5. To unbind the authentication server from the specific nFactor flow, perform the following steps:
   - On the **nFactor Flows** page, click **Show Bindings** from the hamburger icon.
   - On the **Authentication Server Bindings** page, select the authentication server to unbind and click **Unbind**. Click **Close**.



For more information on nFactor authentication, see the following topics:

- Concept: Multi-Factor (nFactor) authentication.
- Workflow: How nFactor authentication works.
- Configuration: Configuring nFactor authentication.

Enhancements to the nFactor Visualizer

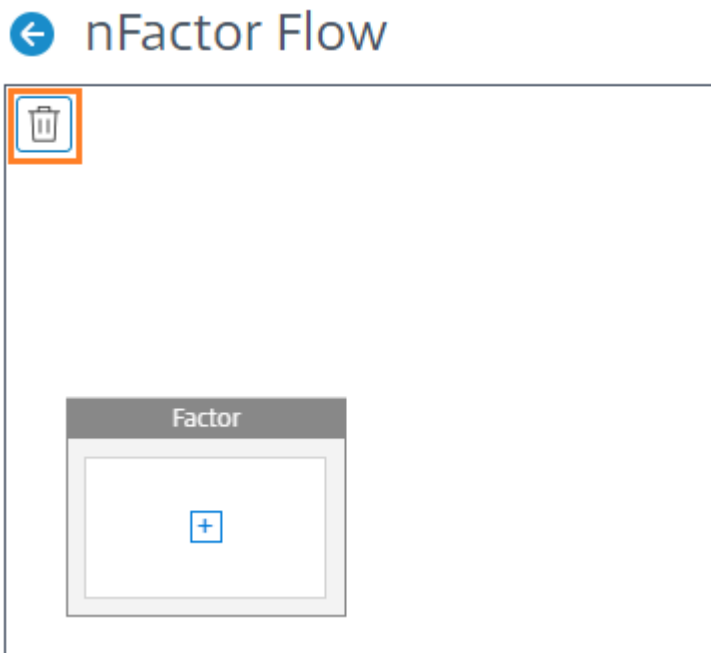
Starting from Citrix ADC release 13.0 build 41.20, the following enhancements are made in the nFactor Visualizer.

- Admins can move the created factors to the trash icon.
- View the nFactor flows in the Authentication Virtual server page.

Trash icon. Admins can only delete the nodes that have no connections. However, the underlying policies or the schemas that are created for the factor are not deleted if the factor is moved to trash.

To view the trash icon,

1. Navigate to **Security > AAA – Application Traffic > nFactor Visualizer > nFactor Flows**.

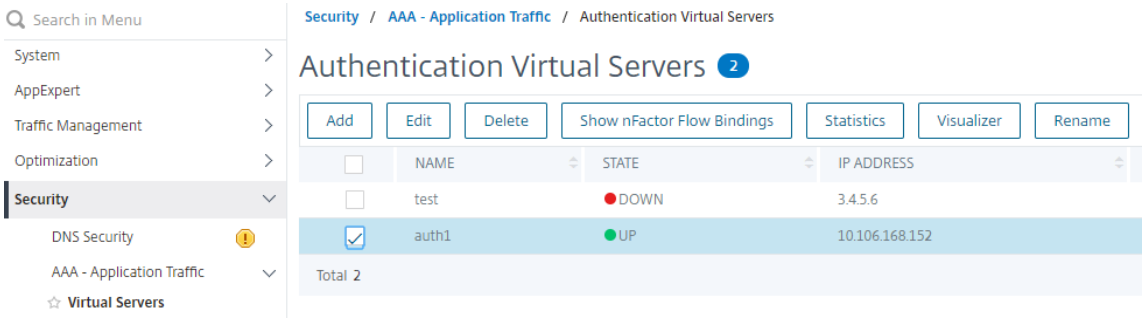


2. To delete the factor, click the factor block and drag it to trash.

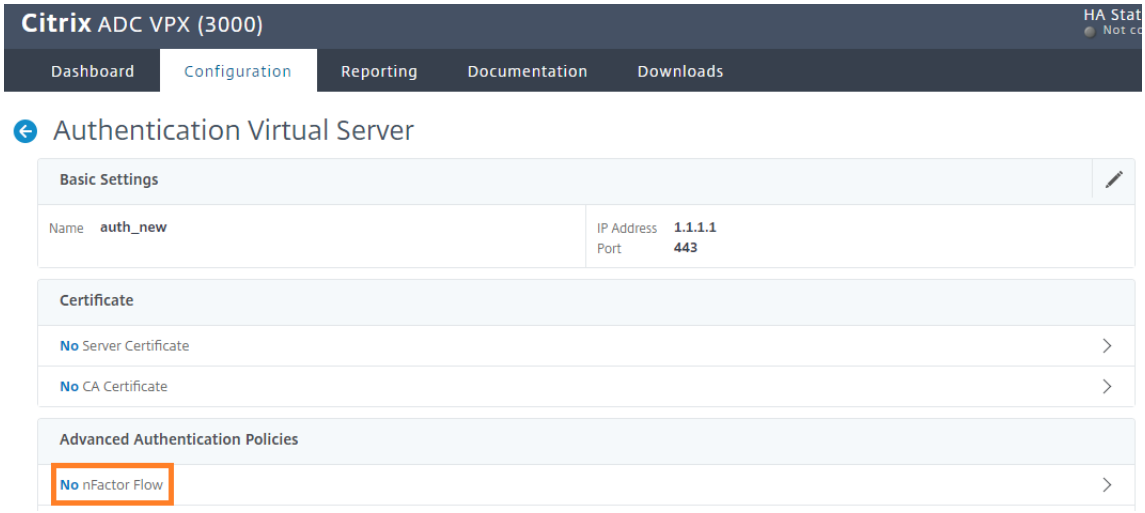
View nFactor flow from Authentication Virtual Server. Admins can also view the created nFactor flows from theAuthentication Virtual Server page.

To view the nFactor flow from Authentication Virtual Server page,

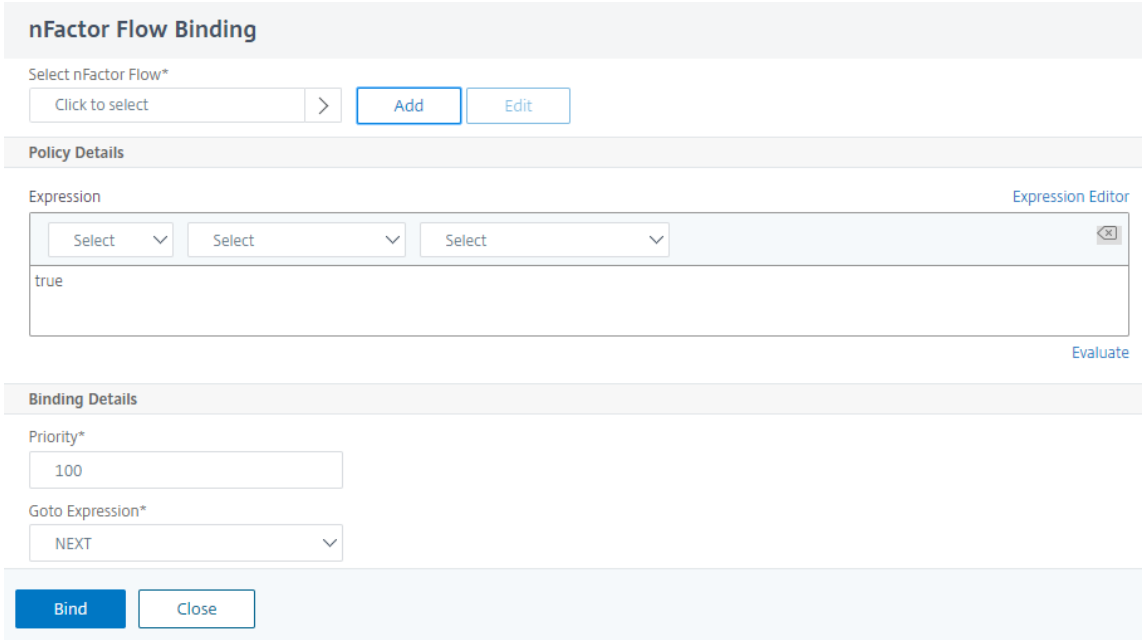
1. Navigate to **Security > AAA – Application Traffic > Virtual Servers**. On the **Authentication Virtual Servers** page, you can perform the following steps:
   - To add an authentication virtual server, click **Add**.
   - To edit an existing authentication virtual server, click **Edit** option from the details pane.



2. On the **Authentication Virtual Server** page, you can view the **nFactor Flow** option under **Advanced Authentication Policies**.



3. If there is no nFactor flow bound to the virtual server, you can click **No nFactor Flow** option under **Advanced Authentication Policies** section to either add a new nFactor flow or select the existing nFactor flow from the list.



nFactor Extensibility

September 14, 2021

nFactor authentication framework provides the flexibility of adding customizations to make the logon interface more intuitive for rich user experience. You can add custom login labels, custom login credentials, customizing UI displays and so on.

With nFactor, each factor can have its own logon screen. In each logon screen you can present any information from any of the previous factors or more information that is invisible in other factors. For example, your last factor can be an informative page where the user reads instructions and click continue.

Before nFactor, custom login pages were limited and customizations and needed support. It was possible to replace the tmindex.html or apply rewrite rules to change some of its behavior. However, it was not possible to achieve the underlying functionality.

The following nFactor related customizations are captured in detail in this topic.

- Customize login labels
- Customize UI to display images
- Customize Citrix ADC nFactor logon form

Assumptions

You are familiar with nFactor, Shell commands, XML, and text editors.

Prerequisites

- Customization described in this topic is possible only when RfWeb UI theme (or theme based) is configured on Citrix ADC.
- Authentication policy must be bound to the authentication, authorization, and auditing virtual server, else the flow does not work as intended. For details, see CTX224241.
- You have the following items related to nFactor
  - XML schema
  - JavaScript
  - Authentication actions
  - Authentication virtual server
  - Citrix ADC version 11.1 and later

Customize logon labels

To customize logon labels, you need the following:

- The XML schema that describes how the logon page looks.
- The script.js file that contains the JavaScript that is used to change the rendering process.

How it works

The JavaScript parses the XML file, rendering each item inside the `<Requirements>` tag. Each element corresponds to a line in the HTML form. For example, a login field is a line, the password field

is another line, and so is the logon button. To introduce new lines, you must specify them in the XML schema file using the StoreFront SDK. The StoreFront SDK allows the logon page with an XML schema to use the <Requirement> tag and define elements on it. These elements allow to use JavaScript to introduce in that space whatever HTML elements are required. In this case, a line is created with some text in the form of HTML.

The XML that can be used is as follows:

```
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<!--NeedCopy-->
```

<Requirement>: Space provided in the logon page. The credential fills the space, and the other parts route the engine into the correct information. In this case, type nsg-custom-cred. This is defined as plain text and the label is defined for its body.
The requirement XML is paired with the JavaScript code to achieve the required results.

```
// Custom Label Handler for Self Service Links
CTXS.ExtensionAPI.addCustomAuthLabelHandler({

getLabelTypeName: function () {
  return "nsg-custom-label";  }
 ,
getLabelTypeMarkup: function (requirements) {

return $("< Your HTML Code Here>");
 }
 ,
// Instruction to parse the label as if it was a standard type
parseAsType: function () {

return "plain";
 }

 }
 );
//Custom Credential Handler for Self Service Links
CTXS.ExtensionAPI.addCustomCredentialHandler({
```



```
22
23 getCredentialTypeName: function () {
24   return "nsg-custom-cred";  }
25  ,
26 getCredentialTypeMarkup: function (requirements) {
27
28 return $("<div/>");
29  }
30  ,
31  }
32  );
33 <!--NeedCopy-->
```

The XML portion indicates the logon page what to display, and the JavaScript code provides the actual text. The credential handler opens up the space and the label fills the space. Because all authentication traffic is now invisible to rewrite and responder, you can change the look and feel of the page. Configuration to customize login labels

1. Create and bind a theme based on RfWeb.

   ```
   1 add vpn portaltheme RfWebUI_MOD -basetheme RfWebUI
   2
   3 bind vpn vserver TESTAAA  - portaltheme RfWebUI_MOD
   4 <!--NeedCopy-->
   ```

   The path for the files based on the theme is available in the directory; /var/netscaler/logon/themes/RfWebUI_MOD

2. Add the following snippet to the end of script.js file:

   Note: Failing to include the preceding lines inside the correct file or missing to include any JavaScript functions prevent the XML from being loaded. The error can only be seen in the Developer Console of the browser with the following text: “Undefined Type nsg-custom-cred.”

   ```
   1 // Custom Label Handler for Self Service Links
   2 CTXS.ExtensionAPI.addCustomAuthLabelHandler({
   3
   4 getLabelTypeName: function () {
   5   return "nsg-custom-label";  }
   6  ,
   7 getLabelTypeMarkup: function (requirements) {
   8
   9 return $("<a href=\"https://identity.test.com/identity/faces/
       register\" style=\"font-size: 16px;\" style=\"text-align:
       center;\">Self Registration</a><br><a href=\"https://identity.
       test.com/identity/faces/forgotpassword\" style=\"font-size: 16
   ```



```
      px;\" style=\"text-align: center;\">Forgot Password</a><br><a
      href=\"https://identity.test.com/identity/faces/forgotuserlogin
      \" style=\"font-size: 16px;\" style=\"text-align: center;\">
      Forgot User Login</a>");
 }
 ,
// Instruction to parse the label as if it was a standard type
parseAsType: function () {

return "plain";
 }

 }
 );
//Custom Credential Handler for Self Service Links
CTXS.ExtensionAPI.addCustomCredentialHandler({

getCredentialTypeName: function () {
  return "nsg-custom-cred";  }
 ,
getCredentialTypeMarkup: function (requirements) {

return $("<div/>");
 }
 ,
 }
 );
<!--NeedCopy-->
```

Loginschema used in this example

```
<?xml version="1.0" encoding="utf-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response
   /1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext/>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/Citrix/Authentication/ExplicitForms/CancelAuthenticate
   </CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement>
```



```
<Credential>
<ID>login</ID>
<SaveID>Username</SaveID>
<Type>username</Type>
</Credential>
<Label>
<Text>User name</Text>
<Type>plain</Type>
</Label>
<Input>
<AssistiveText>Please supply either domain\username or user@fully.
   qualified.domain</AssistiveText>
<Text>
<Secret>false</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue></InitialValue>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<ID>passwd</ID>
<SaveID>Password</SaveID>
<Type>password</Type>
</Credential>
<Label>
<Text>Password:</Text>
<Type>plain</Type>
</Label>
<Input>
<Text>
<Secret>true</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue/>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
```



```
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<Requirement>
<Credential>
<ID>loginBtn</ID>
<Type>none</Type>
</Credential>
<Label>
<Type>none</Type>
</Label>
<Input>
<Button>Please Log On</Button>
</Input>
</Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

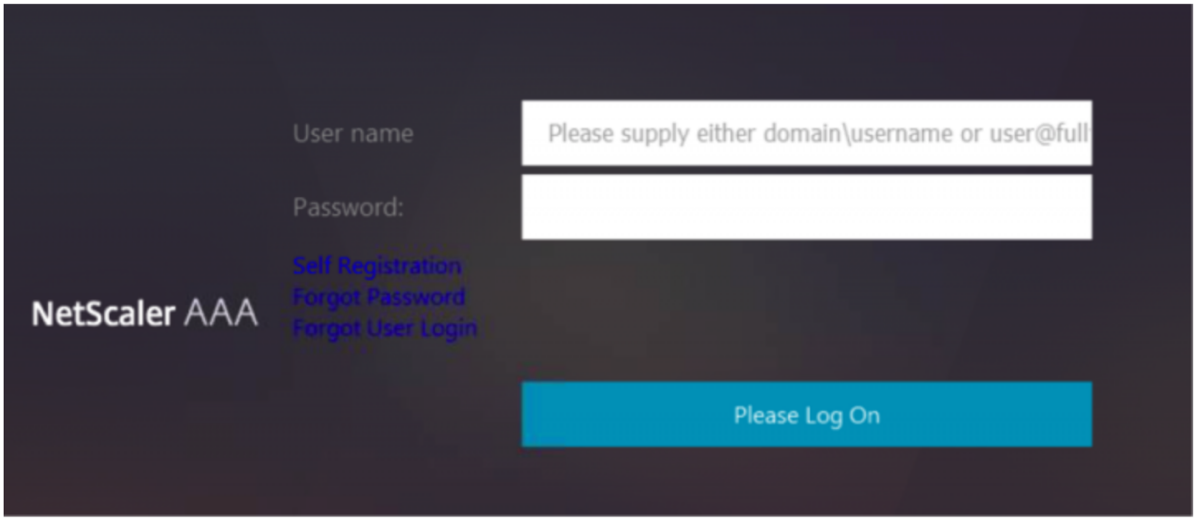
Execute the following commands to load custom schema to config.

```
add authentication loginSchema custom -authenticationSchema custom.xml

add authentication loginSchemaPolicy custom -rule true -action custom

bind authentication vserver AAATEST -policy custom -priority 100 -
   gotoPriorityExpression END
<!--NeedCopy-->
```

The following figure displays the login page that is rendered with this configuration.



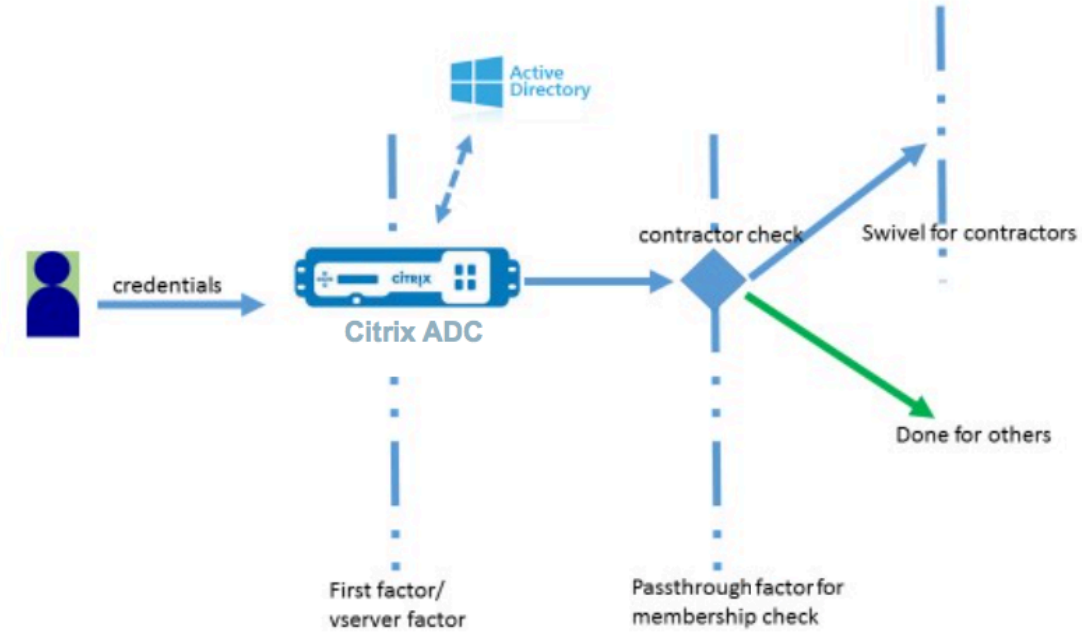
Customize UI to display images

nFactor allows for customized display with the use of loginschema files. There might be a requirement for further customizations other than those offered by the built-in loginschema files. For example, displaying a hyperlink or writing custom logic in the UI. These can be achieved using 'custom credentials' that comprise of loginschema extension and corresponding javascript file.

For UI customization to display images, a deployment flow in "Citrix ADC-Swivel" integration is used as an example.

There are two factors in this flow.

- First factor: Checks user's AD credentials.
- Second factor: Prompts for user logon based on group membership.



In this flow, all users go through first factor. Before second factor, there is a pseudo factor to check if some users can be omitted from “swivel” factor. If user requires “swivel” factor, an image and a text box are displayed to enter code.

Solution

The solution for customizing UI to display images contains two parts;

- Loginschema extension.
- Custom script to process the loginschema extension.

Loginschema extension

To control form rendering, a custom ‘id’/’credential’ is injected into the loginschema. This can be done by reusing existing schema and modifying as per the requirement.

In the example, a loginschema that has only one text field (such as /nsconfig/loginschema/LoginSchema/OnlyPassword.xml) is considered.
The following snippet is added to the loginschema.

```
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><
   Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<!--NeedCopy-->
```


In the snippet, “swivel_cred” is specified as “Type” of the credential. Because this is not recognized as a built-in ‘credential,’ UI looks for a handler for this type, and calls it if it exists.
An initial value is sent for this credential which is an expression that Citrix ADC dynamically fills. In the example, it is the user’s name used to notify swivel server of the user name. It might not be needed all the time or it can be augmented with some other data. Those details must be added as required.

Javascript to handle custom credential

When the UI finds a custom credential, it looks for a handler. All custom handlers are written in /var/netscaler/logon/LogonPoint/custom/script.js for default portal theme.
For custom portal themes, script.js can be found in the directory /var/netscaler/logon/themes/<custom_theme>/.

The following script is added to render mark-up for custom credentials.

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the server
    getCredentialTypeName: function () {
  return "swivel_cred";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        var image = $("<img/>");
        var username = requirements.input.text.initialValue; //Get the secret from the response
        image.attr({

            "style" : "width:200px;height:200px;",
            "id" : "qrcodeimg",
            "src" : "https://myswivelserver.citrix.com:8443/pinsafe/SCImage?username=" + username
         }
 );
        div.append(image);
        return div;
     }

 }
 );
<!--NeedCopy-->
```


This snippet is for handling the mark-up for ‘swivel_cred’. Credential name highlighted must match the ‘type’ specified earlier in the loginschema extension.
To generate mark-up, an image whose source points to swivel server needs to be added. Once this is done, UI loads image from specified location. Because this loginschema also has a textbox, UI renders that text box.

Note: Administrator can modify “style” of image element to resize the image. Currently it configured for 200x200 pixels.

Configuration for customizing UI to display images

nFactor configuration is better constructed bottom-up, that is the last factor first because when you try to specify ‘nextFactor’ for the previous factors, you require the subsequent factor’s name.

Swivel factor configuration:

```
add loginschema swivel_image -authenticationSchema /nsconfig/
    loginschema/SwivelImage.xml

add authentication policylabel SwivelFactor -loginSchema swivel_image

bind authentication policylabel SwivelFactor -policy <policy-to-check-
    swivel-image> -priority 10
<!--NeedCopy-->
```

Note: Download SwivelImage.xml from the loginschema used in the example.

Pseudo factor for group check configuration:

```
add authentication policylabel GroupCheckFactor

add authentication policy contractors_auth_policy -rule 'http.req.
    user.is_member_of( "contractors" )' -action NO_AUTHN

add authentication policy not_contractors _auth_policy -rule true -
    action NO_AUTHN

bind authentication policylabel GroupCheckFactor -policy
    contractors_auth_policy -pri 10 -nextFactor SwivelFactor

bind authentication policylabel GroupCheckFactor -policy
    not_contractors_auth_policy -pri 20
<!--NeedCopy-->
```

First factor for Active Directory login:


```
add ldapAction <>

add authentication policy user_login_auth_policy -rule true -action <>

bind authentication vserver <> -policy user_login_auth_policy -pri 10 -nextFactor GroupCheckFactor
<!--NeedCopy-->
```

In the configuration, three factors are specified of which one is implicit/pseudo.

Loginschema used in this example

The following is an example schema with swivel credential and a text box.

Note: When copying data for web browser, quotes might be displayed differently. Copy data in editors like notepad before saving them to files.

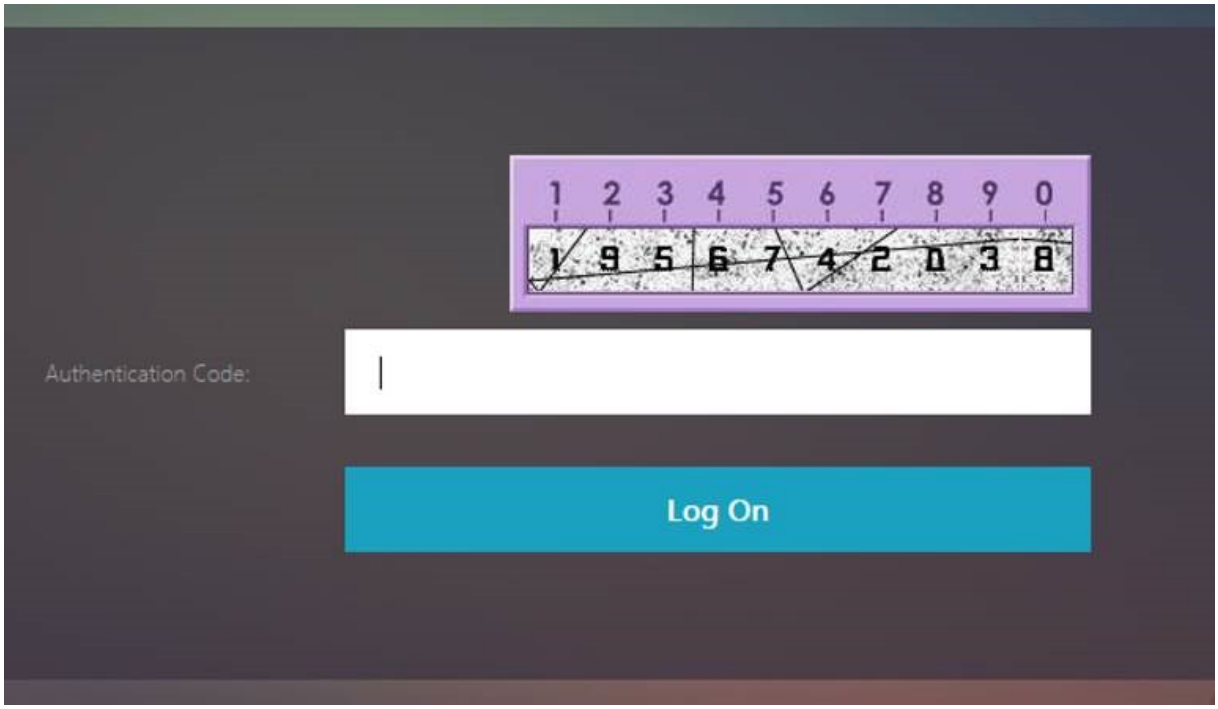
```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password</SaveID><Type>password</Type></Credential><Label><Text>Password:</Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>Hello ${
 http.req.user.name }
 , Please enter passcode from above image.</Text><Type>confirmation</Type></Label><Input /></Requirement>
```



```
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
    </Type></Credential><Label><Text>Remember my password</Text><Type>
    plain</Type></Label><Input><CheckBox><InitialValue>false</
    InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
    ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
    Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

Output

Once the configuration is performed, the following image is displayed.



Note: Image height and placement can be altered in the JavaScript.

Customize Citrix ADC nFactor logon form to show or hide fields

Citrix Gateway's RfWeb UI allows for wide variety of customizations. This capability when combined with nFactor authentication framework lets customers configure complex flows without compromising existing workflows.

In this example, two authentication options, OAuth and LDAP are available from the Logon Type list. When the form is first loaded, user name and password fields (LDAP is shown first) are displayed. If OAuth is selected, all the fields are hidden because OAuth implies offload of authentication to a third party server. This way, administrator can configure intuitive workflows as per user convenience.

Note:

- The values in the Logon Type list can be modified with simple modifications to the script file.
- This section describes only the UI part of the flow. The run time handling of the authentication is outside the scope of this article. Users are recommended to refer to nFactor documentation for authentication configuration.

How to customize nFactor logon form

Customizing nFactor logon form can be classified into two parts

- Sending right loginschema to the UI
- Writing a handler to interpret loginschema and user selections

Send right loginschema to the UI

In this example, a simple claim/requirement is sent in the loginschema.

For this, SingleAuth.xml file is modified. SingleAuth.xml is shipped with Citrix ADC firmware and can be found in /nsconfig/loginschema/LoginSchema directory.

Steps to send loginschema:

1. Log in via SSH and drop to shell (type 'shell').
2. Copy SingleAuth.xml to a different file for modification.

   Note: The destination folder is different from the default Citrix ADC loginschemas folder.

   cp /nsconfig/loginschema/LoginSchema/SingleAuth.xml /nsconfig/loginschema/SingleAuth-Dynamic.xml
3. Add the following claim to SingleAuthDynamic.xml.

```
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></Label></Requirement>
<!--NeedCopy-->
```

4. Configure Citrix ADC to send this loginschema to load the first form.

```
add loginschema single_auth_dynamic -authenticationSchema SingleAuthDynamic.xml
```



```
2
3 add loginschemaPolicy single_auth_dynamic -rule true -action
     single_auth_dynamic
4
5 bind authentication vserver aaa_nfactor -policy
     single_auth_dynamic -pri 10
6 <!--NeedCopy-->
```

Script changes to load form and handle user events

You can modify the JavaScript that enables administrator to customize display for logon form. In this example, user name and password field are displayed if LDAP is chosen and are hidden if OAuth is chosen. Administrator can also hide only the password.
Admins must append the following snippet to “script.js” that is at “/var/netscaler/logon/LogonPoint/-custom” directory.

Note: Because this directory is a global directory, create a portal theme and edit the “script.js” file within that folder, at "/var/netscaler/logon/themes/<THEME_NAME>".

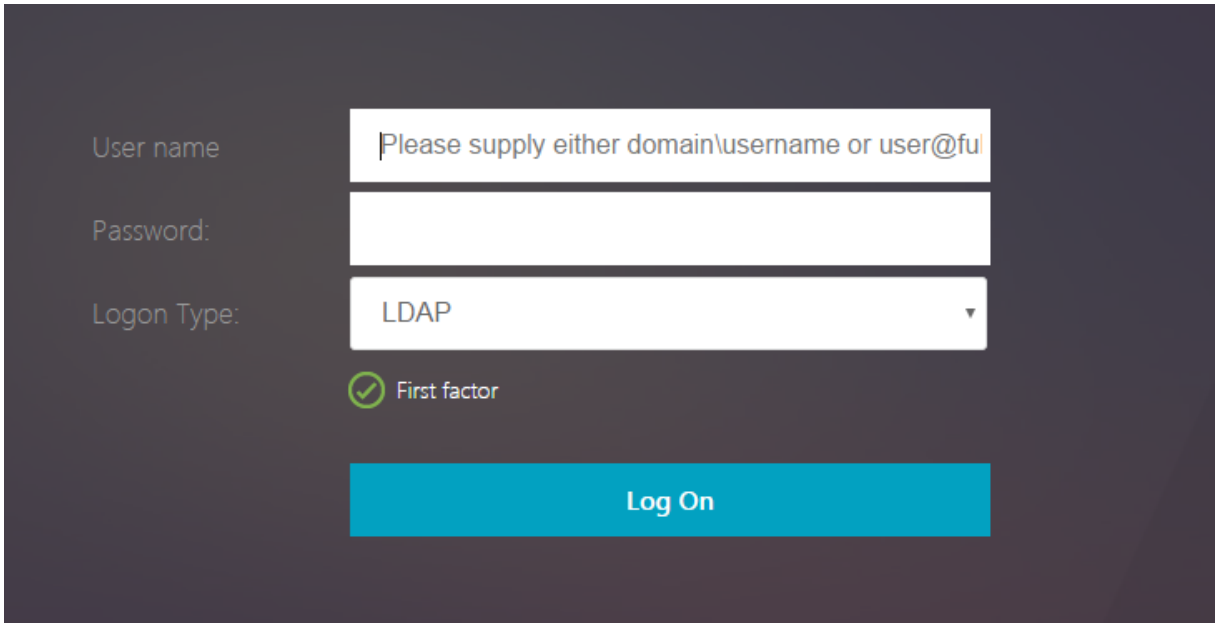
```
1 CTXS.ExtensionAPI.addCustomCredentialHandler({
2
3     // The name of the credential, must match the type returned by the
        server
4     getCredentialTypeName: function () {
5  return "nsg_dropdown";  }
6 ,
7     // Generate HTML for the custom credential
8     getCredentialTypeMarkup: function (requirements) {
9
10         var div = $("<div></div>");
11         var select = $("<select name='nsg_dropdown'></select>").attr("
              id", "nsg_dropdown");
12
13         var rsa = $("<option></option>").attr("selected", "selected").
              text("LDAP").val("LDAP");
14         var OAuthID = $("<option></option>").text("OAuth").val("OAuth")
              ;
15         select.append(rsa, OAuthID);
16
17         select.change(function(e) {
18
19             var value = $(this).val();
20             var ldapPwd = $($(".credentialform").find(".
                  CredentialTypepassword")[0]);
```



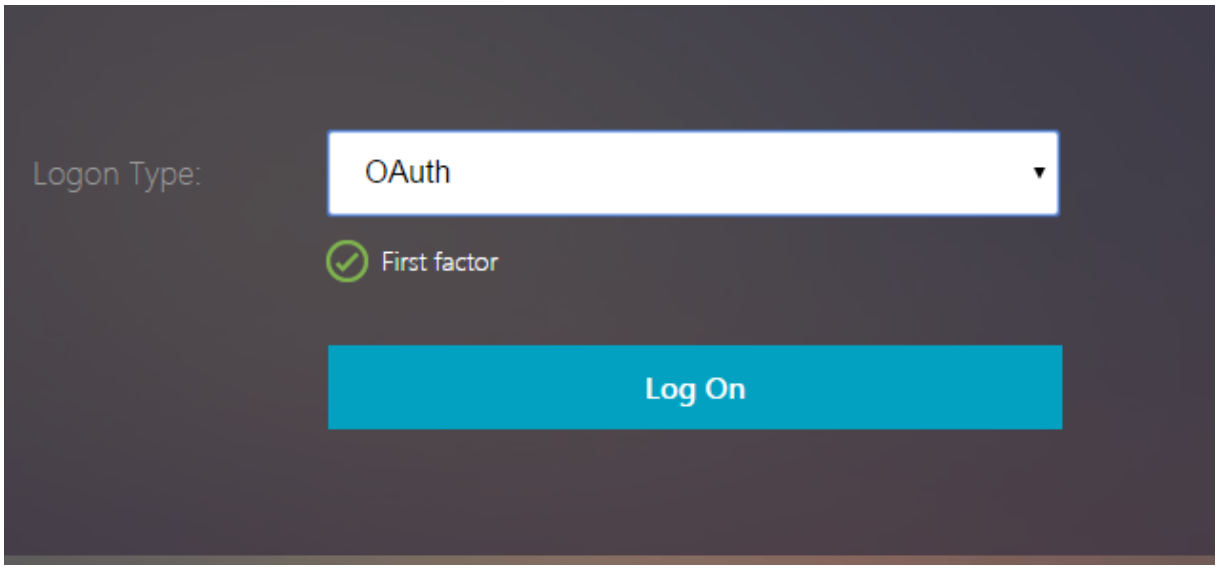
```
21              var ldapUname = $($(".credentialform").find(".
                   CredentialTypeusername"));
22              if(value == "OAuth") {
23
24                  if (ldapPwd.length)
25                          ldapPwd.hide();
26                  if (ldapUname.length)
27                          ldapUname.hide();
28               }
29    else if(value == "LDAP") {
30
31                  if (ldapPwd.length)
32                          ldapPwd.show();
33                  if (ldapUname.length)
34                          ldapUname.show();
35               }
36
37           }
38  );
39        div.append(select);
40         return div;
41      }
42
43  }
44  );
45 <!--NeedCopy-->
```

End-user experience

When an end user loads the logon page first time, the following screen appears.



If **OAuth** is selected in **Logon Type**, user name and password fields are hidden.



If **LDAP** is selected, user name and password are displayed. This way, the logon page can be dynamically loaded based on user selection.

Loginschema used in this example

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
```



```
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</
   SaveID><Type>username</Type></Credential><Label><Text>User name</
   Text><Type>plain</Type></Label><Input><AssistiveText>Please supply
   either domain\username or user@fully.qualified.domain</AssistiveText
   ><Text><Secret>false</Secret><ReadOnly>false</ReadOnly><InitialValue
   ></InitialValue><Constraint>.+</Constraint></Text></Input></
   Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password
   </SaveID><Type>password</Type></Credential><Label><Text>Password:</
   Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><
   ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint
   >.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type
   ></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></
   Label></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
   First factor</Text><Type>confirmation</Type></Label><Input /></
   Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
   </Type></Credential><Label><Text>Remember my password</Text><Type>
   plain</Type></Label><Input><CheckBox><InitialValue>false</
   InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
   ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
   Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

> **Important:** For more details about various nFactor related topics, see nFactor.

Set a cookie using nFactor

September 14, 2021

You can apply the nFactor custom labels and set a cookie as a factor of the authentication flow.

Through custom labels, you can use JavaScript to manipulate the login schema.

To set a cookie as a factor, you do not need to display any information to the user, which is performed with a no schema login. Instead, you must interact with the user's browser to instruct the login schema to store the desired data. A login schema is required to set the cookie when the page is loaded. The cookie is set with a custom label and JavaScript code.

To implement a factor that sets a cookie, create an XML file called cookie.xml to store the schema in the /nsconfig/loginschema/ directory with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response
   /1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>

<Requirement>
<Credential><ID>nsg_cookie</ID><Type>nsg_cookie</Type></Credential>
<Label><Text>Logon Type:</Text><Type>Plain</Type></Label>
</Requirement>

<Requirement>
<Credential><ID>loginBtn</ID><Type>none</Type></Credential>
<Label><Type>none</Type></Label><Input><Button>Log On</Button></Input>
</Requirement>

</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

In this XML;

- The custom label nsg_cookie is used to create the cookie and submit the form, and the form button.
- The RfWebUI_custom is the new Portal theme based on the RfWebUI theme.

Steps to set a cookie using nFactor

1. Create a portal theme based on the RfWebUI theme.

```
add vpn portaltheme RfWebUI_custom -basetheme RfWebUI
<!--NeedCopy-->
```

   This command creates a folder for this theme at /var/netscaler/logon/themes/RfWebUI_custom

2. Edit the file /var/netscaler/logon/themes/RfWebUI_custom/script.js and add the following script:

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by
        the server
    getCredentialTypeName: function () {
  return "nsg_cookie";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        $(document).ready(function() {

        //Set cookie valid for 1000 days
        var exdays = 1000;
        var d = new Date();
        d.setTime(d.getTime() + (exdays*24*60*60*1000));
        var expires = "expires="+ d.toUTCString();
        document.cookie = "NSC_COOKIE_NAME=CookieValeu;" + expires
            + ";path=/";

        //Submit form
        document.getElementById('loginBtn').click();
         }
 );
        return div;
     }

 }
 );
<!--NeedCopy-->
```

   This code performs the following:

- Waits for the browser to finish loading the page
- Sets a cookie called NSC_COOKIE_NAME with the value CookieValue, valid for 1000 days
- Auto-submits the form.

The cookie is created and the user does not need to interact with the page.

3. Create a login schema to bind to the policy label that represents the set cookie factor.

```
1 add authentication loginSchema Cookie_LS -authenticationSchema "/
     nsconfig/loginschema/cookie.xml"
2 <!--NeedCopy-->
```

4. Create a NO_AUTHN authentication policy to bind to the policy label that represents the set cookie factor.

```
1 add authentication Policy NO_AUTHN_POL -rule TRUE -action NO_AUTHN
2 <!--NeedCopy-->
```

This policy always evaluates as true, moving the user to the next factor or completing the authentication flow.

5. Bind the portal theme RfWebUI_custom to the Citrix Gateway virtual server or Citrix ADC AAA virtual server.

Sample deployments using nFactor authentication

September 14, 2021

The following are the sample deployments using nFactor authentication:

- Getting two passwords up-front, pass-through in next factor. Read
- Group extraction followed by certificate or LDAP authentication, based on group membership. Read
- SAML followed by LDAP or certificate authentication, based on attributes extracted during SAML. Read
- SAML in first factor, followed by group extraction, and then LDAP or certificate authentication, based on groups extracted. Read
- Prefilling user name from certificate. Read
- Certificate authentication followed by group extraction for 401 enabled traffic management virtual servers. Read
- Username and two passwords with group extraction in third factor. Read
- Certificate fallback to LDAP in same cascade; one virtual server for both certificate and LDAP authentication. Read

- LDAP in first factor and WebAuth in second factor. Read
- Domain drop down in first factor, then different policy evaluations based on group. Read

How to articles

September 14, 2021

The Authentication, authorization, and auditing "How to articles" are simple, relevant, and easy to implement articles. These articles contain information about some of the popular Authentication, authorization, and auditing features such LDAP authentication and multifactor authentication. For some of the popular articles on configuring and troubleshooting authentication through Citrix ADC, see Citrix ADC Authentication: How do I?

Endpoint Analysis

Configure pre-authentication Endpoint Analysis scan as a factor in nFactor authentication

Configure post-authentication Endpoint Analysis scan as a factor in Citrix ADC nFactor authentication

Configure pre-authentication and post-authentication EPA scan as a factor in nFactor authentication

Configure periodic Endpoint Analysis scan as a factor in nFactor authentication

Configure Citrix Gateway preauthentication EPA scan for the domain check

First factor and second factor configuration combinations

Configure nFactor for Citrix Gateway with WebAuth in first factor and LDAP with password change in second factor

Configure SAML followed by LDAP or certificate authentication based on SAML attribute extraction in nFactor authentication

Configure certificate authentication as first factor and LDAP as second factor in Citrix ADC nFactor authentication

Configure two-factor authentication with one login schema and one passthrough schema in Citrix ADC nFactor authentication

Configure user name and two passwords with group extraction in third factor by nFactor authentication

Configure domain drop-down, username, and password field in the first factor and policy evaluation based on groups in the next factor

Configure email ID (or user name) input based group extraction at first factor to decide the next factor authentication flow

Configure a domain drop-down list for user input in the first factor to decide the next factor authentication flow

EULA as an authentication factor

Configure EULA as an authentication factor in Citrix ADC nFactor system

Prefill user name from certificate

Configure prefill user name from certificate in Citrix ADC nFactor authentication

Step-up authentication

Configure nFactor for applications with different login site requirements including step-up authentication

SAML authentication

September 14, 2021

Security Assertion Markup Language (SAML) is an XML-based authentication mechanism that provides single sign-on capability and is defined by the OASIS Security Services Technical Committee.

> Note
>
> Starting from NetScaler 12.0 Build 51.x, Citrix ADC appliance used as a SAML Service Provider (SP) with Multi-Factor (nFactor) authentication now prepopulates the user-name field on the login page. The appliance sends a NameID attribute as part of a SAML authorization request, retrieves the NameID attribute value from the Citrix ADC SAML Identity Provider (IdP), and prepopulates the user-name field.

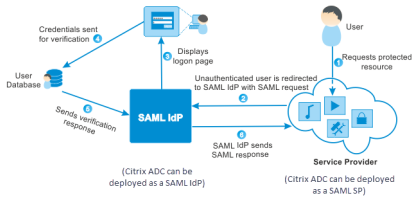
Why use SAML authentication

Consider a scenario in which a service provider (LargeProvider) hosts a number of applications for a customer (BigCompany). BigCompany has users that must seamlessly access these applications. In a traditional setup, LargeProvider would need to maintain a database of users of BigCompany. This raises some concerns for each of the following stakeholders:

- LargeProvider must ensure security of user data.

- BigCompany must validate the users and keep the user data up-to-date, not just in its own database, but also in the user database maintained by LargeProvider. For example, a user removed from the BigCompany database must also be removed from the LargeProvider database.
- A user has to log on individually to each of the hosted applications.

The SAML authentication mechanism provides an alternative approach. The following deployment diagram shows how SAML works (SP initiated flow).



The concerns raised by traditional authentication mechanisms are resolved as follows:

- LargeProvider does not have to maintain a database for BigCompany users. Freed from identity management, LargeProvider can concentrate on providing better services.
- BigCompany does not bear the burden of making sure the LargeProvider user database is kept in sync with its own user database.
- A user can log on once, to one application hosted on LargeProvider, and be automatically logged on to the other applications that are hosted there.

The Citrix ADC appliance can be deployed as a SAML Service Provider (SP) and a SAML Identity Provider (IdP). Read through the relevant topics to understand the configurations that must be performed on the Citrix ADC appliance.

A Citrix ADC appliance configured as a SAML service provider can now enforce an audience restriction check. The audience restriction condition evaluates to “Valid” only if the SAML replying party is a member of at least one of the specified audiences.

You can configure a Citrix ADC appliance to parse attributes in SAML assertions as group attributes. Parsing them as group attributes enables the appliance to bind policies to the groups.

Citrix ADC as a SAML SP

September 14, 2021

The SAML Service Provider (SP) is a SAML entity deployed by the service provider. When a user tries to access a protected application, the SP evaluates the client request. If the client is unauthenticated (does not have a valid NSC_TMAA or NSC_TMAS cookie), the SP redirects the request to the SAML Identity Provider (IdP).

The SP also validates SAML assertions that are received from the IdP.

When the Citrix ADC appliance is configured as an SP, all user requests are received by a traffic management virtual server (load balancing or content switching) that is associated with the relevant SAML action.

The Citrix ADC appliance also supports POST and Redirect bindings during logout.

> **Note**
>
> A Citrix ADC appliance can be used as a SAML SP in a deployment where the SAML IdP is configured either on the appliance or on any external SAML IdP.

When used as a SAML SP, a Citrix ADC appliance:

- Can extract the user information (attributes) from the SAML token. This information can then be used in the policies that are configured on the Citrix ADC appliance. For example, if you want to extract the GroupMember and emailaddress attributes, in the SAMLAction, specify the **Attribute2** parameter as GroupMember and the **Attribute3** parameter as emailaddress.

  > **Note**
  >
  > Default attributes such as username, password, and logout URL must not be extracted in attributes 1–16, because they as are implicitly parsed and stored in the session.

- Can extract attribute names of up to 127 bytes from an incoming SAML assertion. The previous limit was 63 bytes.
- Supports post, redirect, and artifact bindings.

  > **Note**
  >
  > Redirect binding should not be used for large amount of data, when the assertion after inflate or decoding is greater than 10K.

- Can decrypt assertions.
- Can extract multi-valued attributes from a SAML assertion. These attributes are sent is nested XML tags such as:

  <AttributeValue> <AttributeValue>Value1</AttributeValue>
  <AttributeValue>Value2</AttributeValue>
  </AttributeValue>

  > **Note**
  >
  > From Citrix ADC 13.0 Build 63.x and above, the individual maximum length for SAML attributes has been increased to allow a maximum of 40k bytes. The size of all the attributes must not exceed 40k bytes.

When presented with previous XML, the Citrix ADC appliance can extract both Value1 and Value2 as values of a given attribute, as opposed to the old firmware that extracts only Value1.

- Can specify the validity of a SAML assertion.

  If the system time on Citrix ADC SAML IdP and the peer SAML SP is not in sync, the messages might get invalidated by either party. To avoid such cases, you can now configure the time duration for which the assertions are valid.

  This duration, called the "skew time," specifies the number of minutes for which the message should be accepted. The skew time can be configured on the SAML SP and the SAML IdP.

- Can send extra attribute called 'ForceAuth' in the authentication request to external IdP (Identity Provider). By default, the ForceAuthn is set to 'False'. It can be set to 'True' to suggest IdP to force authentication despite existing authentication context. Also, Citrix ADC SP does authentication request in query parameter when configured with artifact binding.

To configure the Citrix ADC appliance as a SAML SP by using the command line interface

1. Configure a SAML SP action.

   Example

   The following command adds a SAML action that redirects unauthenticated user requests.

   ```
   add authentication samlAction SamlSPAct1 -samlIdPCertName nssp -samlSigningCertNa
    nssp -samlRedirectUrl https://auth1.example.com -relaystateRule "AAA.
   LOGIN.RELAYSTATE.EQ(\"https://lb.example1.com/\")"
   ```

   Points to note

   - Certificate provided for `-samlIdPCertName` in the samlAction command must match the corresponding certificate from IdP for the signature verification to succeed.
   - SAML supports only RSA certificate. Other certificates like HSM, FIPS, and so on are not supported.
   - Citrix recommends to have a full domain name with trailing '/' in the expression.
   - Administrators must configure an expression for **relaysStateRule** in the samlAction command. The expression must contain the list of published domains that the user connects to before being redirected to the authentication virtual server. For example, the expression must contain the domains of the front-end virtual server (VPN, LB, or CS) that use this SAML action for authentication.

   For more details on the command, see https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction and https://support.citrix.com/article/CTX316577.

2. Configure the SAML policy.

   Example

   The following command defines a SAML policy that applies the previously defined SAML action to all traffic.

   ```
   add authentication policy SamlSPPol1 -rule true -action SamlSPAct1
   ```

3. Bind the SAML policy to the authentication virtual server.

   Example

   The following command binds the SAML policy to an authentication virtual server named "av_saml".

   ```
   bind authentication vserver av_saml -policy SamlSPPol1
   ```

4. Bind the authentication virtual server to the appropriate traffic management virtual server.

   Example

   The following command adds a load balancing virtual server named "lb1_ssl" and associates the authentication virtual server named "av_saml" to the load balancing virtual server.

   ```
   add lb vserver lb1_ssl SSL 10.217.28.224 443 -persistenceType NONE -
   cltTimeout 180 -AuthenticationHost auth1.example.com -Authentication ON
    -authnVsName av_saml
   ```

   For more details on the command, see https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction

To configure a Citrix ADC appliance as a SAML SP by using the GUI

1. Navigate to **Security>AAA-Policies>Authentication>Basic Policies>SAML**.
2. Select **Servers** tab, click **Add**, enter values for the following parameters, and click **Create**.

   Parameter description:

   Name - Name of the server

   Redirect URL - URL that users will authenticate against. Some IdP's have special URLs that are not reachable unless under SAML setup.

   Single Logout URL - URL specified so that the Citrix ADC can recognize when to send the client back to the IdP to complete the Sign out process. We will not use it in this simple deployment.

   SAML Binding - Method that is be used to move the client from the SP to the IdP. This needs to be the same on the IdP so that it understands how the client will connect to it.
   When the Citrix ADC acts as an SP, it supports POST, REDIRECT and ARTIFACT bindings.

Logout Binding - REDIRECT

IDP Certificate Name - IdPCert Certificate (Base64) present under SAML Signing Certificate.

User Field - Section of the IdP's SAML authentication form that contains the username for SP to extract if required.

Signing Certificate Name - Select the SAML SP certificate (with private key) that Citrix ADC uses to sign authentication requests to the IdP. The same certificate (without private key) must be imported to the IdP, so that the IdP can verify the authentication request signature. This field is not needed by most IdPs.

IssuerName - Identifier. Unique ID that is specified on both the SP and IdP to help identify the Service Provider to each other.

Reject unsigned assertion - Option that you can specify if you require the Assertions from the IdP to be signed. You can ensure that only the Assertion needs to be signed (ON) or both the Assertion and Response from the IdP need to be signed (STRICT).

Audience - Audience for which assertion sent by IdP is applicable. This is typically entity name or URL that represents ServiceProvider.

Signature Algorithm - RSA-SHA256

Digest Method - SHA256

Default Authentication Group - The default group that is chosen when the authentication succeeds in addition to extracted groups.

Group Name Field - Name of the tag in assertion that contains user groups.

Skew Time (mins) - This option specifies the allowed clock skew in number of minutes that Citrix ADC ServiceProvider allows on an incoming assertion.

3. Similarily, create a corresponding SAML policy and bind it to the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the SAML policy with the authentication virtual server.

4. Associate the authentication server with the appropriate traffic management virtual server.

   Navigate to **Traffic Management > Load Balancing** (or **Content Switching**) > **Virtual Servers**, select the virtual server, and associate the authentication virtual server with it.

Citrix ADC as a SAML IdP

September 14, 2021

The SAML IdP (Identity Provider) is a SAML entity that is deployed on the customer network. The IdP receives requests from the SAML SP and redirects users to a logon page, where they must enter their credentials. The IdP authenticates these credentials with the active directory (external authentication server, such as LDAP) and then generates a SAML assertion that is sent to the SP.

The SP validates the token, and the user is then granted access to the requested protected application.

When the Citrix ADC appliance is configured as an IdP, all requests are received by an authentication virtual server that is associated with the relevant SAML IdP profile.

> **Note**
>
> A Citrix ADC appliance can be used as a IdP in a deployment where the SAML SP is configured either on the appliance or on any external SAML SP.

When used as a SAML IdP, a Citrix ADC appliance:

- Supports all authentication methods that it supports for traditional logons.
- Digitally signs assertions.
- Supports single-factor and two-factor authentication. SAML must not be configured as the secondary authentication mechanism.
- Can encrypt assertions by using the public key of the SAML SP. This is recommended when the assertion includes sensitive information.
- Can be configured to accept only digitally signed requests from the SAML SP.
- Can log on to the SAML IdP by using the following 401-based authentication mechanisms: Negotiate, NTLM, and Certificate.
- Can be configured to send 16 attributes in addition to the NameId attribute. The attributes must be extracted from the appropriate authentication server. For each of them, you can specify the name, the expression, the format, and a friendly name in the SAML IdP profile.
- If the Citrix ADC appliance is configured as a SAML IdP for multiple SAML SP, a user can gain access to applications on the different SPs without explicitly authenticating every time. The Citrix ADC appliance creates a session cookie for the first authentication, and every subsequent request uses this cookie for authentication.
- Can send multi-valued attributes in a SAML assertion.
- Supports post and redirect bindings. Support for artifact binding is introduced in Citrix ADC release 13.0 Build 36.27.
- Can specify the validity of a SAML assertion.

  If the system time on Citrix ADC SAML IdP and the peer SAML SP is not in sync, the messages might get invalidated by either party. To avoid such cases, you can now configure the time duration for which the assertions are valid.

This duration, called the “skew time,” specifies the number of minutes for which the message must be accepted. The skew time can be configured on the SAML SP and the SAML IdP.

- Can be configured to serve assertions only to SAML SPs that are pre-configured on or trusted by the IdP. For this configuration, the SAML IdP must have the service provider ID (or issuer name) of the relevant SAML SPs.

> **Note**
>
> Before proceeding, make sure that you have an authentication virtual server that is linked to an LDAP authentication server.

To configure a Citrix ADC appliance as a SAML IdP by using the command line interface

1. Configure a SAML IdP profile.

   Example

   Adding Citrix ADC appliance as an IdP with SiteMinder as the SP.

```
add authentication samlIdPProfile samlIDPProf1 -samlSPCertName siteminder
-cert -encryptAssertion ON -samlIdPCertName ns-cert -assertionConsumerServiceURL
 http://sm-proxy.nsi-test.com:8080/affwebservices/public/saml2assertionconsumer
 -rejectUnsignedRequests ON -signatureAlg RSA-SHA256 -digestMethod
SHA256 -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re##^https://
example2\.com/cgi/samlauth$##)
```

   Points to note

   - In SAML IdP profile, configure **acsURLRule** that takes an expression of the list of applicable service provider URLs for this IdP. This expression depends on the SP being used. If Citrix ADC is configured as SP, ACS URL will be `https://<SP-domain_name>/cgi/samlauth`. Citrix recommends having a full URL in the expression for matching.
   - SAML supports only RSA certificate. Other certificates like HSM, FIPS, and so on are not supported.
   - You must specify the starting of the domain with “^” sign (example: ^https) along with the dollar sign “$” at the end of the string (example: samlauth$).

   For more details on the command, see https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction and https://support.citrix.com/article/CTX316577.

2. Configure the SAML authentication policy and associate the SAML IdP profile as the action of the policy.

```
add authentication samlIdPPolicy samlIDPPol1 -rule true -action samlIDPProf1
```


3. Bind the policy to the authentication virtual server.

   ```
   bind authentication vserver saml-auth-vserver -policy samlIDPPol1 -
   priority 100
   ```

   For more details on the command, see https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlIdPProfile.

To configure a Citrix ADC appliance as a SAML IdP by using the GUI

1. Navigate to **Security>AAA-Policies>Authentication>Advanced Policies>SAML IdP**.
2. Select **Servers** tab, click **Add**, enter values for the following parameters, and click **Create**.

   Parameter description:

   Assertion Consumer Service URL - URL that the authenticated user will be redirected to.

   IdP Certificate Name - Certificate-Key pair used for the authentication page.

   SP Certificate Name - Certificate of the Service Provider in this scenario, the key is not required for this.

   Sign Assertion - The option to sign the assertion and the response when redirecting the client back to the Service Provider.

   Issuer Name - Identifier. Unique ID that is specified on both the SP and IdP to help identify the Service Provider to each other.

   Service Provider ID - Unique ID that will be specified on both the SP and IdP to help identify the Service Provider to each other. This can be anything and does not need to be the URL as specified below, but needs to be the same on both the SP and IdP profiles.

   Reject Unsigned Requests - Option you can specify to ensure only assertions signed with the SP Certificate are accepted.

   Signature Algorithm - Algorithm used to sign and verify the assertions between the IdP and SP, this needs to be the same on both the IdP and SP profiles.

   Digest Method - Algorithm used to verify the integrity of the Assertions between the IdP and SP, this needs to be the same on both the IdP and SP profiles.

   SAML Binding - Same as described in the SP profile, it needs to be the same on both the SP and IdP.

3. Associate the SAML IdP policy with an authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the SAML IdP policy with the authentication virtual server.

Configure SAML single sign-on

November 8, 2021

To provide single sign-on capabilities across applications that are hosted on the service provider, you can configure SAML single sign-on on the SAML SP.

Configuring SAML single sign-on by using the command line interface

1. Configure the SAML SSO profile.

   Example

   In the following command, Example is the load balancing virtual server that has a web link from the SharePoint portal. Nssp.example.com is the Traffic Management virtual server that is load balancing the SharePoint server.

   ```
   add tm samlSSOProfile tm-saml-sso -samlSigningCertName nssp -
       assertionConsumerServiceURL "https://nssp2.example.com/cgi/
       samlauth" -relaystateRule "\\"https://nssp2.example.com/
       samlsso.html\\"" -sendPassword ON -samlIssuerName nssp.example
       .com
   <!--NeedCopy-->
   ```

2. Associate the SAML SSO profile with the traffic action.

   Example

   The following command enables SSO and binds the SAML SSO profile created above to a traffic action.

   ```
   add tm trafficAction html_act -SSO ON -samlSSOProfile tm-saml-sso
   <!--NeedCopy-->
   ```

3. Configure the traffic policy that specifies when the action must be executed.

   Example

   The following command associates the traffic action with a traffic policy.

   ```
   add tm trafficPolicy html_pol "HTTP.REQ.URL.CONTAINS(\\"abc.html\\
       ")" html_act
   <!--NeedCopy-->
   ```

4. Bind the traffic policy created previously to a traffic management virtual server (load balancing or content switching). Alternatively, the traffic policy can be associated globally.

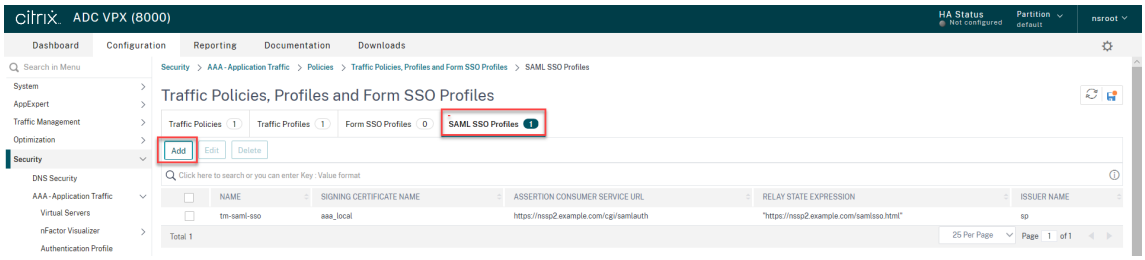
> **Note**
>
> This traffic management virtual server must be associated with the relevant authentication virtual server that is associated with the SAML action.

```
bind lb vserver lb1_ssl -policyName html_pol -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

Configuring SAML single sign-on by using the GUI

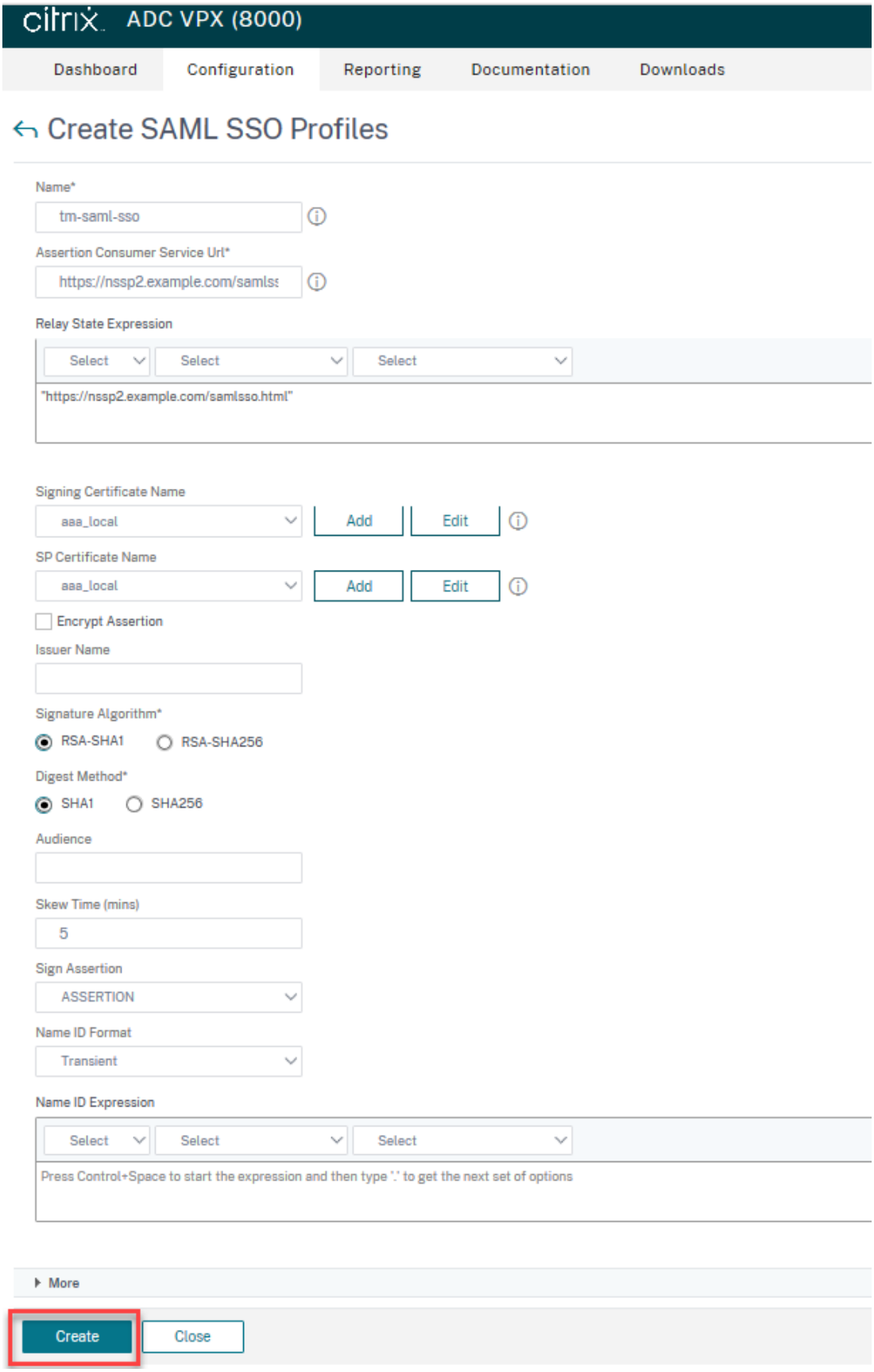
To configure SAML single sign-on you need to define the SAML SSO profile, the traffic profile, and the traffic policy and bind the traffic policy to a traffic management virtual server or globally to the Citrix ADC appliance.

1. Navigate to **Security > AAA-Application Traffic > Policies > Traffic > SAML SSO Profiles** and click **Add**.

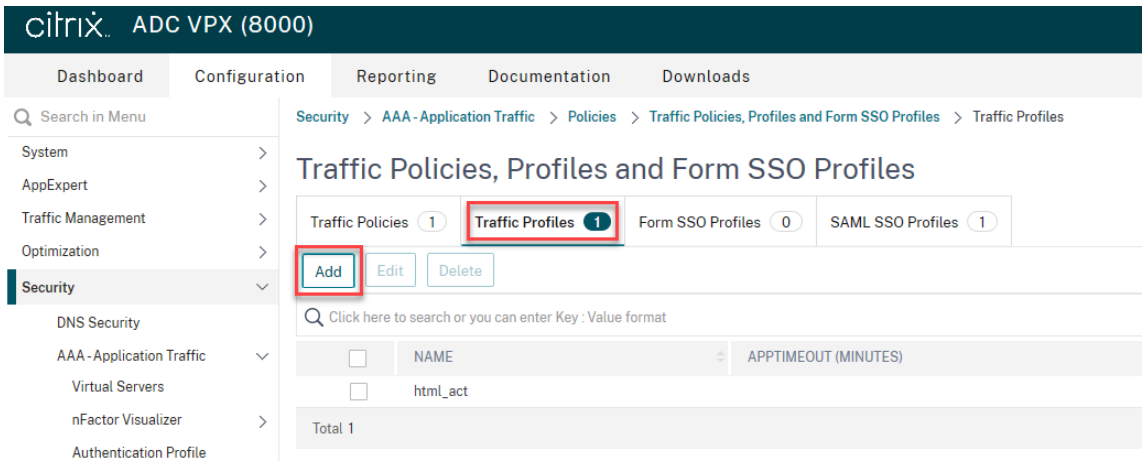


2. On the **Create SAML SSO Profiles** page, enter values for the following fields and click **Create**.
   - Name - Name for the SAML SSO Profile
   - Assertion Consumer Service Url - URL to which the assertion is to be sent
   - Signing Certificate Name - Name of the SSL certificate that is used to Sign Assertion
   - SP Certificate Name - Name of the SSL certificate of a peer/receiving party using which Assertion is encrypted
   - Issuer Name - The name to be used in requests sent from Citrix ADC to IdP to uniquely identify Citrix ADC
   - Signature Algorithm - Algorithm to be used to sign/verify SAML transactions
   - Digest Method - Audience for which assertion sent by IdP is applicable. This is typically entity name or url that represents a ServiceProvider
   - Audience - Audience for which an assertion sent by IdP is applicable. This is typically an entity name or url that represents a ServiceProvider
   - Skew Time (mins) - The number of minutes on either side of current time that the assertion would be valid
   - Sign Assertion - Option to sign portions of assertion when Citrix ADC IDP sends one. Based on the user selection, either Assertion or Response or Both or none can be signed.

- Name ID Format - Format of Name Identifier sent in Assertion
- Name ID Expression - Expression that will be evaluated to obtain NameIdentifier to be sent in assertion

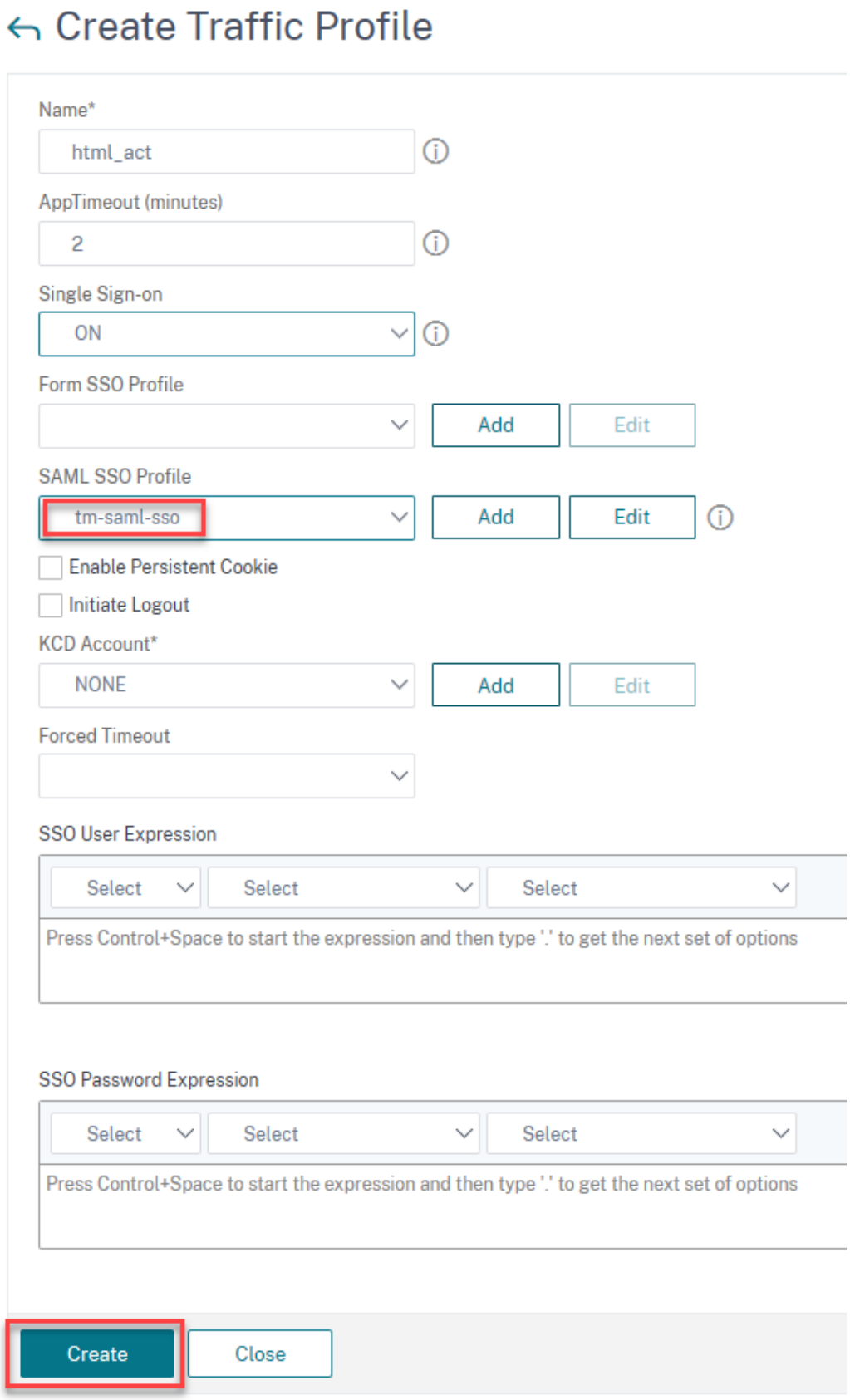


3. Navigate to **Security > AAA-Application Traffic > Policies>Traffic > Traffic Profiles** and click **Add**.

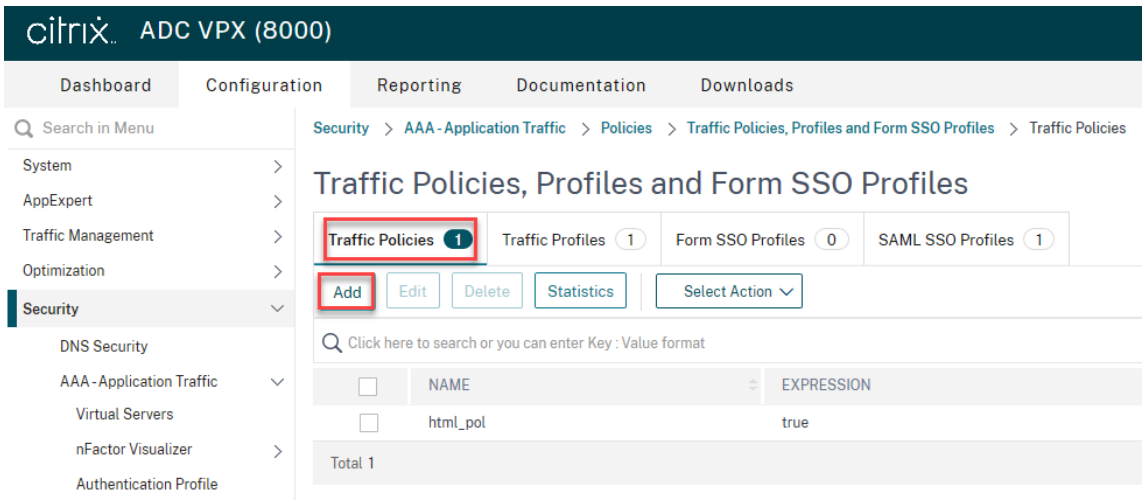


4. On the **Create Traffic Profile** page, enter values for the following fields, and click **Create**.

   - Name - Name for the traffic action.
   - AppTimeout (minutes) - Time interval, in minutes, of user inactivity after which the connection is closed.
   - Single Sign-on - Select ON
   - SAML SSO Profile - Select the created SAML SSSO Profile
   - KCD Account - Kerberos constrained delegation account name
   - SSO User Expression - Expression that will be evaluated to obtain user name for Single-SignOn
   - SSO Password Expression - Expression that will be evaluated to obtain password for SingleSignOn

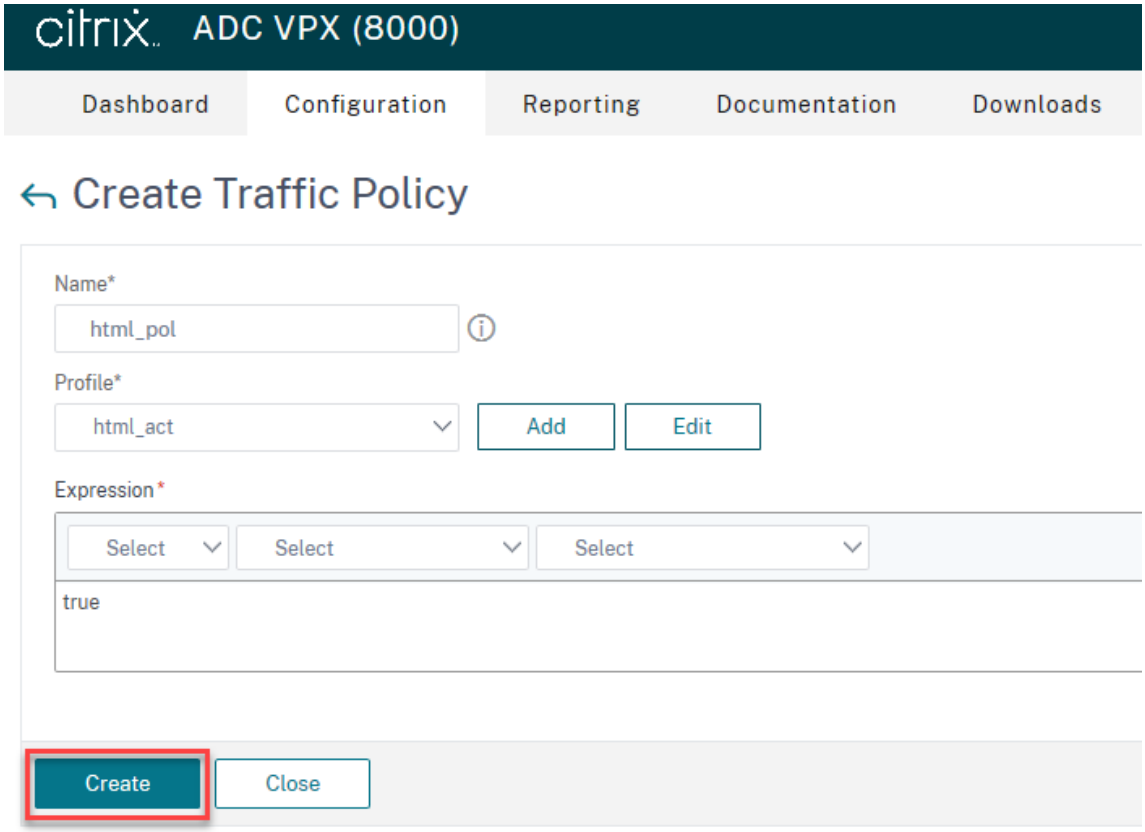


5. Navigate to **Security > AAA-Application Traffic > Policies > Traffic > Traffic Policies** and click **Add**.

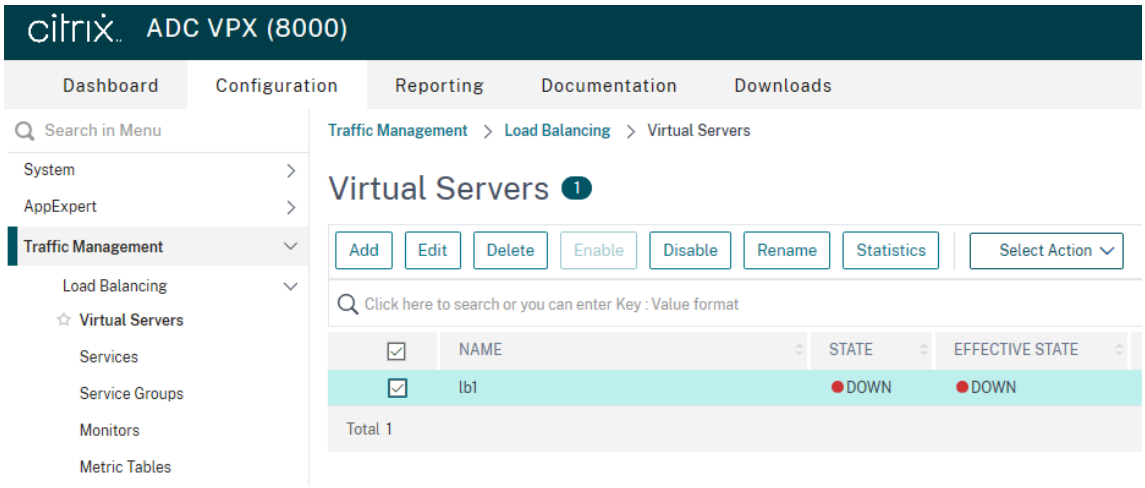


6. On the **Create Traffic policy** page, enter values for the following, and click **Create**.

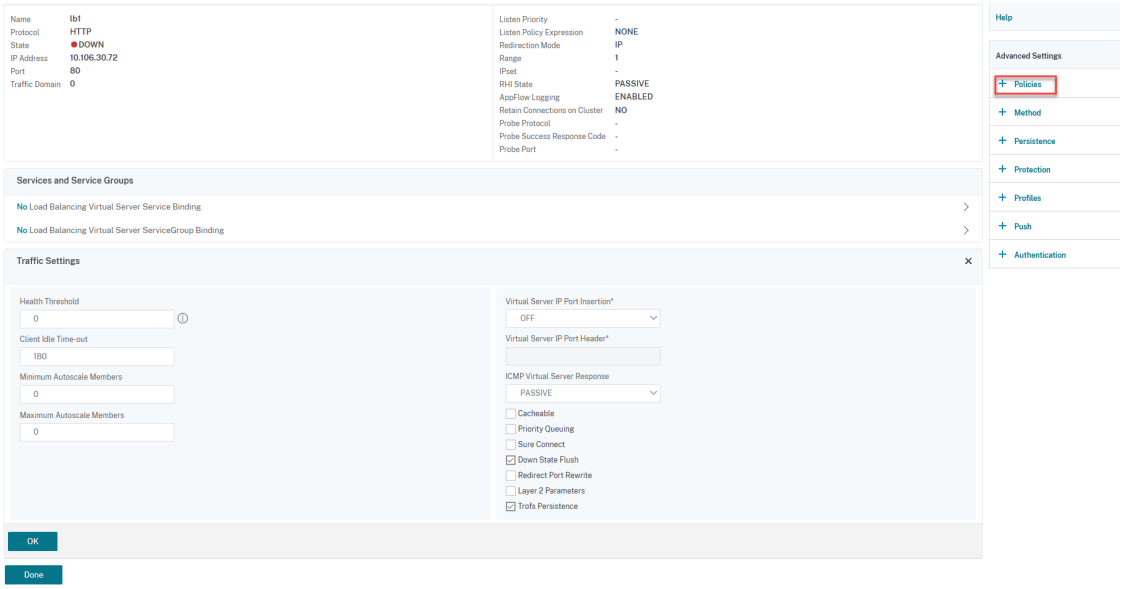
   - Name – Name of the traffic policy to be created
   - Profile – Select the created Traffic profile
   - Expression – Default syntax expression that the policy uses to respond to specific request. For example, true.



7. To bind the traffic policy to a traffic management virtual server, select a virtual server.



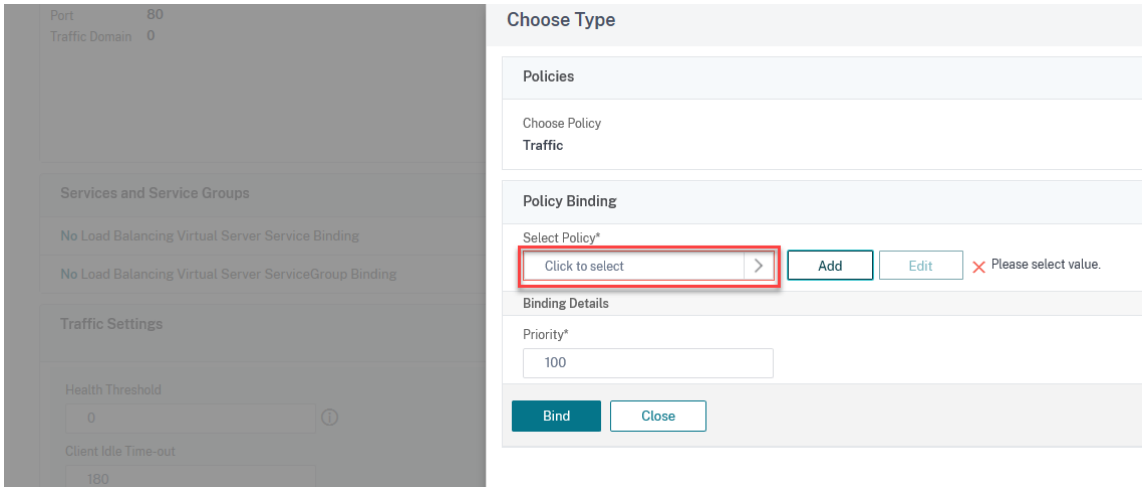
8. Click **Policies**.



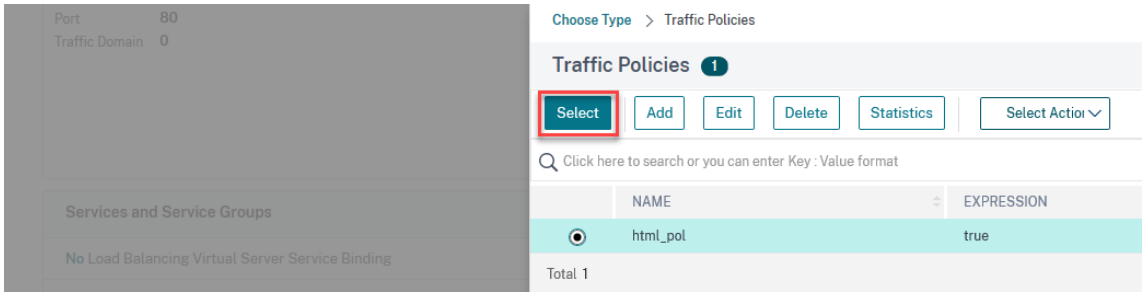
9. Select **Traffic** in the **Choose Policy** field and select **Request** in the **Choose Type** field, and click **Continue**.

   ![Click to add policy(/en-us/citrix-adc/media/saml-9.png)

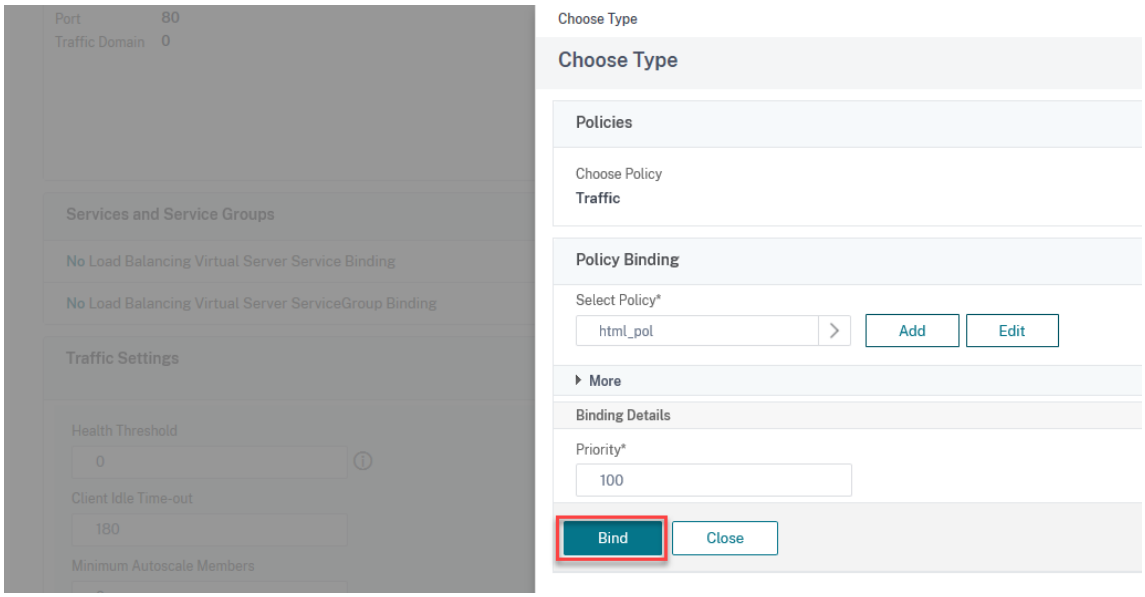
10. Under **Select Policy** field, click to select the created traffic .



11. Click **Select**.



12. Click **Bind** to bind the traffic policy to the virtual server.



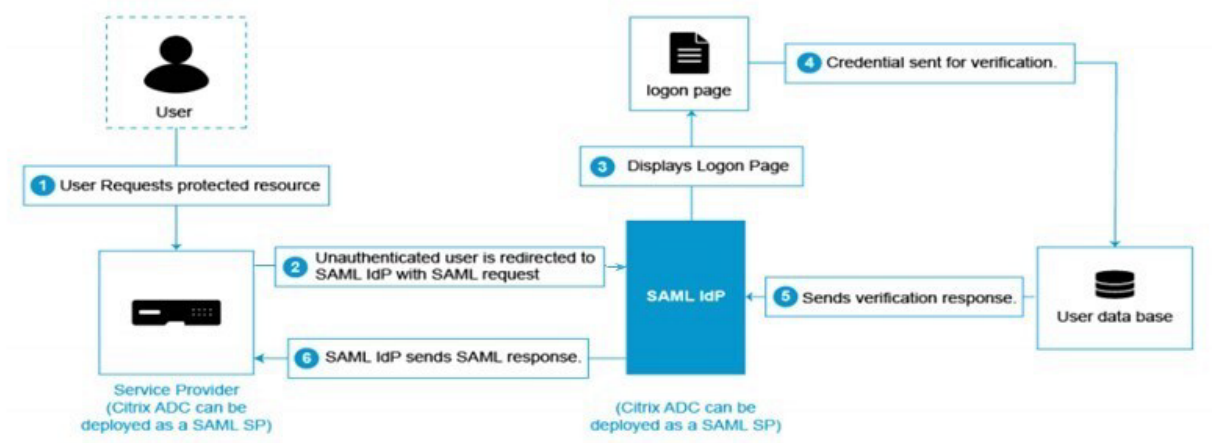
Configure Azure AD as SAML IdP and Citrix ADC as SAML SP

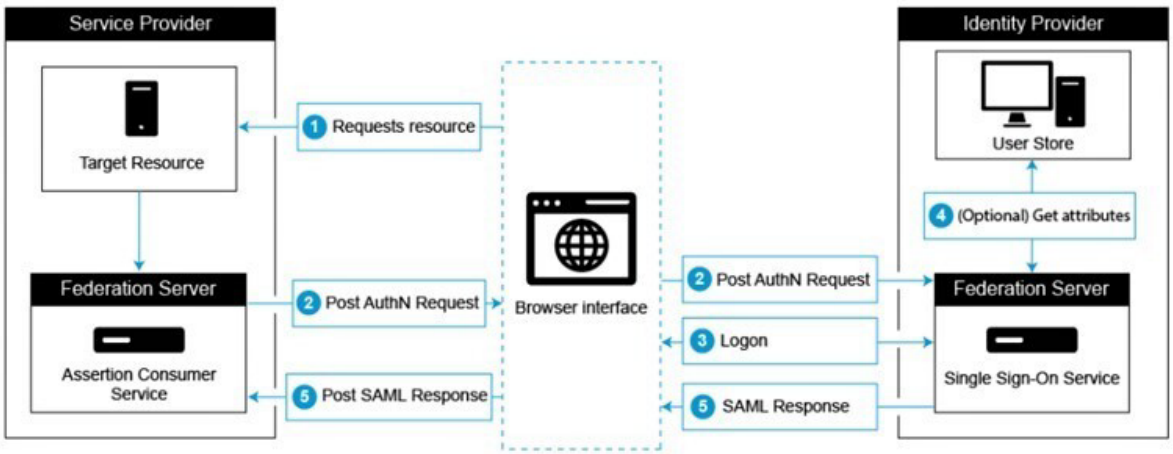
September 14, 2021

The SAML Service Provider (SP) is a SAML entity that is deployed by the service provider. When a user tries to access a protected application, the SP evaluates the client request. If the client is unauthenticated (does not have a valid NSC_TMAA or NSC_TMAS cookie), the SP redirects the request to the SAML Identity Provider (IdP). The SP also validates SAML assertions that are received from the IdP.

The SAML IdP (Identity Provider) is a SAML entity that is deployed on the customer network. The IdP receives requests from the SAML SP and redirects users to a logon page, where they must enter their credentials. The IdP authenticates these credentials with the user directory (external authentication server, such as LDAP) and then generates a SAML assertion that is sent to the SP. The SP validates the token, and the user is then granted access to the requested protected application.

The following diagram depicts the SAML authentication mechanism.

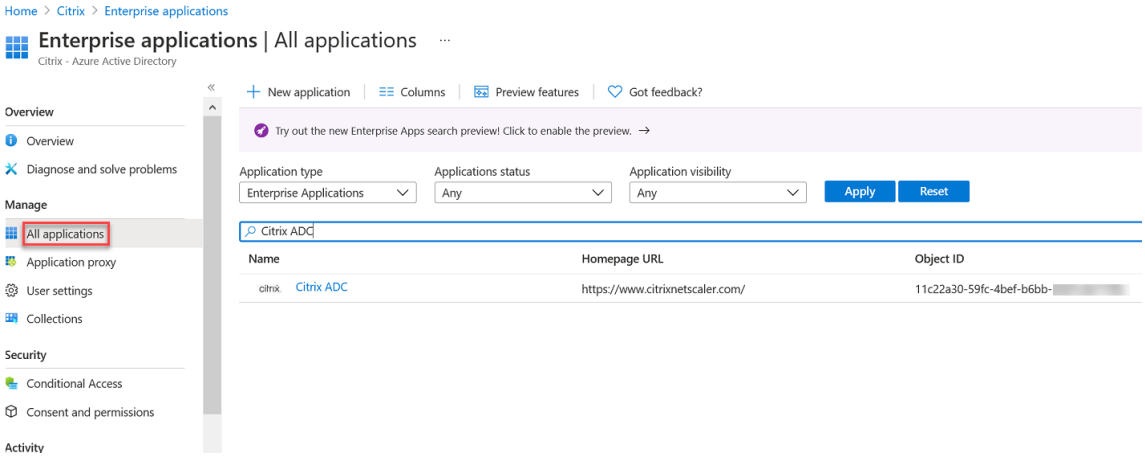




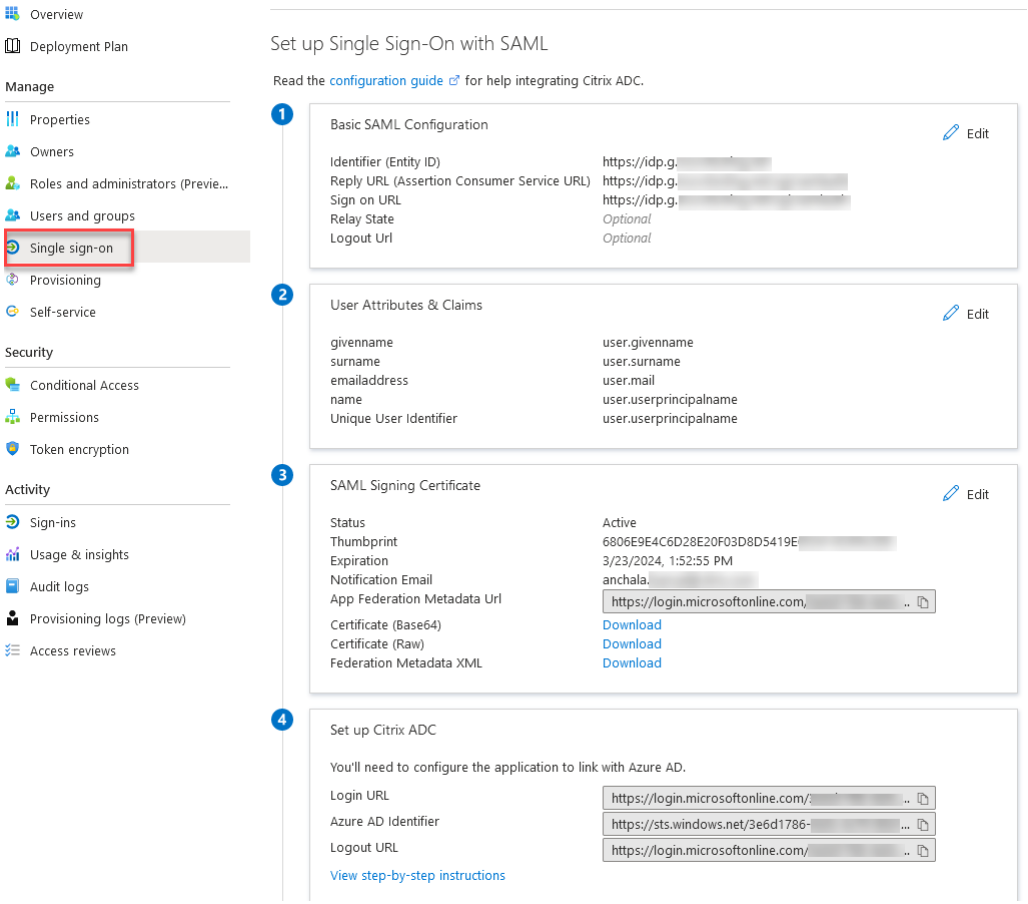
Azure AD side configurations

Configure single sign-on settings:

1. On the Azure portal, click **Azure Active Directory**.
2. Under **Manage** section in the navigation pane, click **Enterprise Applications**. A random sample of the applications in your Azure AD tenant appears.
3. In the search bar, enter Citrix ADC.



4. Under the **Manage** section, select **Single sign-on**.
5. Select **SAML** to configure single sign-on. The **Set up Single Sign-On with SAML - Preview** page appears. Here, Azure is acting as a SAML IdP.
6. Download certificate (Base64) present under **SAML Signing Certificate** to be used as samlidPCertName while configuring Citrix ADC as SAML SP.



7. **Configure basic SAML options:**

   Identifier (Entity ID) - Required for some apps. Uniquely identifies the application for which single sign-on is being configured. Azure AD sends the identifier to the application as the audience parameter of the SAML token. The application is expected to validate it. This value also appears as the Entity ID in any SAML metadata provided by the application.

   Reply URL - Mandatory. Specifies where the application expects to receive the SAML token. The reply URL is also referred as the Assertion Consumer Service (ACS) URL.

   Sign-on URL - When a user opens this URL, the service provider redirects to Azure AD to authenticate and sign on the user.

   Relay State - Specifies to the application where to redirect the user after the authentication is complete.

Citrix ADC side configurations

1. Navigate to **Security>AAA-Policies>Authentication>Basic Policies>SAML**.
2. Select **Servers** tab, click **Add**, enter values for the following parameters, and click **Create**.

Parameter description:

The value for parameters in bold needs to be taken from the Azure side configurations.

Name - Name of the server

Redirect URL - Enter the login URL used previously in the Azure AD “Setup Citrix ADC” section. `https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2`

Single Logout URL - `https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2`

SAML Binding - POST

Logout Binding - REDIRECT

IDP Certificate Name - IdPCert Certificate (Base64) present under SAML Signing Certificate.

User Field - userprincipalName. Taken from “User Attributes and Claims” section of Azure IdP.

Signing Certificate Name - Not needed for Azure AD. Select the SAML SP certificate (with private key) that Citrix ADC uses to sign authentication requests to the IdP. The same certificate (without private key) must be imported to the IdP, so that the IdP can verify the authentication request signature. This field is not needed by most IdPs.

IssuerName - Identifier. `https://idp.g.nssvctesting.net`

Reject unsigned assertion - ON

Audience - Audience for which assertion sent by IdP is applicable. This is typically entity name or URL that represents ServiceProvider.

Signature Algorithm - RSA-SHA256

Digest Method - SHA256

Default Authentication Group - The default group that is chosen when the authentication succeeds in addition to extracted groups.

Group Name Field - Name of the tag in assertion that contains user groups.

Skew Time (mins) - This option specifies the allowed clock skew in number of minutes that Citrix ADC ServiceProvider allows on an incoming assertion.

Two factor - OFF

Requested Authentication Context - exact

Authentication Class Type - None

Send Thumbprint - OFF

Enforce Username - ON

Force Authentication - OFF

Store SAML Response - OFF

Similarily, create a corresponding SAML policy and bind it to the authentication virtual server.

Note: Azure AD does not expect the Subject ID field in the SAML request. For Citrix ADC to not send the Subject ID field, type the following command on the Citrix ADC command prompt.

```
nsapimgr_wr.sh -ys call="ns_saml_dont_send_subject"
```

Additional features supported for SAML

September 14, 2021

The following features are suported for SAML.

Metadata reading and generation support for SAML SP and IdP configuration

Citrix ADC appliance now supports metadata files as means of configuration entities for both SAML Service Provider (SP) and Identity Provider (IdP). The metadata file is a structured XML file that describes the configuration of an entity. The metadata files for SP and IdP are separate. Based on deployment, and at times, one SP or IdP entity can have multiple metadata files.
As an administrator, you can export and import (SAML SP and IdP) metadata files on Citrix ADC.
The functionality of metadata export and import for SAML SP and IdP are explained in the following sections.

Metadata export for SAML SP

Consider an example where Citrix ADC is configured as SAML SP and an SAML IdP would like to import metadata that contains Citrix ADC SP configuration. Assume that Citrix ADC appliance is already configured with a “samlAction” attribute that specifies SAML SP configuration.

To export metadata from users or administrator, query Citrix Gateway or authentication virtual server as shown below:

```
1 https://vserver.company.com/metadata/samlsp/<action-name>
```

Metadata import for SAML SP

Currently, SAML Action configuration on Citrix ADC appliance takes various parameters. Administrator manually specifies these. However, administrators are often unaware of nomenclature if it comes

to interop with different SAML systems. If metadata of IdP is available, then bulk of the configuration in the 'samlAction' entity can be avoided. In fact, the entire IdP specific configuration might be omitted if IdP metadata file is given. The 'samlAction' entity now takes an additional parameter to read configuration from metadata file.

When you import a metadata in a Citrix ADC appliance, the metadata does not contain any signature algorithms to be used, it contains the endpoint details. A metadata can be signed with certain algorithms which can be used to verify the metadata itself. The algorithms are not stored in the 'samlAction' entity.

Therefore, what you specify in the 'samlAction' entity are the ones used when sending the data out. An incoming data can contain a different algorithm for a Citrix ADC appliance to process.

To fetch the metadata files by using command line interface.

```
set samlAction <name> [-metadataUrl <url> [-metadataRefreshInterval <
    int>] https://idp.citrix.com/samlidp/metadata.xml
```

> **Note**
>
> The metadataRefreshInterval parameter is the interval in minutes for fetching metadata information from the specified metadata URL. Default value 36000.

Metadata import for SAML IdP

The "samlIdPProfile" parameter takes a new argument to read the entire configuration that is specific to SP. SAML IdP configuration can be simplified by replacing SP specific properties with an SP metadata file. This file is queried through HTTP.

To read from metadata file using command line interface:

```
set samlIdPProfile <name> [-metadataUrl <url>] [-
    metadataRefreshInterval <int>]
```

Name-value attribute support for SAML authentication

You can now configure SAML authentication attributes with a unique name along with values. The names are configured in the SAML action parameter and the values are obtained by querying for the names. By specifying the name attribute value, admins can easily search for the attribute value associated with the attribute name. Also, admins no longer have to remember the attribute by its value alone.

> **Important**
>
> - In samlAction command, you can configure a maximum of 64 attributes separated by comma with total size less than 2048 bytes.
> - Citrix recommends that you use the attributes list. Use of “attribute 1 to attribute 16” will cause session failure if the extracted attribute size is large.

To configure the name-value attributes by using the CLI

At the command prompt, type:

```
add authentication samlAction <name> [-Attributes <string>]
```

Example:

```
add authentication samlAction samlAct1 -attributes  “mail,sn,
   userprincipalName”
```

Assertion Consumer Service URL support for SAML IdP

A Citrix ADC appliance configured as a SAML Identity Provider (IdP) now supports Assertion Consumer Service (ACS) indexing to process SAML Service Provider (SP) request. The SAML IdP imports ACS indexing configuration from SP metadata or allows for entering ACS indexes information manually. The following table lists some articles that are specific to deployments where the Citrix ADC appliance is used as a SAML SP or a SAML IdP.

The following table lists some articles that are specific to deployments where the Citrix ADC appliance is used as a SAML SP or a SAML IdP.

SAML SP	SAML IdP	Information Link
Citrix ADC	Microsoft Azure AD	Citrix Support
Okta	Citrix ADC	Citrix Support
AWS	Citrix ADC	Citrix Support

Some information on other specific deployments:

- NetScaler as SAML SP on FIPS Device
- Configuring Office365 for Single Sign-on with NetScaler as SAML IdP

WebView credential type support for authentication mechanisms

The authentication of a Citrix ADC appliance can now support AUTHv3 protocol. The WebView credential type in AUTHv3 protocol support all type of authentication mechanisms (including SAML and OAuth). The WebView credential type is a part of AUTHv3, which is implemented by Citrix Receiver and browser in web applications.

The following example explains the flow of WebView events through Citrix Gateway and Citrix Receiver:

1. The Citrix Receiver negotiates to Citrix Gateway for AUTHv3 protocol support.
2. Citrix ADC appliance responds positively and suggests a specific start URL.
3. Citrix Receiver then connects to the specific endpoint (URL).
4. The Citrix Gateway sends a response to the client to start the WebView.
5. Citrix Receiver starts WebView and sends initial request to Citrix ADC appliance.
6. Citrix ADC appliance redirects URI to browser login endpoint.
7. Once authentication is complete, Citrix ADC appliance sends completion response to WebView.
8. The WebView now exits and gives control back to Citrix Receiver to continue AUTHv3 protocol for session establishment.

Increase of SessionIndex size in SAML SP

The SessionIndex size of the SAML Service Provider (SP) is increased to 96 bytes. Previously, the default maximum size of SessionIndex was 63 bytes.

> **Note**
>
> Support introduced in NetScaler 13.0 Build 36.x

Custom authentication class reference support for SAML SP

You can configure custom authentication class reference attribute in the SAML action command. Using the custom authentication class reference attribute, you can customize the class names in the appropriate SAML tags. The custom authentication class reference attribute along with namespace is sent to the SAML IdP as part of SAML SP authentication request.

Previously, using SAML action command, you could configure only a set of predefined classes defined in authnCtxClassRef attribute.

> **Important**
>
> While configuring customAuthnCtxClassRef attribute, ensure the following:

- The names of the classes must include alphanumeric characters or a valid URL with proper XML tags.

- If you have to configure multiple custom classes, each class must be separated by commas

To configure the customAuthnCtxClassRef attributes by using the CLI

At the command prompt, type:

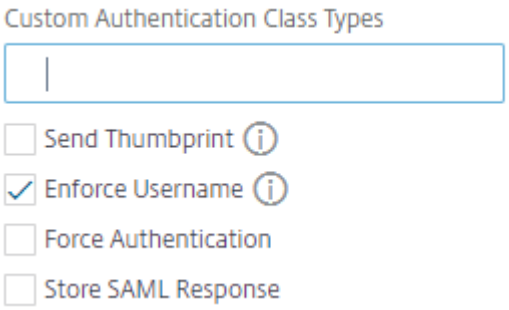
- add authentication samlAction <name> [-customAuthnCtxClassRef <string>]
- set authentication samlAction <name> [-customAuthnCtxClassRef <string>]

Example:

- `add authentication samlAction samlact1 -customAuthnCtxClassRef http://www.class1.com/LoA1,http://www.class2.com/LoA2`
- `set authentication samlAction samlact2 -customAuthnCtxClassRef http://www.class3.com/LoA1,http://www.class4.com/LoA2`

To configure the customAuthnCtxClassRef attributes by using the GUI

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Actions > SAML**.
2. On the SAML page, select **Servers** tab and Click **Add**.
3. On the **Create Authentication SAML Server** page, enter the name for SAML action.
4. Scroll down to configure the class types in **Custom Authentication Class Types** section.



Support for artifact binding in SAML IdP

Citrix ADC appliance configured as SAML Identity Provider (IdP) supports artifact binding. The artifact binding enhances the security of SAML IdP and restricts the malicious users from inspecting the assertion.

Assertion Consumer Service URL support for SAML IdP

A Citrix ADC appliance configured as a SAML Identity Provider (IdP) now supports Assertion Consumer Service (ACS) indexing to process SAML Service Provider (SP) request. The SAML IdP imports ACS indexing configuration from SP metadata or allows for entering ACS indexes information manually.

FIPS offload support

A Citrix ADC MPX FIPS appliance used as a SAML service provider now supports encrypted assertions. Also, a Citrix ADC MPX FIPS appliance functioning as a SAML service provider or a SAML identity provider can now be configured to use the SHA2 algorithms on FIPS hardware.

> **Note**
>
> In FIPS mode, only RSA-V1_5 algorithm is supported as key transport algorithm.

Configuring FIPS offload support using the command line interface:

1. Add SSL FIPS

   add ssl fipsKey fips-key

2. Create a CSR and use it at CA server to generate a certificate. You can then copy the certificate in **/nsconfig/ssl**. Let's assume that the file is *fips3cert.cer*.

   ```
   add ssl certKey fips-cert -cert fips3cert.cer -fipsKey fips-key<!--
   NeedCopy-->
   ```

3. Specify this certificate in the SAML action for SAML SP module

   ```
   set samlAction <name> -samlSigningCertName fips-cert<!--NeedCopy-->
   ```

4. Use the certificate in samlIdpProfile for SAML IDP module

   ```
   set samlidpprofile fipstest -samlIdpCertName fips-cert<!--NeedCopy-->
   ```

Common SAML terminologies

The following are some common SAML terminologies:

- **Assertion:** A SAML assertion is an XML document returned by the Identity Provider to the Service Provider after authentication of the user. The assertion has a very specific structure, as defined by the SAML standard.
- **Types of Assertions:** The following are the types of assertion.
  - Authentication - the user is authenticated by a particular means at a particular time
  - Authorization - the user was granted or denied access to a specified resource
  - Attributes - the user is associated with the supplied attributes

- **Assertion Consumer Service (ACS):** The service provider's endpoint (URL) that is responsible for receiving and parsing a SAML assertion
- **Audience Restriction:** A value within the SAML assertion that specifies who (and only who) the assertion is intended for. The "audience" will be the service provider and is typically a URL but can technically be formatted as any string of data.
- **Identity Provider (IdP):** In terms of SAML, the Identity Provider is the entity that verifies the identity of the user, in response to a request by the Service Provider.

  The Identity Provider is responsible for maintaining and authenticating the user's identity
- **Service Provider (SP):** In terms of SAML, the Service Provider (SP) offers a service to the user and allows the user to sign in by using SAML. When the user attempts to sign in, the SP sends a SAML authentication request to the Identity Provider (IdP)
- **SAML Binding:** SAML requestors and responders communicate by exchanging messages. The mechanism to transport these messages is called a SAML binding.
- **HTTP Artifact:** One of the binding options supported by the SAML protocol. HTTP Artifact is useful in scenarios where the SAML requester and responder are using an HTTP User-Agent and do not want to transmit the entire message, either for technical or security reasons. Instead, a SAML Artifact is sent, which is a unique ID for the full information. The IdP can then use the Artifact to retrieve the full information. The artifact issuer must maintain state while the artifact is pending. An Artifact Resolution Service (ARS) must be set up.

  HTTP Artifact sends the artifact as a query parameter.
- **HTTP POST:** One of the binding options supported by the SAML protocol.

  HTTP POST sends the message content as a POST parameter, in the payload.
- **HTTP Redirect:** One of the binding options supported by the SAML protocol.

  When HTTP Redirect is used, the Service Provider redirects the user to the Identity Provider where the login happens, and the Identity Provider redirects the user back to the Service Provider. HTTP Redirect requires intervention by the User-Agent (the browser).

  HTTP Redirect sends the message content in the URL. Because of this, it cannot be used for the SAML response, because the size of the response will typically exceed the URL length allowed by most browsers.

  Note: The Citrix ADC appliance supports POST and Redirect bindings during logout.
- Metadata: Metadata is the configuration data in SP and IDP to know how to communicate to each other which will be in XML standards

Other useful Citrix articles related to SAML authentication

You might find the following articles related to SAML authentication useful.

- https://support.citrix.com/article/CTX277558
- https://support.citrix.com/article/CTX259127
- https://support.citrix.com/article/CTX228135
- https://support.citrix.com/article/CTX221631
- https://support.citrix.com/article/CTX138988

OAuth authentication

September 14, 2021

The authentication, authorization, and auditing traffic management feature supports OAuth and OpenID Connect (OIDC) authentication. It authorizes and authenticates users to services that are hosted on applications such as Google, Facebook, and Twitter.

Points to note

- Citrix ADC Advanced Edition and higher is required for the solution to work.
- A Citrix ADC appliance must be on version 12.1 or later for the appliance to work as an OAuth IdP using OIDC.
- OAuth on a Citrix ADC appliance is qualified for all SAML IdPs that are compliant with "OpenID connect 2.0".

A Citrix ADC appliance can be configured to behave as a Service Provider (SP) or an Identity Provider (IdP), using SAML and OIDC. Previously, a Citrix ADC appliance configured as IdP supported only SAML protocol. Starting from Citrix ADC 12.1 version, Citrix ADC supports the OIDC as well.

OIDC is an extension to OAuth authorization/delegation. A Citrix ADC appliance supports OAuth and OIDC protocols in the same class of other authentication mechanisms. OIDC is an add-on to OAuth as it provides a way for getting user information from the authorization server as opposed to OAuth that gets only a token which cannot be gleaned for user information.

The authentication mechanism facilitates the inline verification of OpenID tokens. A Citrix ADC appliance can be configured to obtain certificates and verify signatures on the token.

A major advantage of using the OAuth and OIDC mechanisms is that the user information is not sent to the hosted applications. Therefore, the risk of identity theft is considerably reduced.

The Citrix ADC appliance configured for authentication, authorization, and auditing now accepts incoming tokens that are signed using the HMAC HS256 algorithm. In addition, the public keys of the SAML Identity Provider (IdP) are read from a file, instead of learning from a URL endpoint.

In the Citrix ADC implementation, the application is accessed by the authentication, authorization, and auditing traffic management virtual server. So, to configure OAuth, you must configure an OAuth policy which must then be associated with an authentication, authorization, and auditing traffic management virtual server.

Configure the OpenID Connect protocol

A Citrix ADC appliance can now be configured as an identity provider by using OIDC protocol. OIDC protocol strengthens the identity providing capabilities of the Citrix ADC appliance. You can now access the enterprise wide hosted application with a single sign-on. The OIDC offers more security by not transferring user password but works with tokens with specific lifetime. OIDC also is designed to integrate with non-browser clients such as apps and services. Therefore, many implementations adopt OIDC widely.

Advantages of having the OpenID Connect support

- OIDC eliminates the overhead of maintaining multiple authentication passwords as the user has a single identity across the organization.
- OIDC provides a robust security for your password as the password is shared only with your identity provider and not with any application you access.
- OIDC has vast interoperability with various systems making it easier for the hosted applications to accept OpenID.
- OIDC is a simple protocol that enables native clients to easily integrate with servers.

To configure a Citrix ADC appliance as an IdP using the OpenID Connect protocol by using GUI

1. Navigate to **Configuration > Security > AAA-Application Traffic > Policies > Authentication > Advanced Policies > OAuth IdP**.
2. Click **Profile** and click **Add.**

   On the **Create Authentication OAuth IDP Profile** screen, set values for the following parameters and click **Create**.

   - **Name** – Name of the authentication profile.
   - **Client ID** – Unique string that identifies SP.
   - **Client Secret** – Unique secret that identifies SP.
   - **Redirect URL** – Endpoint on SP to which code/token has to be posted.
   - **Issuer Name** – String that identifies IdP.
   - **Audience** – Target recipient for the token being sent by IdP. This might be checked by the recipient.
   - **Skew Time** – The time for which the token remains valid.

- **Default Authentication Group** – A group added to the session for this profile to simplify policy evaluation and help in customizing policies.

3. Click **Policies** and click **Add.**
4. On the **Create Authentication OAuth IDP Policy** screen, set values for the following parameters and click **Create**.

   - **Name –** The name of the authentication policy.
   - **Action** – Name of profile created earlier.
   - **Log Action** –Name of message log action to use when a request matches this policy. Not a mandatory filed.
   - **Undefined-Result Action** – Action to perform if the result of policy evaluation is undefined(UNDEF). Not a mandatory field.
   - **Expression** – Default syntax expression that the policy uses to respond to specific request. For example, true.
   - **Comments** – Any comments about the policy.

Binding the OAuthIdP policy and LDAP policy to the authentication virtual server

1. Navigate to **Configuration > Security > AAA-Application Traffic > Policies >Authentication > Advanced Policies > Actions > LDAP**.
2. On **LDAP Actions** screen, click **Add**.
3. On **Create Authentication LDAP Server** screen, set the values for the following parameters, and click **Create**.

   - **Name –** The name of the LDAP action
   - **ServerName/ServerIP –** Provide FQDN or IP of the LDAP server
   - Choose appropriate values **for Security Type, Port, Server Type, Time-Out**
   - Make sure **Authentication** is checked
   - **Base DN –** Base from which to start LDAP search. For example, dc=aaa,dc=local.
   - **Administrator Bind DN:** User name of the bind to LDAP server. For example, admin@aaa.local.
   - **Administrator Password/Confirm Password: Password to bind LDAP**
   - Click **Test Connection** to test your settings.
   - **Server Logon Name Attribute:** Choose **"sAMAccountName"**
   - Other fields are not mandatory and hence can be configured as required.

4. Navigate to **Configuration > Security > AAA-Application Traffic > Policies >Authentication > Advanced Policies > Policy.**
5. On **Authentication Policies** screen, click **Add**.

6. On **Create Authentication Policy** page, set the values for the following parameters, and click **Create.**
    - **Name –** Name of the LDAP Authentication Policy.
    - **Action Type –** Choose **LDAP.**
    - **Action –** Choose the LDAP action.
    - **Expression –** Default syntax expression that the policy uses to respond to specific request. For example, true**.

To configure the Citrix ADC appliance as an IdP using the OpenID Connect protocol by using CLI

At the command prompt, type the following commands:

- `add authentication OAuthIDPProfile <name> [-clientID <string>][-clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]<!--NeedCopy-->`
- `add authentication OAuthIdPPolicy <name> -rule <expression> [-action <string> [-undefAction <string>] [-comment <string>][-logAction <string >]<!--NeedCopy-->`
- `add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -ldapBase "dc=aaa,dc=local"<!--NeedCopy-->`
- `ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -ldapLoginName sAMAccountName<!--NeedCopy-->`
- `add authentication policy aaa-ldap-adv-pol -rule true -action aaa-ldap-act<!--NeedCopy-->`
- `bind authentication vserver auth_vs -policy <ldap_policy_name> -priority 100 -gotoPriorityExpression NEXT<!--NeedCopy-->`
- `bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -priority 5 -gotoPriorityExpression END<!--NeedCopy-->`
- `bind vpn global -certkey <><!--NeedCopy-->`

> **Note**
>
> You can bind more than one key. Public parts of certificates bound are sent in response to `jwks \\_uri query (https://gw/oauth/idp/certs).`

Citrix ADC as an OAuth SP

September 14, 2021

The authentication, authorization, and auditing traffic management feature supports OAuth authentication for authenticating users to applications that are hosted on applications such as Google, Facebook, and Twitter.

Points to note

- Citrix ADC Advanced Edition and higher is required for the solution to work.
- OAuth on Citrix ADC appliance is qualified for all SAML IdPs that are compliant with “OpenID connect 2.0”.

To configure OAuth by using the configuration utility

1. Configure the OAuth action and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, and create a policy with OAuth as the action type, and associate the required OAuth action with the policy.

2. Associate the OAuth policy with an authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the OAuth policy with the authentication virtual server.

> **Note**
>
> Attributes (1 to 16) can be extracted in the OAuth response. Currently these attributes are not evaluated. They are added for the future reference.

To configure OAuth by using the command line interface:**

1. Define an OAuth action.

```
1 add authentication OAuthAction <name> -authorizationEndpoint <URL>
      -tokenEndpoint <URL> [-idtokenDecryptEndpoint <URL>] -clientID
      <string> -clientSecret <string> [-defaultAuthenticationGroup <
     string>][-tenantID <string>][-GraphEndpoint <string>][-
     refreshInterval <positive_integer>] [-CertEndpoint <string>][-
     audience <string>][-userNameField <string>][-skewTime <mins>][-
     issuer <string>][-Attribute1 <string>][-Attribute2 <string>][-
     Attribute3 <string>]...
2 <!--NeedCopy-->
```

2. Associate the action with an advanced authentication policy.


```
1 add authentication Policy** <name> -rule <expression> -action <
     string>
2 <!--NeedCopy-->
```

Example:

```
add authentication oauthAction a -authorizationEndpoint https://example
.com/ -tokenEndpoint https://example.com/ -clientiD sadf -clientsecret
df
```

For more information on authentication OAuthAction parameters, see authentication OAuthAction.

> **Note**
>
> When a certEndpoint is specified, the Citrix ADC appliance polls that endpoint at the configured frequency to learn the keys.

To configure a Citrix ADC to read the local file and parse keys from that file, a new configuration option is introduced as follows:

```
1 set authentication OAuthAction <> -**CertFilePath** <path to local file
     with jwks>
2 <!--NeedCopy-->
```

OAuth feature now supports the following capabilities in the token API from the Relying Party (RP) side and from the IdP side of Citrix Gateway and Citrix ADC.

- PKCE (Proof Key for Code Exchange) support
- Support for client_assertion

Name-value attribute support for OAuth authentication

You can now configure OAuth authentication attributes with a unique name along with the values. The names are configured in the OAuth action parameter either as “Attributes” and the values are obtained by querying for the names. The extracted attributes are stored in authentication, authorization, and auditing session. Admins can query these attributes either using `http.req.user.attribute("attribute name")` or `http.req.user.attribute(1)`, based on the chosen method of specifying attribute names.

By specifying the name of the attribute, admins can easily search for the attribute value associated with that attribute name. Also, admins no longer have to remember the “attribute1 to attribute16” by its number alone.

> **Important**
>
> In a OAuth command, you can configure a maximum of 64 attributes separated by comma with a total size less than 1024 bytes.
>
> **Note**
>
> The session failure can be avoided if the total value size of “attribute 1 to attribute 16” and the values of attributes specified in “Attributes” are not more than 10 KB.

To configure the name-value attributes by using the CLI

At the command prompt, type:

- `add authentication OAuthAction <name> [-Attributes <string>]`
- `set authentication OAuthAction <name> [-Attributes <string>]`

Examples:

- `add authentication OAuthAction a1 -attributes "email,company"-attribute1 email`
- `set authentication OAuthAction oAuthAct1 -attributes "mail,sn,userprincipalName"`

Citrix ADC as an OAuth IdP

September 14, 2021

A Citrix ADC appliance can now be configured as an identity provider by using the OpenID-Connect (OIDC) protocol. OIDC protocol strengthens the identity providing capabilities of the Citrix ADC appliance. You can now access the enterprise wide hosted application with a single sign-on as OIDC offers more security by not transferring the user password but using tokens with specific lifetime. OpenID also is designed to integrate with non-browser clients such as apps and services. Therefore, the OIDC protocol is widely adopted by many implementations.

> **Note**
>
> Citrix ADC must be on version 12.1 or later for the appliance to work as an OAuth IdP using the OIDC protocol.

Advantages of having Citrix ADC as an OAuth IdP

- Eliminates the overhead of maintaining multiple authentication passwords as the user has a single identity across an organization.

- Provides a robust security for your password as the password is shared only with your identity provider and not with any application you access.
- Provided vast interoperability with various systems making it easier for the hosted applications to accept OpenID.

> **Note**
>
> Citrix ADC Advanced Edition and higher is required for the solution to work.

To configure the Citrix ADC appliance as an OAuth IdP using the GUI

1. Navigate to **Configuration > Security > AAA-Application Traffic > Policies > Authentication > Advanced Policies > OAuth IdP**.
2. Click **Profile** and click **Add.**

   On the **Create Authentication OAuth IDP Profile** screen, set values for the following parameters and click **Create**.

   - **Name** – Name of the authentication profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created.
   - **Client ID** – Unique string that identifies SP. Authorization server infers client configuration using this ID. Maximum Length: 127.
   - **Client Secret** – Secret string established by user and authorization server. Maximum Length: 239.
   - **Redirect URL** – Endpoint on SP to which code/token has to be posted.
   - **Issuer Name** – Identity of the server whose tokens are to be accepted. Maximum Length: 127.
   - **Audience** – Target recipient for the token being sent by IdP. This might be checked by the recipient.
   - **Skew Time** – This option specifies the allowed clock skew in number of minutes that Citrix ADC allows on an incoming token. For example, if skewTime is 10, then the token would be valid from (current time - 10) min to (current time + 10) min, that is 20 min in all. Default value: 5.
   - **Default Authentication Group** – A group added to session internal grouplist when this profile is chosen by IdP which can be used in nFactor flow. It can be used in expression (AAA.USER.IS_MEMBER_OF("xxx")) for authentication policies to identify relying party related nfactor flow. Maximum Length: 63

A group added to the session for this profile to simplify policy evaluation and help in customizing policies. This is the default group that is chosen when the authenticatiob succeeds in addition to the extracted groups. Maximum Length: 63.

3. Click **Policies** and click **Add.**
4. On the **Create Authentication OAuth IDP Policy** screen, set values for the following parameters and click **Create**.
    - **Name** – The name of the authentication policy.
    - **Action** – Name of profile created above.
    - **Log Action** – Name of messagelog action to use when a request matches this policy. Not a mandatory filed.
    - **Undefined-Result Action** – Action to perform if the result of policy evaluation is undenfined(UNDEF). Not a mandatory field.
    - **Expression** – Default syntax expression that the policy uses to respond to specific request. For example, true.
    - **Comments** – Any comments about the policy.

Binding the OAuthIDP policy and LDAP policy to the authentication virtual server

1. Navigate to **Configuration > Security > AAA-Application Traffic > Policies >Authentication > Advanced Policies > Actions > LDAP**.
2. On **LDAP Actions** screen, click **Add**.
3. On the **Create Authentication LDAP Server** screen, set the values for the following parameters, and click **Create**.
    - **Name –** The name of the ldap action
    - **ServerName/ServerIP –** Provide FQDN or IP of the LDAP server
    - Choose appropriate values **for Security Type, Port, Server Type, Time-Out**
    - Make sure **Authentication** is checked
    - **Base DN –** Base from which to start LDAP search. For example, dc=aaa,dc=local.
    - **Administrator Bind DN:** Username of the bind to LDAP server. For example, admin@aaa.local.
    - **Administrator Password/Confirm Password: Password to bind LDAP**
    - Click **Test Connection** to test your settings.
    - **Server Logon Name Attribute:** Choose **"sAMAccountName"**
    - Other fields are not mandatory and hence can be configured as required.
4. Navigate to **Configuration > Security > AAA-Application Traffic > Policies >Authentication > Advanced Policies > Policy.**
5. On the **Authentication Policies** screen, click **Add**.

6. On the **Create Authentication Policy** page, set the values for the following parameters, and click **Create.**

   - **Name –** Name of the LDAP Authentication Policy.
   - **Action Type –** Choose **LDAP.**
   - **Action –** Choose the LDAP action.
   - **Expression –** Default syntax expression that the policy uses to respond to specific request. For example, true**.

OAuth feature now supports the following capabilities in the token API from the Relying Party (RP) side and from the IdP side of Citrix Gateway and Citrix ADC.

- PKCE (Proof Key for Code Exchange) support
- Support for client_assertion

To configure the Citrix ADC appliance as an IdP using the OIDC protocol with the command line

At the command prompt, type the following commands:

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string> [-undefAction <string>] [-comment <string>][-logAction <
    string>]

add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -
    ldapBase "dc=aaa,dc=local"

ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName

add authentication policy aaa-ldap-adv-pol -rule true -action aaa-ldap-
    act

bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority 100 -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority 5 -gotoPriorityExpression END

bind vpn global  - certkey <>
<!--NeedCopy-->
```


> **Note**
>
> You can bind more than one key. Public parts of certificates bound are sent in response to `jwks \\_uri query (https://gw/oauth/idp/certs).`

Encrypted tokens support on OIDC protocol

Citrix ADC appliance with the OIDC mechanism now supports the sending of encrypted tokens along with signed tokens. The Citrix ADC appliance uses JSON web encryption specifications to compute the encrypted tokens and supports only compact serialization of encrypted tokens. To encrypt an OpenID token, a Citrix ADC appliance needs the public key of the relying party (RP). The public key is obtained dynamically by polling the relying party's well-known configuration endpoint.

A new "relyingPartyMetadataURL" option is introduced in the "authentication OAuthIDPProfile." profile.

To configure the relying party's endpoint by using CLI

At the command prompt, type:

"'set authentication OAuthIDPProfile [-relyingPartyMetadataURL] [-refreshInterval] [-status < >]

```
- **relyingPartyMetadataURL** - Endpoint at which Citrix ADC IdP can
  get details about the relying party being configured. Metadata
  response must include endpoints for jwks_uri for RP public keys.

- **refreshInterval** - Defines the rate at which this endpoint must
  be polled to update the certificates in minutes.

- **status** - Reflects the status of the polling operation. The
  status is complete once the Citrix ADC appliance successfully
  obtains the public keys.

   **Example**

   ```
   set authentication OAuthIDPProfile sample_profile -
      relyingPartyMetadataURL https://rp.customer.com/metadata -
      refreshInterval 50 -status < >
   <!--NeedCopy-->
```

After the endpoint is configured, a Citrix ADC appliance first polls the relying party's well-known endpoint to read configuration. Currently, the Citrix ADC appliance processes only the 'jwks_uri' endpoint.


```

- If the 'jwks_uri' is absent in the response, the status of the profile is not complete.
- If the 'jwks_uri' is present in the response, Citrix ADC polls that endpoint also to read the public keys of the relying party.

> **Note:** Only RSAES-OAEP and AES GCM encryption type algorithms are supported for token encryption.

### Custom attributes support on OpenID Connect

OpenID relying parties may require more than a user name or a user principal name (UPN) in the token to create the user profile or make authorization decisions. Most commonly, the user groups are required to apply authorization policies for the user. Sometimes, more details, such as the first or the last name is required for provisioning a user account.

Citrix ADC appliance configured as an IdP can be used to send extra attributes in the OIDCid_token using expressions. Advanced policy expressions are used to send the custom attributes as per the requirement. The Citrix IdP evaluates the expressions corresponding to the attributes and then computes the final token.

Citrix ADC appliance automatically JSONify the output data. For example, numbers (such as SSN) or boolean values (true or false) are not surrounded by quotes. Multi-valued attributes, such as groups are placed within an array marker ("[" and "]"). The complex type attributes are not automatically computed, and you can configure the PI expression of those complex values as per your requirement.

#### To configure the relying party's endpoint by using CLI

At the command prompt, type:

```
set oauthidpprofile <name> -attributes <AAA-custom-attribute-pattern>
<!--NeedCopy-->
```

The <AAA-custom-attribute-pattern> can be described as:

Attribute1=PI-Expression@@@attribute2=PI-Expression@@@

'attribute1','attribute2' are literal strings that represent the name of the attribute to be inserted in the id_token.

> **Note:** You can configure up to 2,000 bytes of attributes.

Example: `set oauthidpprofile sample_1 -attributes q{ myname=http.req.user.name@@@ssn="123456789"@@@jit="false"@@@groups=http.req.user.groups }`

- Preceding PI expression is an advanced policy expression that represents the value to be used against the attribute. The PI expression can be used to send a string literal, such as "'hardcoded

string"'. The string literal is surrounded by double quotes around single quotes or double quotes around a start and pattern (as stated earlier, the start pattern is "q{"). If the value of the attribute is not a string literal, the expression is evaluated at runtime and its value is sent in token. If the value at runtime is empty, the corresponding attribute is not added to the ID token.

- As defined in the example, "false" is a literal string for the attribute "jit". Also, "ssn" has hard-coded value for reference. Groups and "myname" are PI expressions that yield strings.

Support for active-active GSLB deployments on Citrix Gateway

Citrix Gateway configured as an Identity Provider (IdP) using the OIDC protocol can support active-active GSLB deployments. The active-active GSLB deployment on Citrix Gateway IdP provides the capability to load balance an incoming user login request across multiple geographic locations.

> **Important**
>
> Citrix recommends you to bind CA certificates to the SSL service and enable certificate validation on the SSL service for enhanced security.

For more information on configuring GSLB setup, see Example of a GSLB setup and configuration.

API authentication with the Citrix ADC appliance

September 14, 2021

There is a paradigm shift in the way modern applications interact with their clients. Traditionally, browser clients were used to access services. Applications usually set session cookies to track user context. Modern and distributed applications make it hard to maintain user sessions across microservices. Due to this, most of the application accesses have become API based.
Clients that communicate with these distributed services have also evolved. Most clients obtain tokens from a trusted entity called Authorization Server to prove user identity and access. These clients then present the token to the application with each access request. Therefore, traditional proxy devices like Citrix ADC need to evolve to support these clients. A Citrix ADC appliance provides a way for administrators to handle such traffic. Citrix ADC can be deployed as an API Gateway to front-end all the traffic that destined to the published services. An API Gateway can be deployed for traditional (Hybrid Multi Cloud or HMC) or Cloud native environments. The API Gateway terminates all the inbound traffic to offer several services such as authentication, authorization, rate limiting, routing, caching, SSL offload, application firewall, and so on. Therefore, it becomes a critical component in the infrastructure.

Token types

Tokens exchanged during the API access mostly conform to the OAuth/OpenID Connect (OIDC) protocol. Access tokens that are used only for 'delegated access' conform to the OAuth protocol, whereas ID Tokens that comply with OIDC carry user information as well.
Access tokens are normally an opaque or random blob of data. However, they can sometimes be singed tokens conforming to JWT (Json Web Token) standards. ID Tokens are always signed JWTs.

API Access with OAuth

OAuth authentication type on a Citrix ADC appliance can be used to handle both OAuth and OIDC protocols. OIDC is an extension to the OAuth protocol.

OAuthAction on a Citrix ADC appliance can be used to handle interactive clients such as browsers and native clients such as client apps. Interactive clients are redirected to Identity Provider for login using the OIDC protocol. Native clients can obtain tokens out of band and can present those tokens at a Citrix ADC appliance for access.

> **Note:**
>
> The access token obtained from endpoints can be cached for subsequent requests, thereby enhancing the API performance.

To configure token caching support by using the command line interface, type the following command at the command prompt:

```
set aaaparameter  - apITokenCache <ENABLED>

<!--NeedCopy-->
```

The following sections describe the API access method performed by native clients.

Virtual server for API Access

To deploy a Citrix ADC appliance for an API access, a Traffic Management (TM) virtual server is deployed with 401 Authentication. It is associated with an authentication (authentication, authorization, and auditing) virtual server to hold the authentication and session policies. Following configuration snippet creates one such virtual server.

```
Add lb vserver lb-api-access SSL <IP> 443 -authn401 On -AuthnVsName
    auth-api-access

Bind ssl vserver lb-api-access -certkeyName <ssl-cert-entity>

Add authentication vserver auth-api-access SSL
```



```
<!--NeedCopy-->
```

> **Note:**
>
> You would need to bind a service to the TM vserver, and an authentication policy (with OAuthAction described as follows) to the authentication virtual server to complete the configuration.

After creating the virtual server, one needs to add an OAuthAction along with corresponding policy. There are several other options within an OAuth action depending on the token type, and other security mechanisms.

OAuth Configuration for ID Tokens

ID Tokens are always signed JWTs. That is, they carry header, payload and signature. Since these are self-contained tokens, a Citrix ADC appliance can validate these tokens locally. To validate these tokens, the appliance would need to know the public key of the corresponding private key used to sign these tokens.

Following is an example of OAuthAction with certain mandatory arguments along with “certEndpoint”.

```
Add authentication OAuthAction oauth-api-access -clientid <your-client-id> -clientsecret <your-client-secret> -authorizationEndpoint <URL to which users would be redirected for login> -tokenEndpoint <endpoint at which tokens could be obtained> -certEndpoint <uri at which public keys of IdP are published>
<!--NeedCopy-->
```

Where,

- **Client ID** – Unique string that identifies SP. Authorization server infers client configuration using this ID. Maximum Length: 127.
- **Client Secret** – Secret string established by user and authorization server. Maximum Length: 239.
- **authorizationEndpoint** - URL at which users would normally log in (when using interactive clients).
- **tokenEndpoint** - URL on Authorization Server at which tokens/code are obtained/exchanged
- **certEndpoint** - URL at which Authorization Server publishes public keys used to sign the tokens. Authorization Server can publish more than one key and choose one of them to sign tokens.

> **Note:** Client ID/Client Secret/authorizationEndpoint/TokenEndpoint are optional parameters for API Access. However, it is a good practice to provide values for these parameters as the action entity can be reused for different purposes.

In the preceding configuration 'certEndpointpoint' is essential for ID Token validation. This endpoint contains public keys of the certificate used to sign the tokens. These public keys must correspond to JWKs (Json Web Keys) specification.

Once the certEndpoint is configured at the Citrix ADC appliance, it polls the endpoint periodically (with the default interval of 1 day that can be customizable in the configuration) to keep the public keys up to date. After the public keys are available, ADC can perform local validation of the incoming ID Tokens.

OAuth Configuration for opaque access tokens

Opaque tokens cannot be verified locally on the Citrix ADC appliance. These need to be validated on the Authorization server. A Citrix ADC appliance uses 'introspection protocol' mentioned in OAuth specifications in order to verify these tokens. A new option, introspectURL, is provided in OAuth configuration for verifying opaque tokens.

```
1 set oauthAction oauth-api-acccess -introspectURL <uri of the
     Authorization Server for introspection>
2
3 <!--NeedCopy-->
```

The format of the introspection API conforms to the specification at https://tools.ietf.org/html/rfc7662##section-2.1 as follows:

```
1 POST /introspect HTTP/1.1
2 Host: server.example.com
3 Accept: application/json
4 Content-Type: application/x-www-form-urlencoded
5 Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
6 token=mF_9.B5f-4.1JqM&token_type_hint=access_token
7
8 <!--NeedCopy-->
```

Binding policy to Authentication vserver

Once OAuthAction is created, corresponding policy needs to be created for invoking it.

```
1 add authentication policy oauth-api-access -rule <> -action <oauth-api-
     access><!--NeedCopy-->
```


bind authentication vserver auth-api-access -policy oauth-api-access -pri 100

```
## Additional security settings on a Citrix ADC appliance

Token validation includes token lifetime checks. Tokens outside of
    acceptable time are rejected. Following are the additional settings
    for extra security. Some of these are recommended to be configured
    always.

**Audience**: OAuth Action can be configured with an intended recipient
     of the token. All tokens are matched against this configured URL. A
     Citrix ADC appliance has an additional capability where the
    audience field actually points to a pattern set on the appliance.
    Using this pattern set, an administrator can configure more than one
     url for the audience.

<!--NeedCopy-->
```

add policy patset oauth_audiences

bind patset oauth_audiences https://app1.company.com

bind patset oauth_audiences https://app2.company.com

bind patset oauth_audiences httpsL//app1.company.com/path1

set oAuthAccess oauth-api-access -audience oauth_audiences

```
In the preceding example, more than one audience is specified in a
    pattern set. Therefore, an incoming token is allowed only if it
    contains any of the configured URLs in the pattern set.

**Issuer**: Identity of the server whose tokens are to be accepted.
    Maximum Length: 127. It is a good practice to configure the issuer
    of the tokens in OAuth action. This ensures that tokens issued by
    wrong Authorization Server are not allowed.

**SkewTime**: Specifies the allowed clock skew in number of minutes
    that a Citrix ADC appliance allows on an incoming token. For example
    , if skewTime is 10, then the token would be valid from (current
    time - 10) min to (current time + 10) min, that is 20 min in all.
    Default value: 5

**AllowedAlgorithms**: This option allows administrator to restrict
    certain algorithms in the incoming tokens. By default, all the
    supported methods are allowed. However, these can be controlled
    using this option.
```



```
8
9  The following configuration ensures only tokens that use RS256 and
       RS512 are allowed:
10
11 <!--NeedCopy-->
```

set oAuthAction oauth-api-access -allowedAlgorithms RS256 RS512

```
1  After above configuration, only tokens that use RS256 and RS512 are
       allowed.
2
3  ## Bypassing certain traffic from authentication
4
5  In many instances, there are some discovery APIs that are publicly
       accessible to the clients. These APIs typically reveal configuration
        and capabilities of the service itself. AN administrator can
       configure the Citrix ADC appliance to bypass authentication from
       these metadata URLs using 'No Authentication' policy described as
       follows:
6
7  <!--NeedCopy-->
```

add authentication policy auth-bypass-policy -rule <> -action NO_AUTHN

bind authentication vserver auth-api-access -policy auth-bypass-policy -pri 110

```
1  NO_AUTHN is an implicit action that results in authentication to be
       completed when the rule matches. There are other uses of NO_AUTHN
       action beyond the scope of API access.
2  <!--NeedCopy-->
```

LDAP authentication

September 14, 2021

As with other types of authentication policies, a Lightweight Directory Access Protocol (LDAP) authentication policy comprises an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. In addition to standard authentication functions, LDAP can search other active directory (AD) servers for user accounts for users that do not exist locally. This function is called referral support or referral chasing.

Normally you configure the Citrix ADC to use the IP address of the authentication server during authentication. With LDAP authentication servers, you can also configure the ADC to use the FQDN of the LDAP server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex authentication, authorization, and auditing configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you use the **serverName** parameter instead of the **serverIP** parameter, and substitute the server's FQDN for its IP address.

Before you decide whether to configure the ADC to use the IP or the FQDN of your LDAP server to authenticate users, consider that configuring authentication, authorization, and auditing to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

LDAP referral support is disabled by default and cannot be enabled globally. It must be explicitly enabled for each LDAP action. Make sure that the AD server accepts the same `binddn credentials` that are used with the referring (GC) server. To enable referral support, you configure an LDAP action to follow referrals, and specify the maximum number of referrals to follow.

If referral support is enabled, and the Citrix ADC receives an LDAP_REFERRAL response to a request, authentication, authorization, and auditing follows the referral to the active directory (AD) server contained in the referral and performs the update on that server. First, authentication, authorization, and auditing looks up the referral server in DNS, and connects to that server. If the referral policy requires SSL/TLS, it connects via SSL/TLS. It then binds to the new server with the `binddn credentials` that it used with the previous server, and performs the operation which generated the referral. This feature is transparent to the user.

The port numbers for LDAP connections are:

- 389 for unsecured LDAP connections (for plain text LDAP)
- 636 for secure LDAP connections (for SSL LDAP)
- 3268 for Microsoft unsecure LDAP connections (for plain text Global Catalog Server)
- 3269 for Microsoft secure LDAP connections (for SSL Global Catalog Server)

The following table contains examples of user attribute fields for LDAP servers:

LDAP server	User attribute	Case sensitive
Microsoft Active Directory Server	sAMAccountName	No
Novell eDirectory	ou	Yes

LDAP server	User attribute	Case sensitive
IBM Directory Server	uid	Yes
Lotus Domino	CN	Yes
Sun ONE directory (formerly iPlanet)	uid or cn	Yes

This table contains examples of the base DN:

LDAP server	Base DN
Microsoft Active Directory Server	DC=citrix,DC=local
Novell eDirectory	ou=users,ou=dev
IBM Directory Server	cn=users
Lotus Domino	OU=City,O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	ou=People,dc=citrix,dc=com

The following table contains examples of bind DN:

LDAP server	Bind DN
Microsoft Active Directory Server	CN=Administrator, CN=Users, DC=citrix, DC=local
Novell eDirectory	cn=admin, o=citrix
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	uid=admin,ou=Administrators, ou=TopologyManagement,o=NetscapeRoot

For more information about setting up authentication policies in general, see Authentication Policies. For more information about Citrix ADC expressions, which are used in the policy rule, see Policies and Expressions.

To create LDAP authentication server by using the command line interface

At the command prompt, type the following commands:

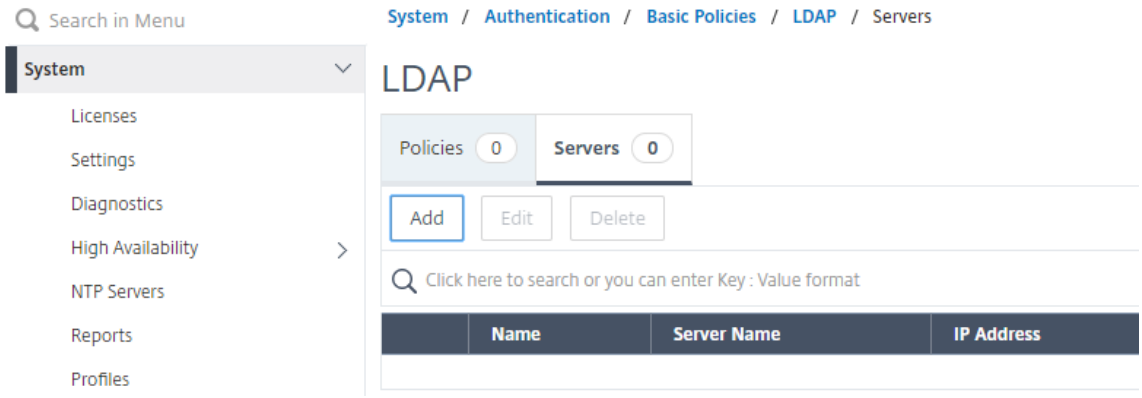
```
add authentication ldapAction <name> {
 -serverIP }
  <ip_addr|ipv6_addr|> | {
 -serverName <string> }
  }
```

Example

```
add authentication ldapAction ldap_server -serverip 1.1.1.1 -serverName
    ldap_test
```

To create LDAP authentication server by using the configuration utility

1. Navigate to **System > Authentication > Basic Policies > LDAP > Servers > Add**.



2. On the **Create Authentication LDAP Server** page, configure the parameters for the LDAP server.
3. Click **Create**.

To enable an authentication policy by using the command line interface

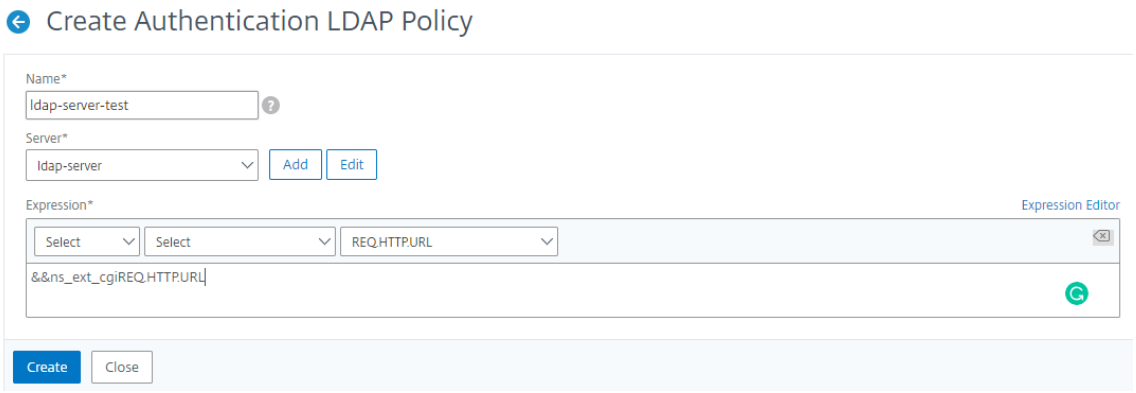
```
add authentication ldappolicy <name> <rule> [<reqAction>]
```

Example:

```
add authentication ldappolicy ldap-service-policy ns_true ldap_Server
```


To create LDAP authentication policy by using the configuration utility

1. Navigate to **System > Authentication > Basic Policies > LDAP > Policies > Add**
2. On the **Create Authentication LDAP Policy** page, configure the parameters for LDAP policy.



3. Click **Create**.

> **Note**
>
> You can configure LDAP servers/policies through **Security** tab. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Basic Policies > LDAP > Servers / Policies**.

To enable LDAP referral support by using the command line interface

At the command prompt, type the following commands:

- `set authentication ldapAction <name> -followReferrals ON<!--NeedCopy-->`
- `set authentication ldapAction <name> -maxLDAPReferrals <integer><!--NeedCopy-->`

Example

```
1 > set authentication ldapAction ldapAction-1 -followReferrals ON
2 > set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
```

Key-based authentication support for LDAP users

With key-based authentication, you can now fetch the list of public keys that are stored on the user object in LDAP server through SSH. The Citrix ADC appliance during the role-based authentication (RBA) process must extract public SSH keys from the LDAP server. The retrieved public key, which is compatible with SSH, must allow you to log in through RBA method.

A new attribute “sshPublicKey” is introduced in the “add authentication ldapAction” and “set authentication ldapAction” commands. By using this attribute, you can obtain the following benefits:

- Can store the retrieved public key, and the LDAP action uses this attribute to retrieve SSH key information from LDAP server.
- Can extract attribute names of up to 24 KB.

> Note
>
> The external authentication server, such as LDAP is used only to retrieve SSH key information. It is not used for authentication purpose.

Following is an example of the flow of events through SSH:

- SSH daemon sends an AAA_AUTHENTICATE request with password field empty to authentication, authorization, and auditing daemon port.
- If LDAP is configured to store the SSH public key, authentication, authorization, and auditing responds with “sshPublicKey” attribute along with other attributes.
- SSH daemon verifies these keys with the client keys.
- SSH daemon passes user name in the request payload, and authentication, authorization, and auditing returns the keys specific to this user along with generic keys.

To configure the sshPublicKey attribute, at the command prompt type the following commands:

- With add operation, you can add “sshPublicKey” attribute while configuring ldapAction command.

```
add authentication ldapAction <name> ( -serverIP <ip_addr|ipv6_addr
|*> | ( -serverName <string> ) ) [-serverPort <port>] … [-Attribute1 <
string>] … [-Attribute16 <string>][-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

- With set operation, you can configure “sshPublicKey” attribute to an already added ldapAction command.

```
set authentication ldapAction <name> [-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

Name-value attribute support for LDAP authentication

You can now configure the attributes of LDAP authentication with a unique name along with values. The names are configured in the LDAP action parameter and the values are obtained by querying for the name. By using this feature, a Citrix ADC appliance administrator can now achieve the following benefits:

- Minimizes the effort for administrators by remembering the attribute by name (not just by value)
- Enhances the search to query the attribute value associated with a name
- Provides an option to extract multiple attributes

To configure this feature at the Citrix ADC appliance command prompt, type:

"'add authentication ldapAction [-Attribute1]

```
Example

```add authentication ldapAction ldapAct1 attribute1 mail<!--NeedCopy
   -->
```

Support for validating end-to-end LDAP authentication

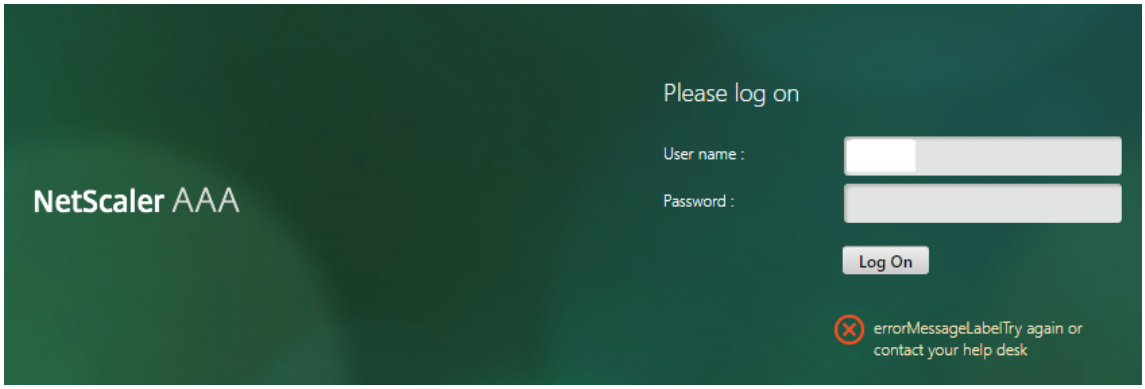
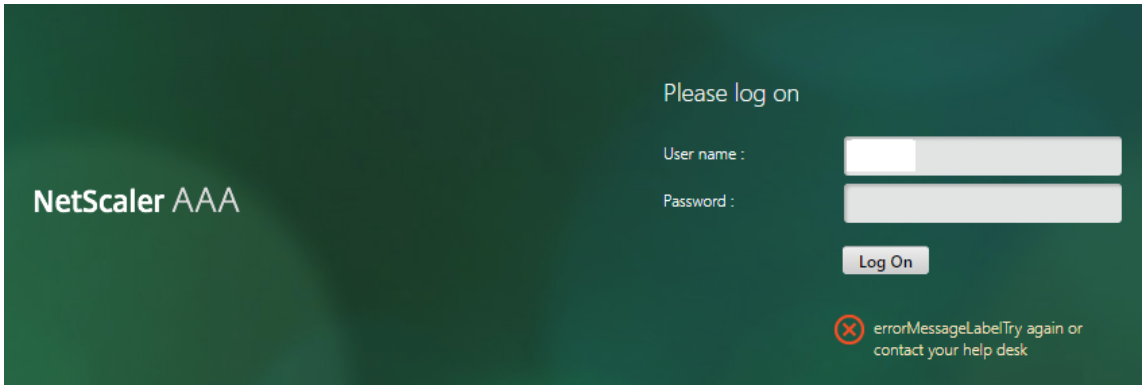
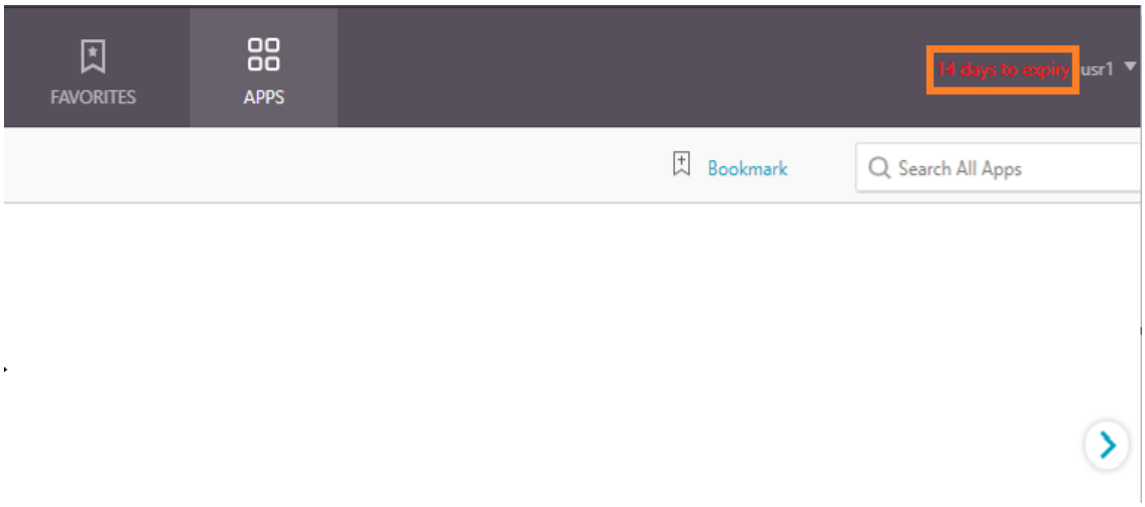
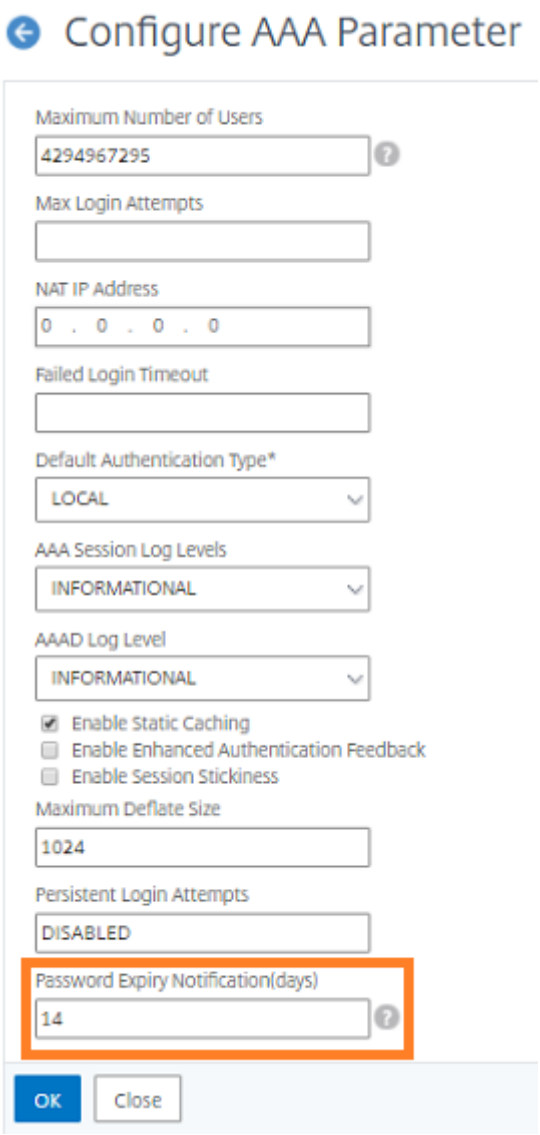
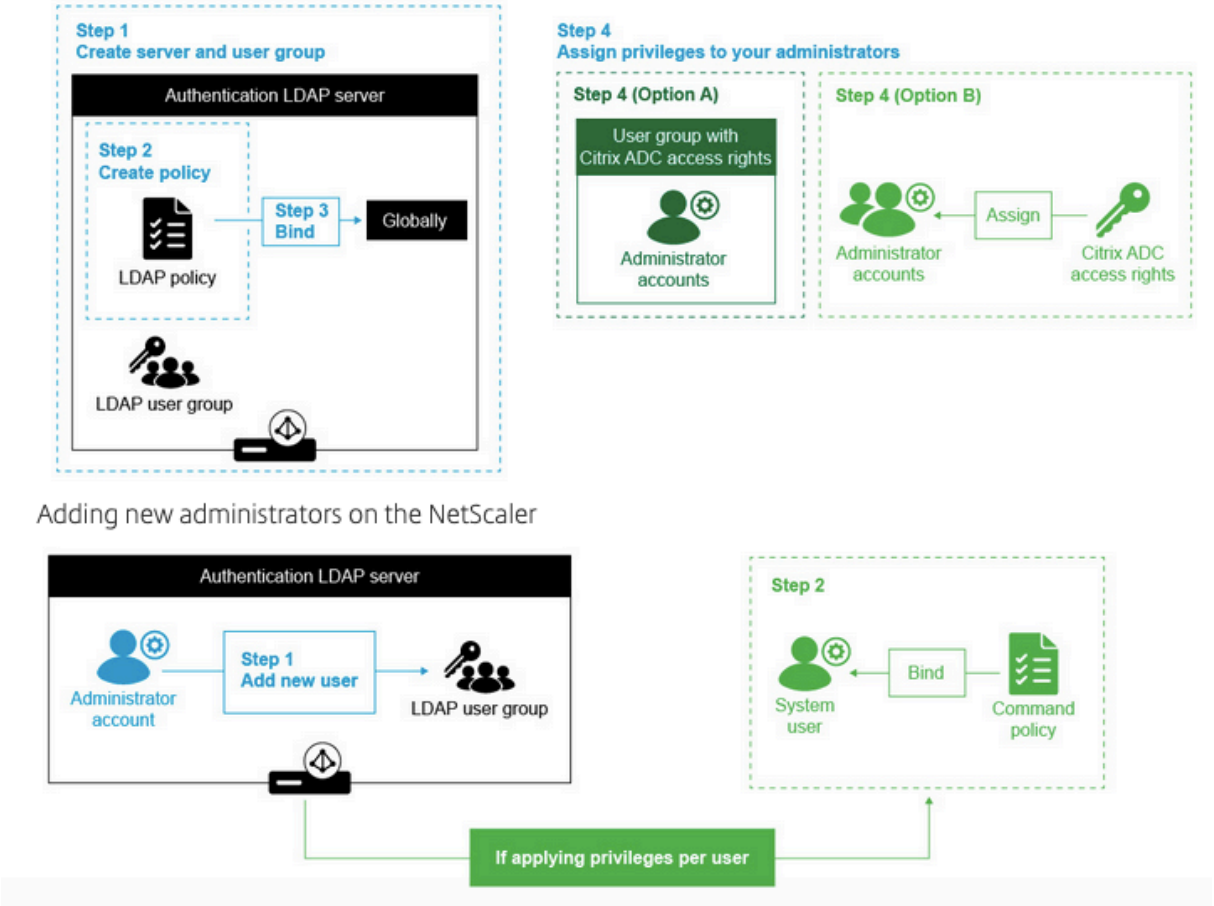
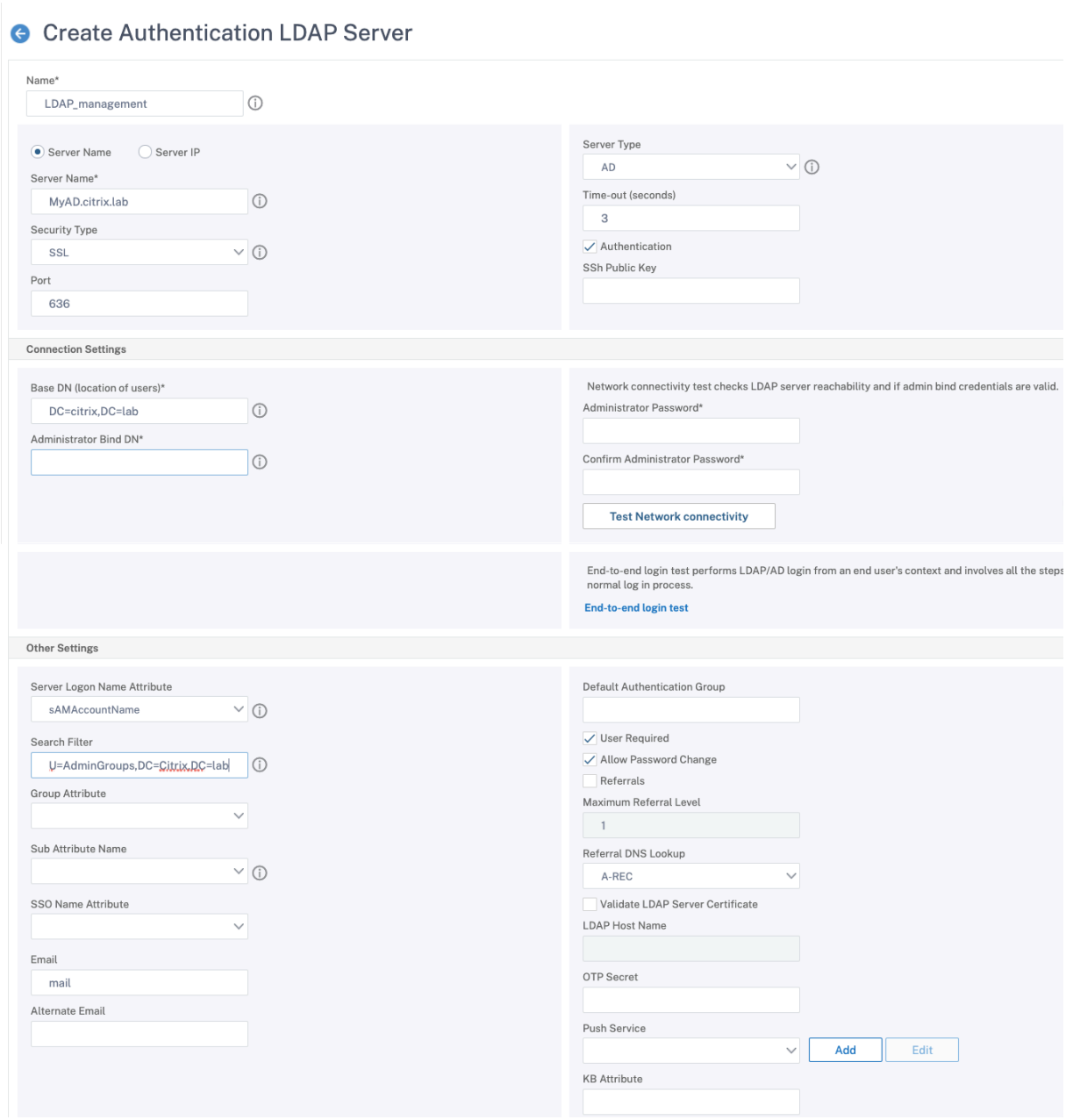
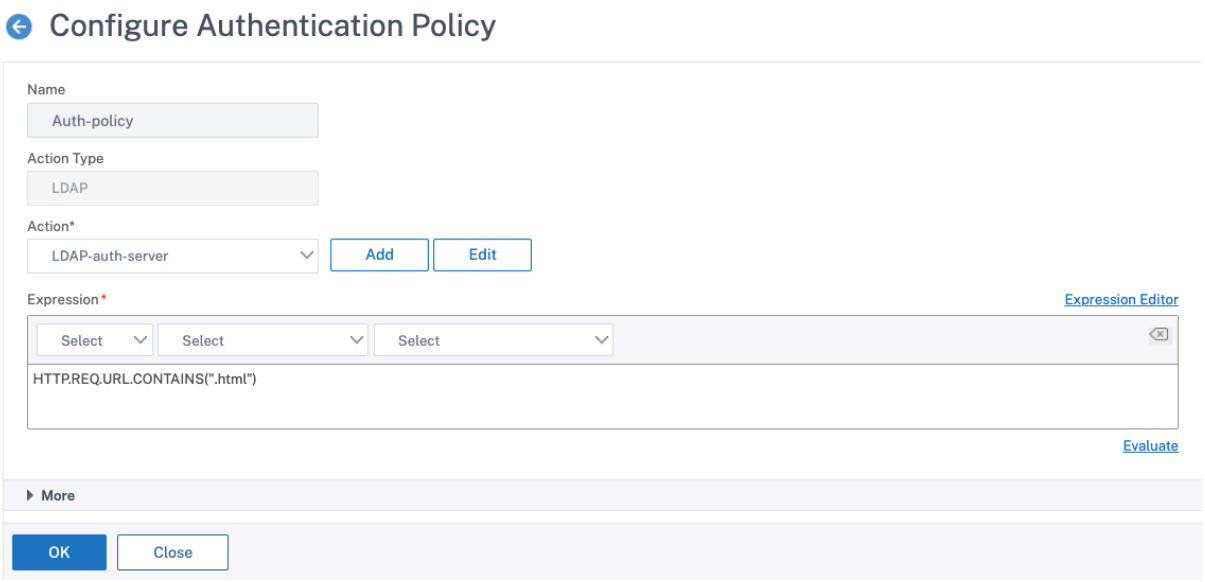
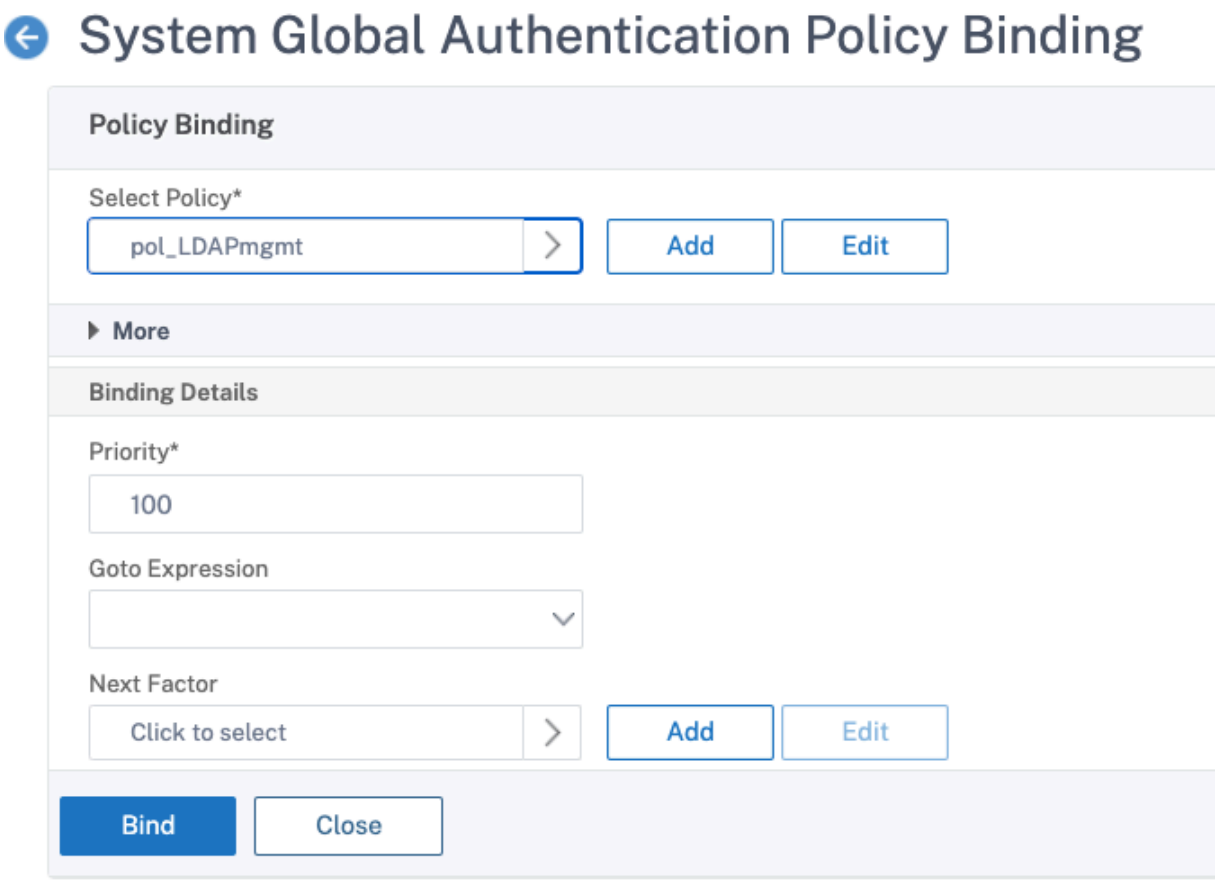
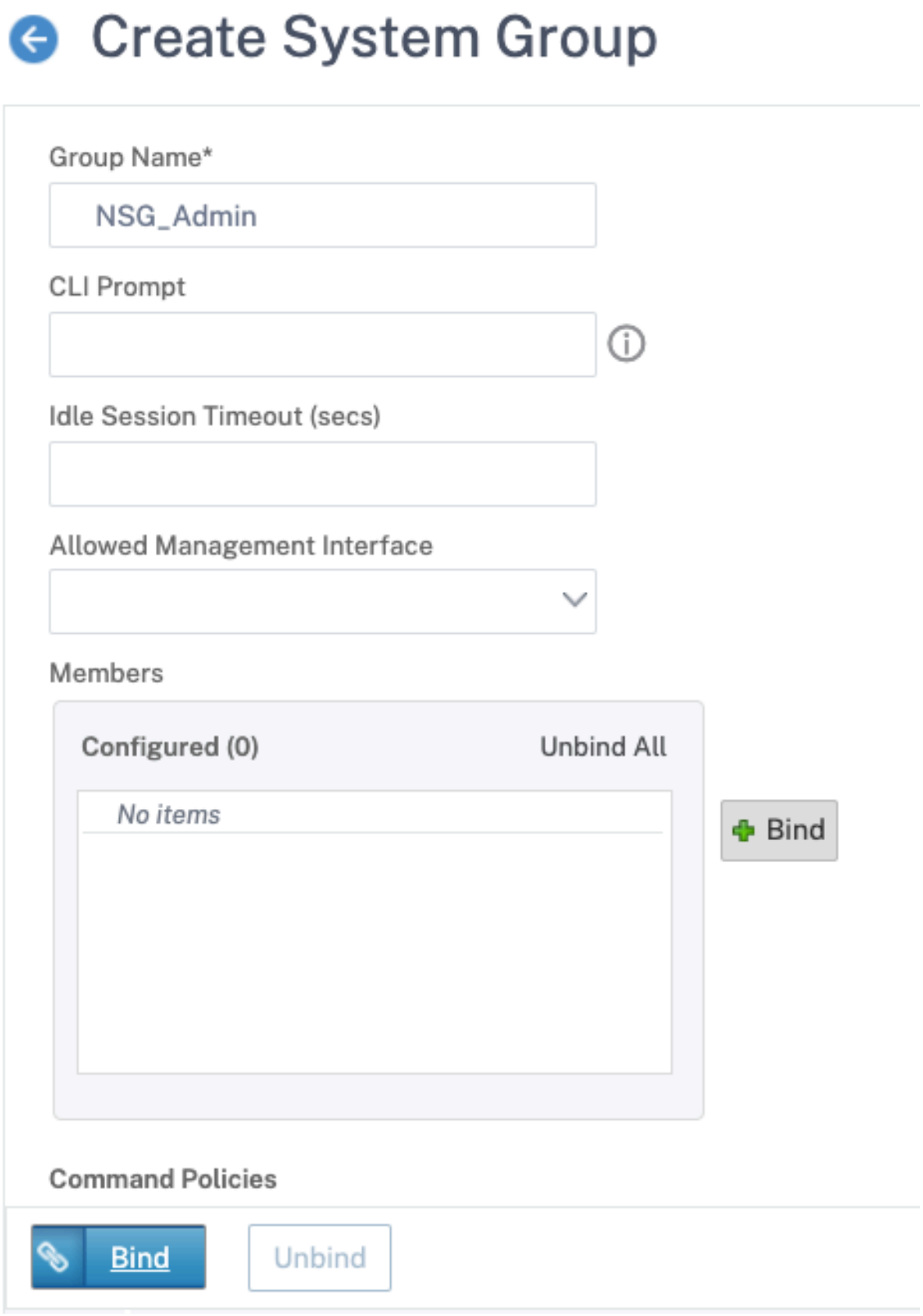
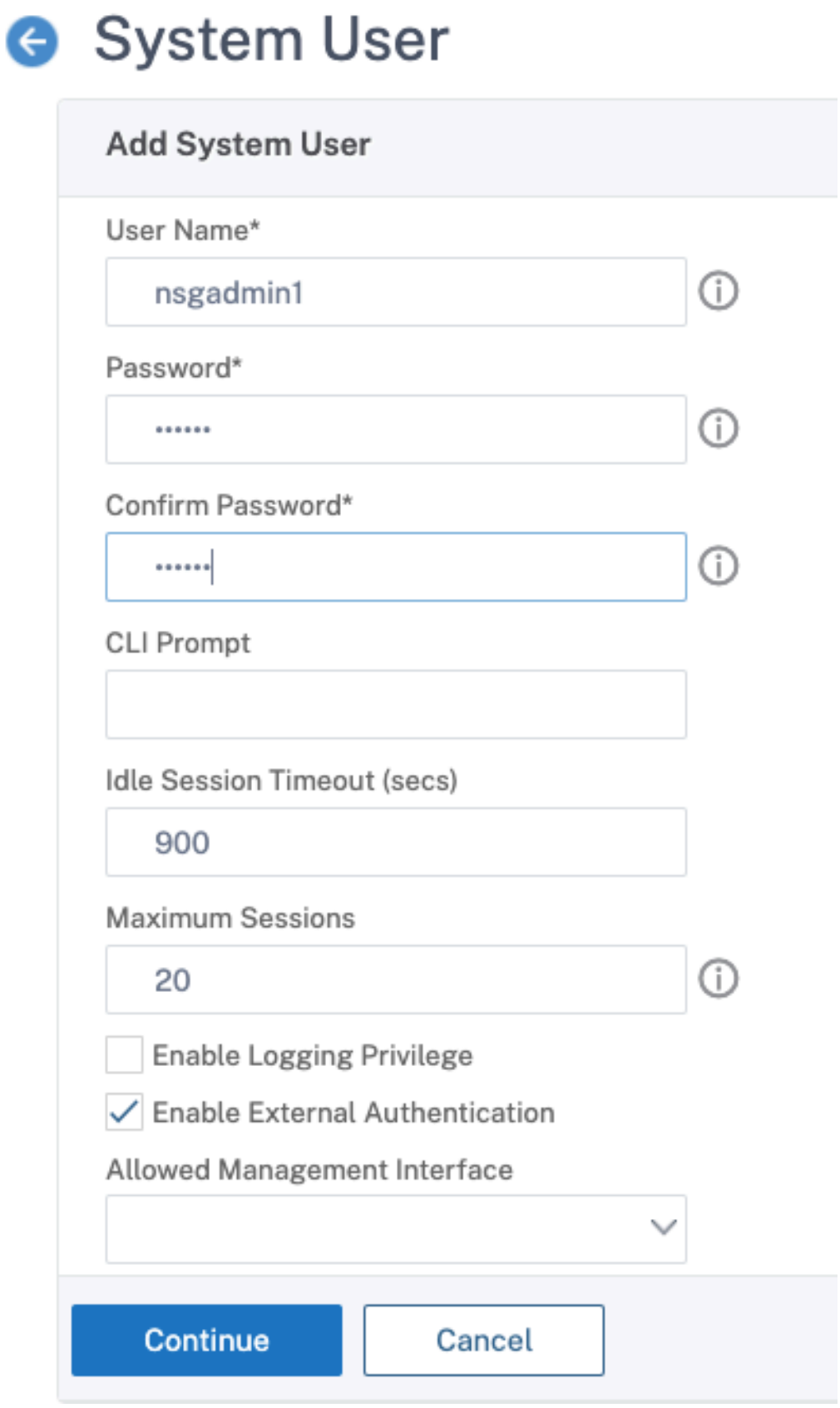
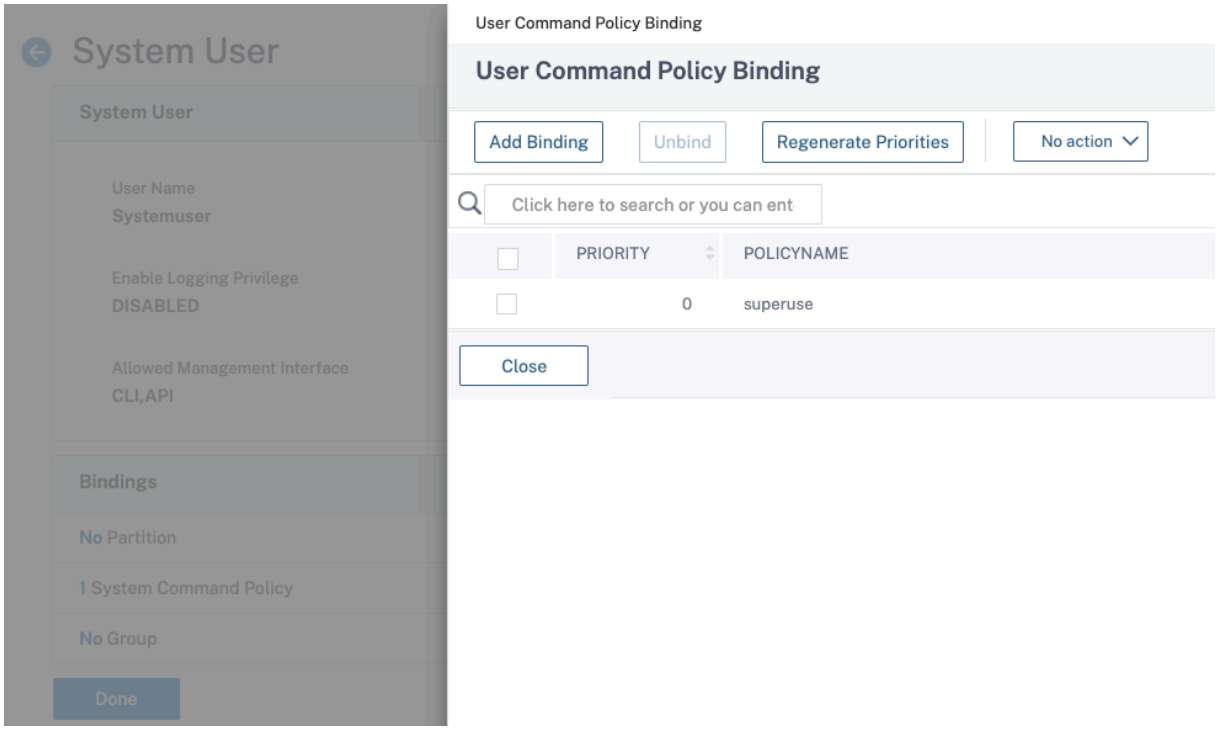
The Citrix ADC appliance can now validate end-to-end LDAP authentication through GUI. To validate this feature, a new "test" button is introduced in the GUI. A Citrix ADC appliance administrator can use this feature to achieve the following benefits:

- Consolidates the complete flow (packet engine – Citrix ADC AAA daemon – external server) to provide better analysis
- Reduces time on validating and troubleshooting issues related to individual scenarios

You have two options to configure and view the test results of LDAP end-to-end authentication by using the GUI.

From system option

1. Navigate to **System > Authentication > Basic Policies > LDAP,** click **Servers** tab.
2. Select the available **LDAP action** from the list.
3. On the **Configure Authentication LDAP Server** page, you have two options under **Connections Settings** section.
4. To check the LDAP server connection, click **Test LDAP Reachability** tab. You can view a pop-up message of successful connection to LDAP server with TCP port details and authenticity of valid credentials.
5. To view the end-to-end LDAP authentication, click **Test End User Connection** link.
6. On the **Test End User Connection** page, click **Test**.
    - On the authentication page, enter the valid credentials to log in. The success screen is displayed.


```


- If the authentication fails, the error screen is displayed.




**From Authentication option**

1. Navigate to **Authentication > Dashboard**, select the available LDAP action from the list.
2. On the **Configure Authentication LDAP Server** page, you have two options under **Connections Settings** section.
3. To check the LDAP server connection, click **Test LDAP Reachability** tab. You can view a pop-up message of successful connection to LDAP server with TCP port details and authenticity of valid credentials.
4. To view the end-to-end LDAP authentication status, click **Test End User Connection** link.
5. On the **Test End User Connection** page, click **Test**.
    - On the authentication page, enter the valid credentials to log in. The success screen is displayed.


- If the authentication fails, the error screen is displayed.

 




### 14-day password expiry notification for LDAP authentication

The Citrix ADC appliance now supports 14-day password expiry notification for LDAP based authentication. By using this feature, administrators can notify the end users about the password expiry threshold time in days. The 14-day password expiry notification is a precursor to self-service password reset (SSPR).

> **Note**
>
> The maximum value or threshold time in days for password expiry notification is 255 days.

#### Advantages of password expiry notification

- Permit users to reset their passwords on their own and provide administrators a flexible way to notify end user about their password expiry in days.
- Eliminates end user dependence to track their password expiration days.
- Sends notifications to the VPN portal page to the users (based on the number of days) to change their password before expiry.

> **Note**
>
> This feature is applicable only for LDAP based authentication schemes, not for RADIUS or TACACS.

#### Understanding the 14-day password notification

The Citrix ADC appliance fetches two attributes (`Max-Pwd-Age and Pwd-Last-Set`) from LDAP authentication server.

- **Max-Pwd-Age.** This attribute denotes the maximum amount of time, in 100-nanosecond intervals, until the password is valid. The value is stored as a large integer that represents the number of 100-nanosecond intervals from the time the password was set before the password expires.

 

- **Pwd-Last-Set.** This attribute determines the date and time at which the password for an account was last changed.

By fetching the two attributes from the LDAP authentication server, Citrix ADC appliance determines the time left for the password to expire for a particular user. This information is collected when any user credentials are validated on the authentication server and a notification is sent back to the user.

A new parameter "pwdExpiryNotification" is introduced in `set aaa parameter`command. By using this parameter, an administrator can keep track the number of days left for password expiry. The Citrix ADC appliance can now start notifying the end user about their password expiry.

> **Note**
>
> Currently, this feature works only for authentication servers having Microsoft AD servers with LDAP implementation. Support for OpenLDAP based servers is targeted later.

Following is an example of the flow of events for setting a 14-day password expiry notification:

1. An administrator, by using Citrix ADC appliance, sets a time (14-days) for password expiration.
2. The user sends an HTTP or HTTPS request to access a resource on the back-end server.
3. Before providing access, the Citrix ADC appliance validates the user credentials with what is configured on the LDAP authentication server.
4. Along with this query to the authentication server, the Citrix ADC appliance carries the request to fetch the details of two attributes (`Max-Pwd-Age and Pwd-Last-Set`).
5. Based on the time left for the password to expire, an expiry notification is displayed.
6. The user then takes appropriate action to update the password.

#### To configure 14-day expiry notification by using the command line interface

> **Note**
>
> 14-day expiry notification can be configured for clientless VPN and Full VPN use cases and not for ICA Proxy.

At the command prompt, type the following commands:

- `set aaa parameter -pwdExpiryNotificationDays <positive_integer><!--NeedCopy-->`
- `show aaa parameter<!--NeedCopy-->`

Example

```
1 > set aaa parameter -pwdExpiryNotificationDays 14
2 Done
3 > show aaa parameter   Configured AAA parameters
      EnableStaticPageCaching: YES EnableEnhancedAuthFeedback: NO
    DefaultAuthType: LOCAL MaxAAAUsers: Unlimited
```


```
                                                AAAD nat ip: None
      EnableSessionStickiness : NO  aaaSessionLoglevel : INFORMATIONAL
                    AAAD Log Level : INFORMATIONAL                 Dynamic
       address: OFF
4     GUI mode: ON
5     Max Saml Deflate Size: 1024              Password Expiry
         Notification Days: 14
```

To configure 14-day expiry notification by using GUI

1. Navigate to **Security > AAA - Application Traffic > Authentication Settings**.
2. Click **Change authentication AAA settings**.
3. On the **Configure AAA Parameter** page, specify the days in the **Password Expiry Notification(days)** field.



4. Click **OK.**

   The notification appears on the top right corner of VPN portal page.



Configure LDAP authentication on the Citrix ADC appliance for management purposes

November 8, 2021

You can configure user logon to the Citrix ADC appliance using the active directory credentials (user name and password) for management purposes (superuser, read-only, network privileges and all others).

Prerequisites

- Windows Active Directory domain controller servers
- A dedicated domain group for NetScaler administrators
- Citrix Gateway 10.1 and later versions

The following figures illustrate the LDAP authentication on the Citrix ADC appliance.



High level configuration steps

1. Create an LDAP server
2. Create an LDAP policy
3. Bind the LDAP policy
4. Assign privileges to your administrators by one of the following ways
    - Apply privileges on group
    - Apply privileges individually for each user

Create an authentication LDAP server

1. Navigate to **System > Authentication > LDAP**.
2. Click the **Server** tab and then click **Add**.
3. Complete the configuration, and then click **Create**.



> **Note:**
>
> In this example, the access is limited to the Citrix ADC appliance by filtering the authentication on the user group membership by setting Search Filter. The value used for this example is - &(memberof=CN=NSG_Admin,OU=AdminGroups,DC=Citrix,DC=lab)

Create an LDAP Policy

1. Navigate to **System > Authentication > Advanced Policies > Policy**.
2. Click **Add**.
3. Enter a name for the policy, select the server that you created in the previous steps.
4. In the Expression text field, enter the appropriate expression, and then click **Create**.



Bind the LDAP policy globally

1. Navigate to **System > Authentication > Advanced Policies > Policy**.
2. In the Authentication Policies page, click **Global Bindings**.
3. Select the policy you created (in this example, pol_LDAPmgmt).
4. Choose a priority accordingly (the lower the number, the higher the priority)
5. Click **Bind,** and then **Done**. A green checkmark appears in the **Globally Bound** column.



Assign privileges to your administrators

You can choose one of the following two options.

- **Apply privileges on a group:** Add a group in the Citrix ADC appliance and assign the same access rights for each user who is a member of this group.
- **Apply privileges individually for each user:** Create each user administrator account and assign rights for each of them.

Apply privileges on a group

When you apply privileges on a group, users who are member of the Active Directory group configured in the Search filter (in this example, NSG_Admin) can connect to the Citrix ADC Management interface and have superuser command policy.

1. Navigate to **System > User Administration > Groups**.
2. Enter the details as per the requirement, and then click **Create**.



You have defined the active directory group that the users belong to and also the command policy level that must be associated to the account when logging in. You can add new administrator users to the LDAP group you configured on the search filter.

> **Note:**
>
> The group name must match the active directory record.

Apply privileges individually for each user

In this scenario, users who are member of your Active Directory group configured in the search filter (in this example, NSG_Admin) can connect to the Citrix ADC management interface but do not have any privileges until you create the specific user on the Citrix ADC appliance and bind the command policy to it.

1. Navigate to **System > User Administration > Users**.
2. Click **Add**.
3. Enter the details as per the requirement.

   Note: Make sure to select **Enable External Authentication**.



1. Click **Continue**.

You have defined the active directory user and the command policy level that must be associated to the account when logging in.

> **Note:**
>
> - The user name must match the existing user's active directory record.
> - When you add a user to the Citrix ADC for external authentication, you must provide a password, if the external authentication is not available. For the external authentication to work properly, the internal password must not be the same as the user account LDAP password.

Add command policy to the user

1. Navigate to **System > User Administration > Users**.
2. Select the user that you created, and then click **Edit.**
3. In Bindings, click **System Command Policy**.
4. Select the correct command policy to apply to your user.
5. Click **Bind**, then click **Close**.



To add more administrators;

- Add the administrator users to the LDAP group you configured on the search filter.
- Create the system user in Citrix ADC and assign the correct command policy.

To configure LDAP authentication on the Citrix ADC appliance for management purposes by using the CLI

Use the following commands as a reference to configure log on for a group with superuser privileges on the Citrix ADC appliance CLI.

1. Create an LDAP server

```
add authentication ldapAction LDAP_mgmt -serverIP myAD.citrix.lab
    -serverPort 636 -ldapBase "DC=citrix,DC=lab" -ldapBindDn
    readonly@citrix.lab -ldapBindDnPassword -ldapLoginName
    sAMAccountName -searchFilter "&(memberof=CN=NSG_Admin,OU=
    AdminGroups,DC=citrix,DC=lab)" -groupAttrName memberOf
<!--NeedCopy-->
```

2. Create and LDAP policy

```
add authentication policy pol_LDAPmgmt -rule true -action
    LDAP_mgmt
<!--NeedCopy-->
```

3. Binding the LDAP policy

```
bind system global pol_LDAPmgmt -priority 110
<!--NeedCopy-->
```

4. Assign privileges to your administrators

   - To apply privileges on the group

```
add system group NSG_Admin
bind system group NSG_Admin -policyName superuser 100
<!--NeedCopy-->
```

   - To apply privileges individually for each user

```
add system user admyoa
bind system user admyoa superuser 100
<!--NeedCopy-->
```

RADIUS authentication

September 14, 2021

As with other types of authentication policies, a Remote Authentication Dial In User Service (RADIUS) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. However, setting up a RADIUS authentication policy has certain special requirements that are described below.

Normally you configure the Citrix ADC to use the IP address of the authentication server during authentication. With RADIUS authentication servers, you can now configure the ADC to use the FQDN of the RADIUS server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex authentication, authorization, and auditing configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you substitute the **serverName** parameter for the **serverIP** parameter.

Before you decide whether to configure the Citrix ADC to use the IP or the FQDN of your RADIUS server to authenticate users, consider that configuring authentication, authorization, and auditing to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

> **Note**
>
> These instructions assume that you are already familiar with the RADIUS protocol and have already configured your chosen RADIUS authentication server.

To add an authentication action for a RADIUS server by using the command line interface

If you authenticate to a RADIUS server, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

```
1 add authentication radiusAction <name> [-serverip <IP> | -serverName] <
     FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
2  -radKey  }
3   [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
       <positive_integer>][-radAttributeType <positive_integer>][-
       radGroupsPrefix <string>] [-radGroupSeparator <string>][-
       passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
       ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
       pwdVendorID <positive_integer> [-pwdAttributeType <
       positive_integer>]] [-defaultAuthenticationGroup <string>] [-
       callingstationid ( ENABLED | DISABLED )]
4
```



```
5 <!--NeedCopy-->
```

The following example adds a RADIUS authentication action named **Authn-Act-1**, with the server IP **10.218.24.65**, the server port **1812**, the authentication timeout **15** minutes, the radius key **WareTheLorax**, NAS IP disabled, and NAS ID **NAS1**.

```
1 add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -
     serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
     DISABLED -radNASid NAS1
2 Done
3
4 <!--NeedCopy-->
```

The following example adds the same RADIUS authentication action, but using the server FQDN **rad01.example.com** instead of the IP.

```
1 add authentication radiusaction Authn-Act-1 -serverName rad01.example.
     com -serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
     DISABLED -radNASid NAS1
2 Done
3
4 <!--NeedCopy-->
```

To configure an authentication action for an external RADIUS server by using the command line

To configure an existing RADIUS action, at the command prompt, type the following command:

```
1 set authentication radiusAction <name> [-serverip <IP> | -serverName] <
     FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
2  -radKey  }
3   [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
       <positive_integer>][-radAttributeType <positive_integer>][-
       radGroupsPrefix <string>] [-radGroupSeparator <string>][-
       passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
       ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
       pwdVendorID <positive_integer> [-pwdAttributeType <
       positive_integer>]] [-defaultAuthenticationGroup <string>] [-
       callingstationid ( ENABLED | DISABLED )]
4
5 <!--NeedCopy-->
```


To remove an authentication action for an external RADIUS server by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

```
rm authentication radiusAction <name>

<!--NeedCopy-->
```

Example

```
rm authentication radiusaction Authn-Act-1
Done

<!--NeedCopy-->
```

To configure a RADIUS server by using the configuration utility

> **Note**
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Radius**
2. In the details pane, on the **Servers** tab, do one of the following:
   - To create a new RADIUS server, click **Add**.
   - To modify an existing RADIUS server, select the server, and then click **Edit**.
3. In the **Create Authentication RADIUS Server** or **Configure Authentication RADIUS Server** dialog, type or select values for the parameters. To fill out parameters that appear beneath **Send Calling Station ID**, expand **Details**.
   - Name*—radiusActionName (Cannot be changed for a previously configured action)
   - Authentication Type*—authtype (Set to RADIUS, cannot be changed)
   - Server Name / IP Address*—Choose either Server Name or Server IP
     - Server Name*—serverName <FQDN>
     - IP Address*—serverIp <IP> If the server is assigned an IPv6 IP address, select the IPv6 check box.
   - Port*—serverPort
   - Time-out (seconds)*—authTimeout
   - Secret Key*—radKey (RADIUS shared secret.)

- Confirm Secret Key*—Type the RADIUS shared secret a second time. (No command line equivalent.)
- Send Calling Station ID—callingstationid
- Group Vendor Identifier—radVendorID
- Group Attribute Type—radAttributeType
- IP Address Vendor Identifier—ipVendorID
- pwdVendorID—pwdVendorID
- Password Encoding—passEncoding
- Default Authentication Group—defaultAuthenticationGroup
- NAS ID—radNASid
- Enable NAS IP address extraction—radNASip
- Group Prefix—radGroupsPrefix
- Group Separator—radGroupSeparator
- IP Address Attribute Type—ipAttributeType
- Password Attribute Type—pwdAttributeType
- Accounting—accounting

4. Click **Create** or **OK**. The policy that you created appears in the Servers page.

Support to pass through RADIUS attribute 66 (Tunnel-Client-Endpoint)

The Citrix ADC appliance now allows the pass-through of RADIUS attribute 66 (Tunnel-Client-Endpoint) during RADIUS authentication. By applying this feature, the clients IP address is received by second-factor authentication from entrusting to make risk-based authentication decisions.

A new attribute “tunnelEndpointClientIP” is introduced in both “add authentication radiusAction” and “set radiusParams” command.

To use this feature, at the Citrix ADC appliance command prompt, type:

```
add authentication radiusAction <name> {
 -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> }
  }
  [-serverPort <port>] … [-tunnelEndpointClientIP (ENABLED|DISABLED)]

set radiusParams {
 -serverIP <ip_addr|ipv6_addr|*> |{
```



```
 -serverName <string> }
  }
  [-serverPort<port>] … [-tunnelEndpointClientIP(ENABLED|DISABLED)]

<!--NeedCopy-->
```

Example

```
add authentication radiusAction radius -severIP 1.217.22.20 -serverName
    FQDN -serverPort 1812 -tunnelEndpointClientIp ENABLED

set radiusParams -serverIp 1.217.22.20 -serverName FQDN1 -serverPort
   1812 -tunnelEndpointClientIP ENABLED

<!--NeedCopy-->
```

Support for validating end-to-end RADIUS authentication

The Citrix ADC appliance can now validate end-to-end RADIUS authentication through a GUI. To validate this feature, a new “test” button is introduced in GUI. A Citrix ADC appliance administrator can leverage this feature to achieve the following benefits:

- Consolidates the complete flow (packet engine – aaa daemon – external server) to provide better analysis
- Reduces time on validating and troubleshooting issues related to individual scenarios

You have two options to configure and view the test results of RADIUS end-to-end authentication by using the GUI.

From system option

1. Navigate to **System > Authentication > Basic Policies > RADIUS,** click **Servers** tab.
2. Select the available **RADIUS action** from the list.
3. On the **Configure Authentication RADIUS Server** page, you have two options under **Connections Settings** section.
4. To check the RADIUS server connection, click **Test RADIUS Reachability** tab.
5. To view the end-to-end RADIUS authentication, click **Test End User Connection** link.

From Authentication option

1. Navigate to **Authentication > Dashboard**, select the available RADIUS action from the list.
2. On the **Configure Authentication RADIUS Server** page, you have two options under **Connections Settings** section.

3. To check the RADIUS server connection, click **Test RADIUS Reachability** tab.
4. To view the end-to-end RADIUS authentication status, click **Test End User Connection** link.

TACACS authentication

September 14, 2021

TACACS authentication policy authenticates to an external Terminal Access Controller Access-Control System (TACACS) authentication server.
After a user authenticates to a TACACS server, the Citrix ADC connects to the same TACACS server for all subsequent authorizations. When a primary TACACS server is unavailable, this feature prevents any delay while the ADC waits for the first TACACS server to time out. It happens before resending the authorization request to the second TACACS server.

> **Note:**
>
> TACACS authorization server does not support commands whose string length exceeds 255 characters.
>
> **Workaround:** Use local authorization instead of a TACACS authorization server.

When authenticating through a TACACS server, authentication, authorization, and auditing traffic management logs only successfully runs TACACS commands. It prevents the logs from showing TACACS commands that are entered by the users who were not authorized to run them.

Starting from NetScaler 12.0 Build 57.x, the Terminal Access Controller Access-Control System (TACACS) is not blocking the authentication, authorization, and auditing daemon while sending the TACACS request. The allow LDAP, and RADIUS authentication to proceed with the request. The TACACS authentication request resumes once the TACACS server acknowledges the TACACS request.

> **Important:**
>
> - Citrix recommends you do not modify any TACACS related configurations when you run a “clear ns config” command.
> - TACACS related configuration related to advanced policies is cleared and reapplied when the “RBAconfig” parameter is set to NO in “clear ns config” command for advanced policy.

Name-value attribute support for TACACS authentication

You can now configure TACACS authentication attributes with a unique name along with values. The names are configured in the TACACS action parameter and the values are obtained by querying for the names. By specifying the name attribute value, admins can easily search for the attribute value

associated with the attribute name. Also, admins no longer have to remember the attribute by its value alone.

> **Important**
>
> - In the tacacsAction command, you can configure a maximum of 64 attributes separated by comma with total size less than 2048 bytes.

To configure the name-value attributes by using the CLI

At the command prompt, type:

```
add authentication tacacsAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

Example:

```
add authentication tacacsAction tacacsAct1 -attributes  “mail,sn,
   userprincipalName”
<!--NeedCopy-->
```

To add an authentication action by using the command line interface

If you do not use LOCAL authentication, you need to add an explicit authentication action. At the command prompt, type the following command:

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
   port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

Example

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
   serverport 1812                  -authtimeout 15 -tacacsSecret "
   minotaur" -authorization OFF -accounting ON -auditFailedCmds OFF -
   defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

To configure an authentication action by using the command line interface

To configure an existing authentication action, at the command prompt, type the following command:

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
   port>][-authTimeout <positive_integer>][ ... ]
```



```
<!--NeedCopy-->
```

Example

```
    > set authentication tacacsaction Authn-Act-1 -serverip
       10.218.24.65 -serverport 1812                  -authtimeout 15
       -tacacsSecret "minotaur" -authorization OFF -accounting ON -
       auditFailedCmds OFF -defaultAuthenticationGroup "users"    Done
<!--NeedCopy-->
```

To remove an authentication action by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

Example

```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```

Client certificate authentication

September 14, 2021

Web sites that contain sensitive content, such as online banking websites or websites with employee personal information, sometimes require client certificates for authentication. To configure authentication, authorization, and auditing to authenticate users on the basis of client-side certificate attributes, you first enable client authentication on the traffic management virtual server and bind the root certificate to the authentication virtual server. Then, you implement one of two options. You can configure the default authentication type on the authentication virtual server as CERT, or you can create a certificate action that defines what the Citrix ADC must do to authenticate users on the basis of a client certificate. In either case, your authentication server must support CRLs. You configure the ADC to extract the user name from the SubjectCN field or another specified field in the client certificate.

When the user tries to log on to an authentication virtual server for which an authentication policy is not configured, and a global cascade is not configured, the user name information is extracted from the specified field of the certificate. If the required field is extracted, the authentication succeeds. If the user does not provide a valid certificate during the SSL handshake, or if the user name extraction

fails, authentication fails. After it validates the client certificate, the ADC presents a logon page to the user.

The following procedures assume that you have already created a functioning authentication, authorization, and auditing configuration, and therefore they explain only how to enable authentication by using client certificates. These procedures also assume that you have obtained your root certificate and client certificates and have placed them on the ADC in the /nsconfig/ssl directory.

Configure client certificate authentication

To configure the authentication, authorization, and auditing client certificate parameters by using the command line interface

At the command prompt, type the following commands, in the order shown, to configure the certificate and verify the configuration:

```
add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password
    -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod
    <notificationPeriod>

bind ssl certKey <certkeyName> -vServer <certkeyName> -CA -crlCheck
   Mandatory

show ssl certKey [<certkeyName>]

set aaa parameter -defaultAuthType CERT

show aaa parameter

set aaa certParams -userNameField "Subject:CN"

show aaa certParams
<!--NeedCopy-->
```

To configure the authentication, authorization, and auditing client certificate parameters by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Virtual Servers.**
2. In the details pane, select the virtual server that you want to configure to handle client certificate authentication, and then click **Edit.**
3. On the **Configuration** page, under **Certificates**, click the right arrow (>) to open the CA Cert Key installation dialog.
4. In the **CA Cert Key** dialog box, click **Insert.**

5. In the **CA Cert Key - SSL Certificates** dialog box, click **Install.**
6. In the **Install Certificate** dialog box, set the following parameters, whose names correspond to the CLI parameter names as shown:
   - Certificate-Key Pair Name*—certkeyName
   - Certificate File Name—certFile
   - Key File Name—keyFile
   - Certificate Format—inform
   - Password—password
   - Certificate Bundle—bundle
   - Notify When Expires—expiryMonitor
   - Notification Period—notificationPeriod
7. Click **Install**, and then click **Close**.
8. In the **CA Cert Key** dialog box, in the **Certificate** list, select the root certificate.
9. Click **Save**.
10. Click **Back** to return to the main configuration screen.
11. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > CERT**.
12. In the details pane, select the policy you want to configure to handle client certificate authentication, and then click **Edit**.
13. In the **Configure Authentication CERT Policy** dialog, Server drop-down list, select the virtual server you just configured to handle client certificate authentication.
14. Click **OK**. A message appears in the status bar, stating that the configuration completed successfully.

Client certificate authentication using advanced policies

Following are the steps to configure client certificate authentication on Citrix ADC using advanced policies.

1. Navigate to **Security > AAA - Application Traffic > Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure to handle client certificate authentication, and click **Edit**.

   > **Note:**
   >
   > If you have imported a valid CA certificate and server certificate for the virtual server you can skip **step 3 to step 10**.

3. On the **Configuration** page, under **Certificates**, click **>** to open the **CA Cert Key** installation dialog box.
4. In the **CA Cert Key** dialog box, click **Insert**.
5. In the **CA Cert Key - SSL Certificates** dialog box, click **Install**.

6. In the **Install Certificate** dialog box, set the following parameters, whose names correspond to the CLI parameter names as shown:
    - Certificate-Key Pair Name—certkeyName
    - Certificate File Name—certFile
    - Key File Name—keyFile
    - Certificate Format—inform
    - Password—password
    - Certificate Bundle—bundle
    - Notify When Expires—expiryMonitor
    - Notification Period—notificationPeriod
7. Click **Install**, and then click Close.
8. In the **CA Cert Key** dialog box, from the Certificate list, select the root certificate.
9. Click **Save**.
10. Click **Back** to return to the main configuration screen.
11. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies**, and then select **Policy**.
12. In the details pane do one of the following:
    - To create a new policy, click **Add**.
    - To modify an existing policy, select the policy, and then click **Edit**.
13. In the **Create Authentication Policy** or **Configure Authentication Policy** dialog box, type or select values for the parameters.
    - Name - The policy name. Cannot be changed for a previously configured policy.
    - Action Type - Select Cert
    - Action - The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the plus and create a new action of the proper type.
    - Log Action - The audit action to associate with the policy. You can choose an existing audit action, or click the plus and create a new action.
    - Expression - The rule that selects connections to which you want to apply the action that you specified. The rule can be simple (“true” selects all traffic) or complex. You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open Add Expression dialog box and using the drop-down lists in it to construct your expression.)
    - Comment - You can type a comment that describes the type of traffic that this authentication policy will apply to. Optional.

14. Click **Create** or **OK**, and then click **Close**. If you created a policy, that policy appears in the Authentication Policies and Servers page.

Client certificate pass-through

The Citrix ADC can now be configured to pass client certificates through to protected applications that require client certificates for user authentication. The ADC first authenticates the user, then inserts the client certificate into the request and sends it to the application. This feature is configured by adding appropriate SSL policies.

The exact behavior of this feature when a user presents a client certificate depends upon the configuration of the VPN virtual server.

- If the VPN virtual server is configured to accept client certificates but not require them, the ADC inserts the certificate into the request and then forwards the request to the protected application.
- If the VPN virtual server has client certificate authentication disabled, the ADC renegotiatiates the authentication protocol and reauthenticates the user before it inserts the client certificate in the header and forwards the request to the protected application.
- If the VPN virtual server is configured to require client certificate authentication, the ADC uses the client certificate to authenticate the user, then inserts the certificate in the header and forwards the request to the protected application.

In all of these cases, you configure client certificate pass-through as follows.

Create and configure client certificate pass-through by using the command line interface

At the command prompt, type the following commands:

```
1 add vpn vserver <name> SSL <IP> 443
2 <!--NeedCopy-->
```

For *name*, substitute a name for the virtual server. The name must contain from one to 127 ASCII characters, beginning with a letter or underscore (_), and containing only letters, numbers, and the underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. For <IP>, substitute the IP address assigned to the virtual server.

```
1 set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>
2 <!--NeedCopy-->
```

For <name>, substitute the name of the virtual server that you just created. For <clientCert>, substitute one of the following values:

- disabled—disables client certificate authentication on the VPN virtual server.

- mandatory—configures the VPN virtual server to require client certificates to authenticate.
- optional—configures the VPN virtual server to allow client certificate authentication, but not to require it.

```
bind vpn vserver <name> -policy local
<!--NeedCopy-->
```

For <name>, substitute the name of the VPN virtual server that you created.

```
bind vpn vserver <name> -policy cert
<!--NeedCopy-->
```

For <name>, substitute the name of the VPN virtual server that you created.

```
bind ssl vserver <name> -certkeyName <certkeyname>
<!--NeedCopy-->
```

For <name>, substitute the name of the virtual server that you created. For <certkeyName>, substitute the client certificate key.

```
bind ssl vserver <name> -certkeyName <cacertkeyname> -CA -ocspCheck
   Optional
<!--NeedCopy-->
```

For <name>, substitute the name of the virtual server that you created. For <cacertkeyName>, substitute the CA certificate key.

```
add ssl action <actname> -clientCert ENABLED -certHeader CLIENT-CERT
<!--NeedCopy-->
```

For <actname>, substitute a name for the SSL action.

```
add ssl policy <polname> -rule true -action <actname>
<!--NeedCopy-->
```

For <polname>, substitute a name for your new SSL policy. For <actname>, substitute the name of the SSL action that you just created.

```
bind ssl vserver <name> -policyName <polname> -priority 10
<!--NeedCopy-->
```

For <name>, substitute the name of the VPN virtual server.

Example


```
add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert
   optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck
   Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-
   CERT
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority
    10
<!--NeedCopy-->
```

Negotiate authentication

September 14, 2021

As with other types of authentication policies, a Negotiate authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

In addition to standard authentication functions, the Negotiate Action command can now extract user information from a keytab file instead of requiring you to enter that information manually. If a keytab has more than one SPN, authentication, authorization, and auditing selects the correct SPN. You can configure this feature at the command line, or by using the configuration utility.

> **Note**
>
> These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.

To configure authentication, authorization, and auditing to extract user information from a keytab file by using the command line interface

At the command prompt, type the appropriate command:

```
add authentication negotiateAction <name> {
 -domain <string> }
  {
```



```
 -domainUser <string> }
  {
 -domainUserPasswd  }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]

set authentication negotiateAction <name> {
 -domain <string> }
  {
 -domainUser <string> }
  {
 -domainUserPasswd }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]
<!--NeedCopy-->
```

Parameter description

- **name** - Name of the negotiate action to be used.
- **domain** - Domain name of the service principal that represnts Citrix ADC.
- **domainUser** - User name of the account that is mapped with Citrix ADC principal. This can be given along with domain and password when keytab file is not available. If username is given along with keytab file, then that keytab file will be searched for this user's credentials. Maximum Length: 127
- **domainUserPasswd** - Password of the account that is mapped to the Citrix ADC principal.
- **defaultAuthenticationGroup** - This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum Length: 63
- **keytab** - The path to the keytab file that is used to decrypt kerberos tickets presented to Citrix ADC. If keytab is not available, domain/username/password can be specified in the negotiate action configuration. Maximum Length: 127
- **NTLMPath** - The path to the site that is enabled for NTLM authentication, including FQDN of the server. This is used when clients fallback to NTLM. Maximum Length: 127

To configure authentication, authorization, and auditing to extract user information from a keytab file by using the configuration utility

> Note
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Authentication > Advanced Policies > Actions > NEGOTIATE Actions**.

2. In the details pane, on the **Servers** tab, do one of the following:
   - If you want to create a new **Negotiate** action, click **Add**.
   - If you want to modify an existing **Negotiate** action, in the data pane select the action, and then click **Edit**.
3. If you are creating a new **Negotiate** action, in the **Name** text box, type a name for your new action. The name can be from one to 127 characters in length and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing Negotiate action, skip this step. The name is read-only; you cannot change it.
4. Under **Negotiate**, if the Use Keytab file check box is not already checked, check it.
5. In the Keytab file path text box, type the full path and filename of the keytab file that you want to use.
6. In the Default authentication group text box, type the authentication group that you want to set as default for this user.
7. Click **Create** or **OK** to save your changes.

Points to note when advanced encryptions is used for Kerberos authentication

> - **Sample configuration when keytab is used:** add authentication negotiateAction neg_act_aes256 -keytab "/nsconfig/krb/lbvs_aes256.keytab"
> - **Use the following command when keytab has multiple encryption types.** The command additionally captures domain user parameters: add authentication negotiateAction neg_act_keytab_all -keytab "/nsconfig/krb/lbvs_all.keytab" -domainUser "HTTP/lbvs.aaa.local"
> - **Use the following commands when user credential are used:** add authentication negotiateAction neg_act_user -domain AAA.LOCAL -domainUser "HTTP/lbvs.aaa.local" -domainUserPasswd <password>
> - Ensure that the correct **domainUser** information is provided. You can look for the user logon name in AD.

Web authentication

September 14, 2021

Authentication, authorization, and auditing is now able to authenticate a user to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful. As with other types of authentication

policies, a Web authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

To set up web-based authentication with a specific web server, first you create a web authentication action. Since authentication to web servers does not use a rigid format, you must specify exactly which information the web server requires and in which format when creating the action. To do this, you create an expression in Citrix ADC appliance default syntax that contains the following items:

- **Server IP**—The IP address of the authentication Web server.
- **Server Port**—The port of the authentication Web server.
- **Authentication Rule**—An expression in Citrix ADC appliance default syntax that contains the user's credentials in the format that the Web server expects.
- **Scheme**—HTTP (for unencrypted web authentication) or HTTPS (for encrypted web authentication).
- **Success Rule**—An expression in Citrix ADC appliance default syntax that matches the web server response string that signifies that the user authenticated successfully.

For all other parameters, follow the normal rules for the add authentication action command.

Next you create a policy associated with that action. The policy is similar to an LDAP policy, and like LDAP policies uses Citrix ADC appliance syntax.

> **Note**
>
> These instructions assume that you are already familiar with the authentication requirements of the web server(s) to which you want to authenticate, and have already configured the web authentication server.

To configure a Web authentication action by using the command line interface

To create a web authentication action at the command line, at the command line type the following command:

```
1 add authentication webAuthAction <name> -serverIP <ip_addr|ipv6_addr|*>
      -serverPort <port|*> [-fullReqExpr <string>] -scheme ( http | https
      ) -successRule <expression> [-defaultAuthenticationGroup <string
     >][-Attribute1 <string>][-Attribute2 <string>] [-Attribute3 <string
     >][-Attribute4 <string>] [-Attribute5 <string>][-Attribute6 <string
     >] [-Attribute7 <string>][-Attribute8 <string>] [-Attribute9 <string
     >][-Attribute10 <string>] [-Attribute11 <string>][-Attribute12 <
     string>] [-Attribute13 <string>][-Attribute14 <string>] [-
     Attribute15 <string>][-Attribute16 <string>]
2 <!--NeedCopy-->
```


Example

```
add policy expression post_data "\"username=\" + http.REQ.BODY(1000).
    SET_TEXT_MODE(IGNORECASE).AFTER_STR(\"login=\").BEFORE_STR(\"&\") +
    \"&passwort=\" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).
    AFTER_STR(\"passwd=\")"

add policy expression length_post_data "(\"username= \" + http.REQ.BODY
    (1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR(\"login=\").BEFORE_STR
    (\"&\") + \"passwort=\" + http.REQ.BODY(1000).SET_TEXT_MODE(
    IGNORECASE).AFTER_STR(\"passwd=\")).length"

add authentication webAuthAction webAuth_POST -serverIP 10.106.187.54 -
    serverPort 80 -fullReqExpr q{
 "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
     .version.major + "\r\nAccept:*/*\r\nHost: 10.106.187.54\r\nReferer:
      http://10.106.187.54/MyPHP/auth.php\r\nAccept-Language: en-US\r\
     nUser-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
     Trident/5.0)\r\nContent-Type: application/x-www-form-urlencoded\r\n
     " + "Content-Length: " + length_post_data + "\r\nConnection: Keep-
     Alive\r\n\r\n" + post_data }
  -scheme http -successRule "http.res.status.eq(200)"
<!--NeedCopy-->
```

To configure a Web authentication action by using the configuration utility

> **Note**
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Policies > LDAP**.
2. In the details pane, on the **Servers** tab, do one of the following:
   - If you want to create a new web authentication action, click **Add**.
   - If you want to modify an existing web authentication action, in the data pane select the action, and then click **Edit**.
3. If you are creating a new web authentication action, in the **Create Authentication Web** server dialog box, **Name** text box, type a name for the new web authentication action. The name can be from one to 127 characters in length, and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing web authentication action, skip this step. The name is read-only; you cannot change it.
4. In the **Web Server IP Address** text box, type the IPv4 or IPv6 IP address of the authentication web server. If the address is an IPv6 IP address, select the IPv6 check box first.

5. In the Port text box, type the port number on which the web server accepts connections.
6. Select **HTTP** or **HTTPS** in the **Protocol** drop-down list.
7. In the HTTP Request Expression text area, type a PCRE-format regular expression that creates the web server request that contains the user's credentials in the exact format expected by the authentication web server.
8. In the Expression to validate the Authentication text area, type a Citrix ADC appliance default syntax expression that describes the information in the web server response that indicates that user authentication was successful.
9. Fill out the remaining fields as described in the general authentication action documentation.
10. Click **OK**.

SMS two factor authentication using Web authentication

September 14, 2021

Citrix ADC can now be integrated with a third party SMS provider to provide an extra layer of authentication.

Citrix ADC appliance can be configured to send an OTP on the user's mobile as a second factor of authentication. The appliance presents the user with a logon form to enter the OTP after successful AD login. It is only after the successful second factor authentication the user is presented with the requested resource.

Configure SMS two factor authentication with Citrix ADC

Before you configure the SMS two factor authentication feature, you must have an LDAP authentication configured on a Citrix ADC appliance as first factor with authentication enabled. For instructions to configure LDAP authentication, see To configure LDAP authentication by using the configuration utility.

> **Note**
>
> Mobile number can be extracted using AAA.USER.ATTRIBUTE(1) and can be included while sending it to a back-end server.

Assign NS variable

At the command prompt, type the following commands:


```
add ns variable <variable name> -type "map(text(65),text(6),100000)" -
    ifValueTooBig undef -ifNoValue undef -expires 5

add ns assignment<variable name> -variable "$test[AAA.USER.SESSIONID]"
    -set ("000000" + SYS.RANDOM.MUL(1000000).TYPECAST_UNSIGNED_LONG_AT.
    TYPECAST_TEXT_T).SUFFIX(6)
<!--NeedCopy-->
```

Sample NS Variable assignment

```
add ns variable test -type "map(text(65),text(6),100000)" -
    ifValueTooBig undef -ifNoValue undef -expires 5

add ns assignment test -variable "$test[AAA.USER.SESSIONID]" -set ("
    000000" + SYS.RANDOM.MUL(1000000).TYPECAST_UNSIGNED_LONG_AT.
    TYPECAST_TEXT_T).SUFFIX(6)
<!--NeedCopy-->
```

Configure Webauth action

At the command prompt, type the following commands:

```
add policy expression <expression name> "\"method=sendMessage&send_to=&
    msg=OTP i \" + $test[AAA.USER.SESSIONID] + \"for login into secure
    access  gateway. Valid till EXPIRE_TIME. Do not share the OTP with
    anyone for security reasons.&userid=#####&password=###=1.0\""

add authentication webAuthAction webAuth_Get -serverIP <SERVER_IP> -
    serverPort <SERVER_PORT> -fullReqExpr q{
 "GET /GatewayAPI/rest?" + <expression name> + "HTTP/" + http.req.
     version.major + "." + http.req.version.minor.sub(1) + "\r\nAccept
     :*/*\r\nHost: <FQDN>\r\n" }
  -successRule "http.res.status.eq(200)" -scheme -successRule true

set authentication webAuthAction <web auth action name> <server IP
    address> -serverPort 8080 -fullReqExpr q{
 "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
     .version.major + "\r\nAccept:*/*\r\nHost: <server IP address> \r\
     nContent-Length: 10\r\n\r\n" + <name in the format expected by SMS
     server> }
  -scheme http -successRule true
<!--NeedCopy-->
```

Sample Webauth action configuration


```
add policy expression otp_exp "\"method=sendMessage&send_to=&msg=OTP i
    \" + $test[AAA.USER.SESSIONID] + \"for login into secure access
    gateway. Valid till EXPIRE_TIME. Do not share the OTP with anyone
    for security reasons.&userid=#####&password=###=1.0\""

add authentication webAuthAction webAuth_Get -serverIP -serverIP
    10.106.168.210 -serverPort 8080 -fullReqExpr q{
 "GET /GatewayAPI/rest?" + otp_exp + "HTTP/" + http.req.version.major +
     "." + http.req.version.minor.sub(1) + "\r\nAccept:*/*\r\nHost: <
    FQDN>\r\n" }
  -successRule "http.res.status.eq(200)" -scheme -successRule true

set authentication webAuthAction webAuth_POST -serverIP 10.106.168.210
    -serverPort 8080 -fullReqExpr q{
 "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
    .version.major + "\r\nAccept:*/*\r\nHost: 10.106.168.210 \r\
    nContent-Length: 10\r\n\r\n" + otp_set }
  -scheme http -successRule true
<!--NeedCopy-->
```

Sample first factor configuration

```
add authentication ldapAction ldap_action -serverIP 1.1.1.1 -serverPort
    3268 -authTimeout 30 -ldapBase "dc=nsi-test,dc=com" -ldapBindDn
    Administrator@nsi-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samaccountname -groupAttrName memberOf -
    ssoNameAttribute samaccountname  -Attribute1 mobile -email mail -
    CloudAttributes DISABLED

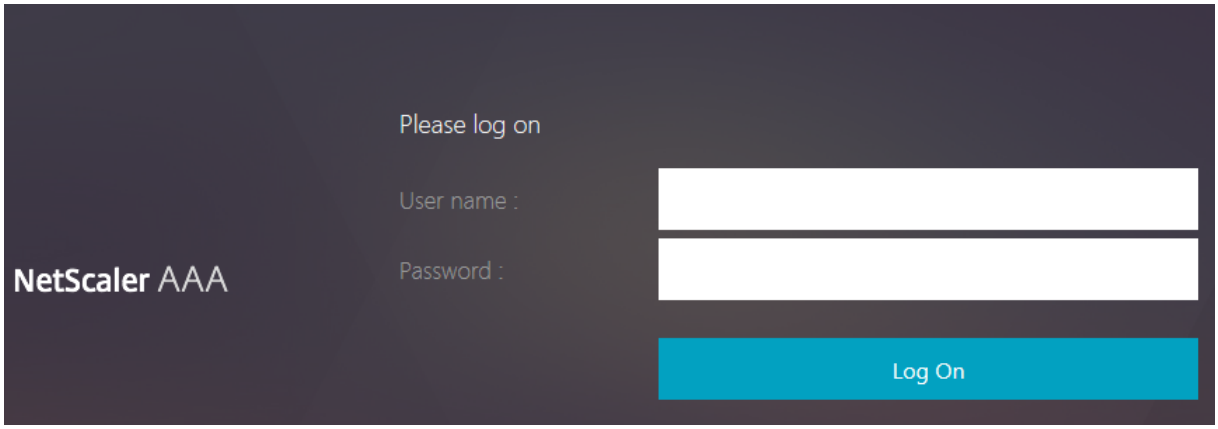
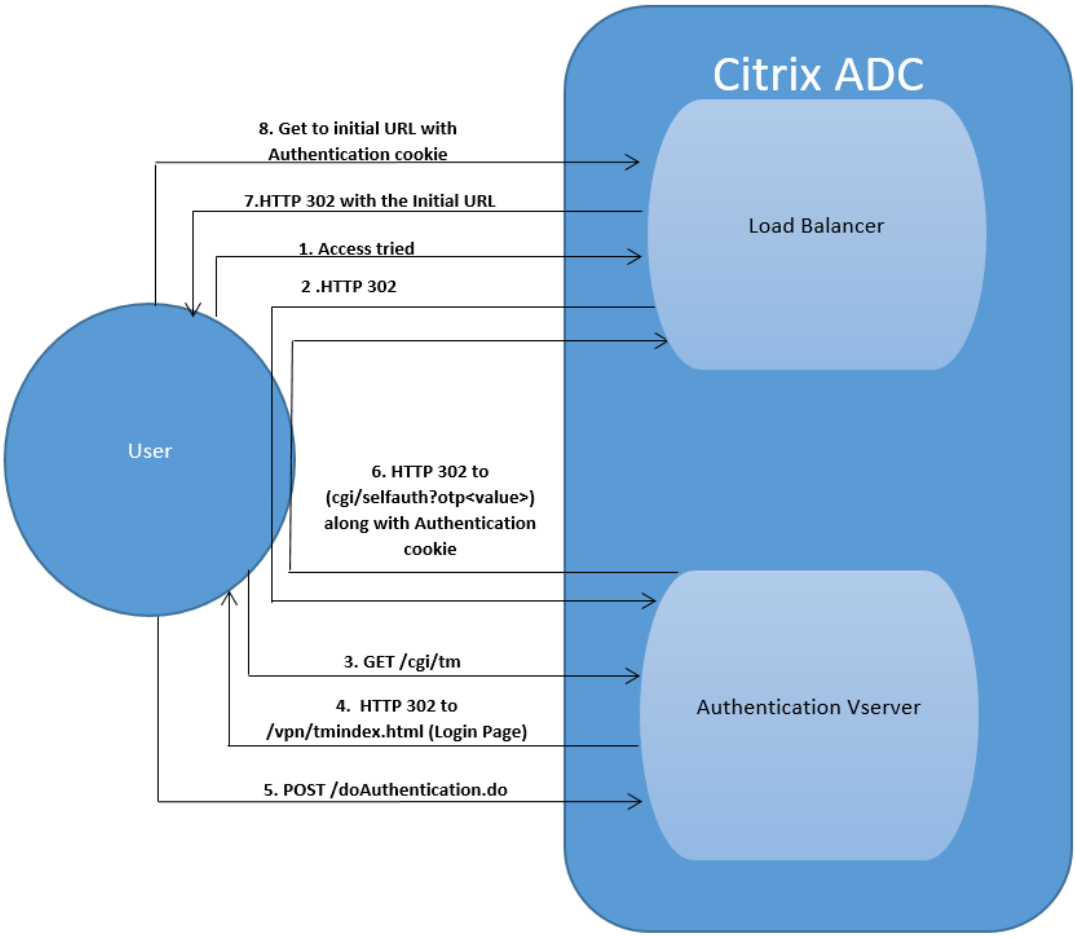
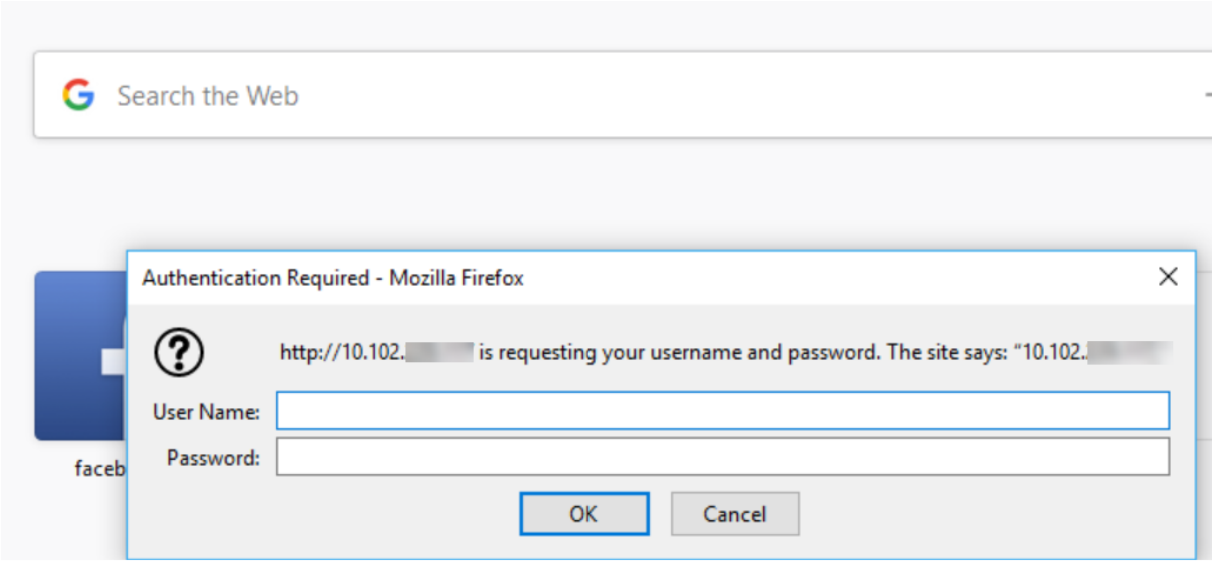
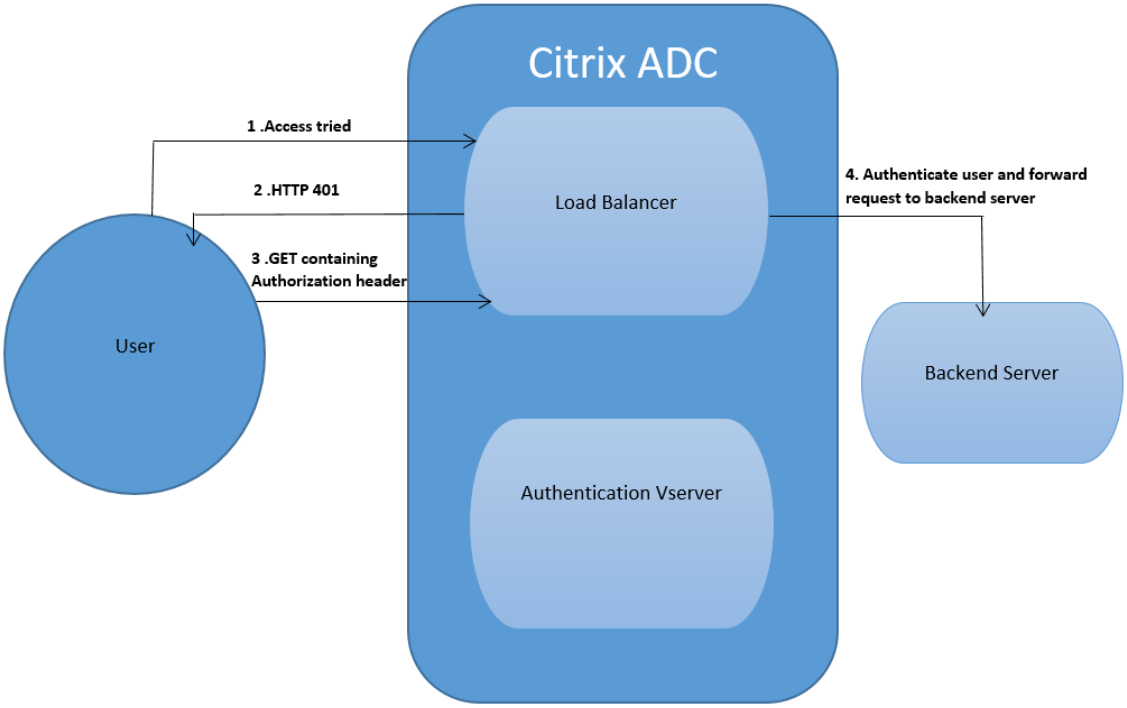
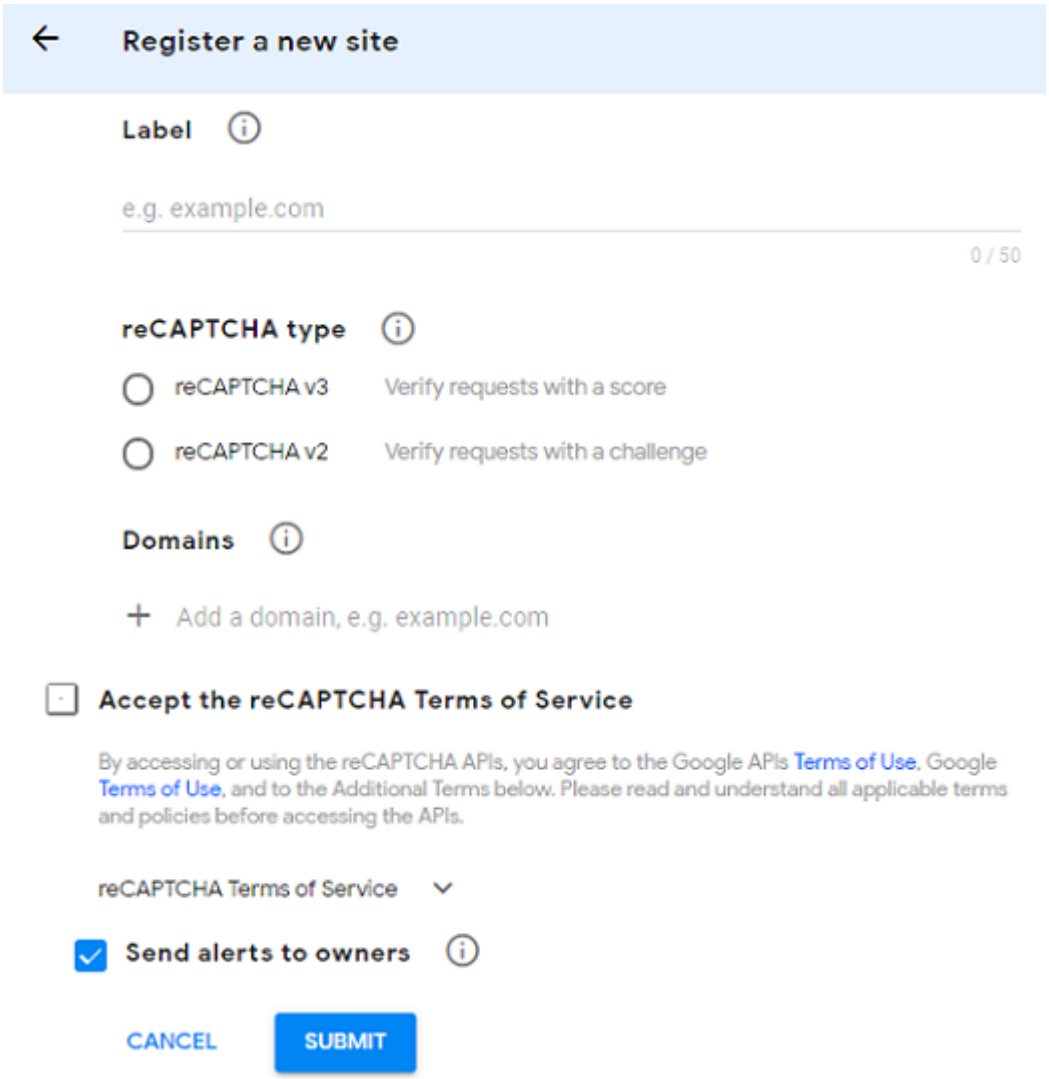
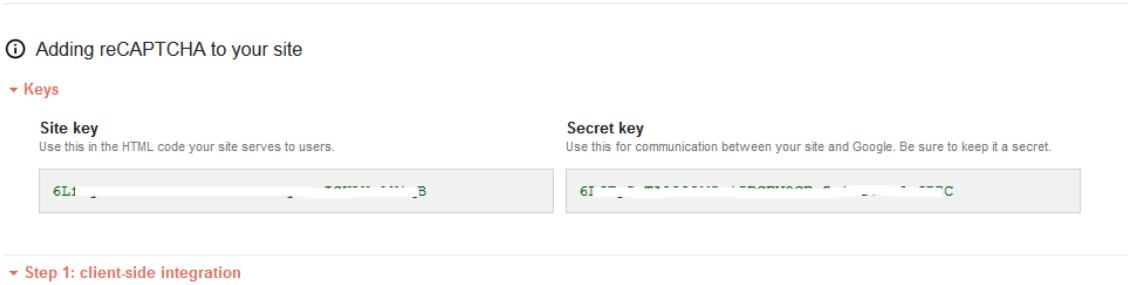
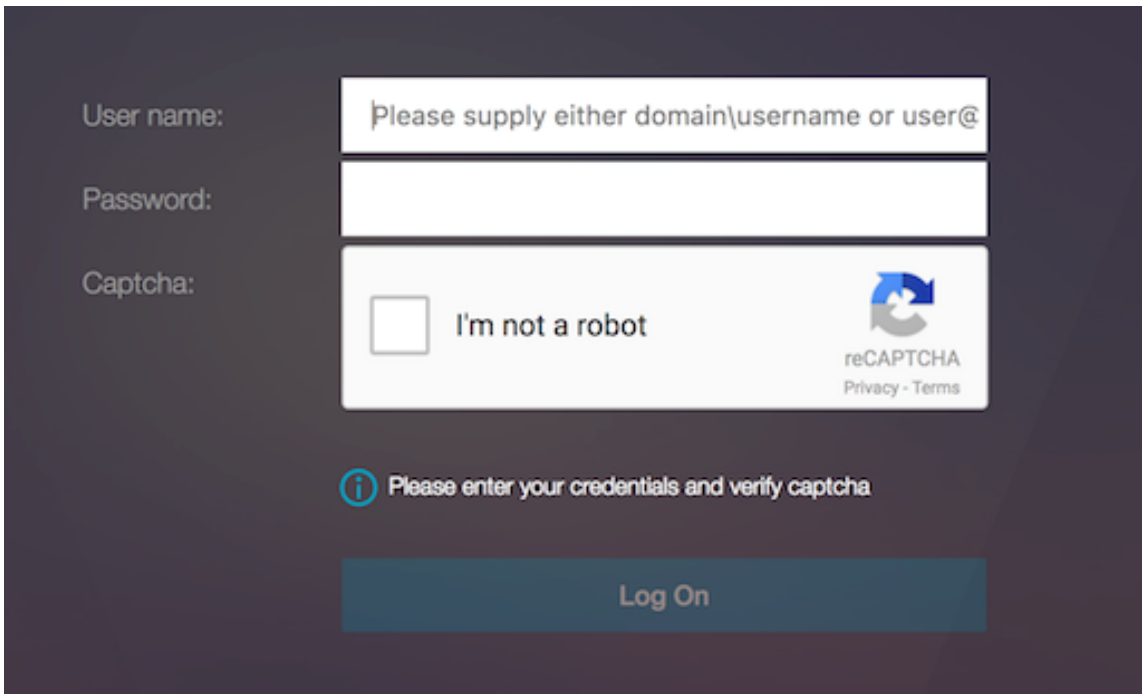
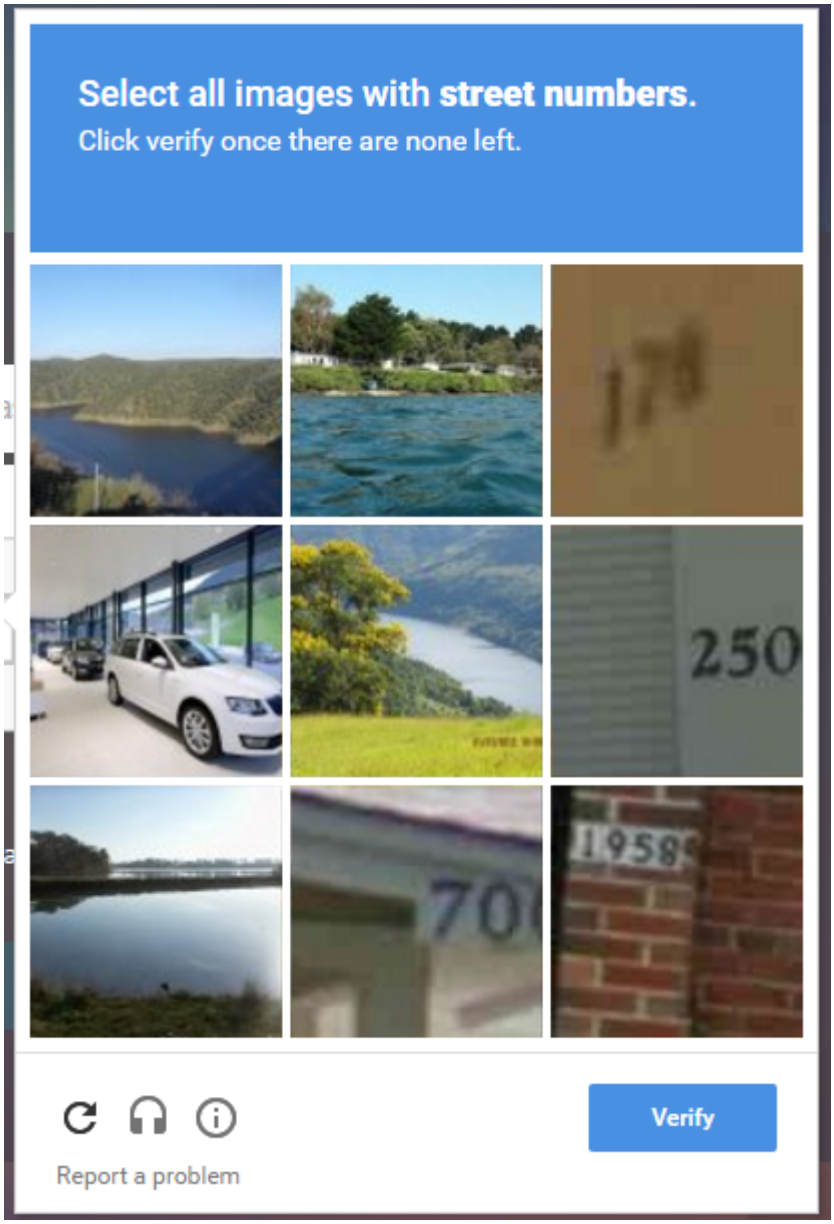
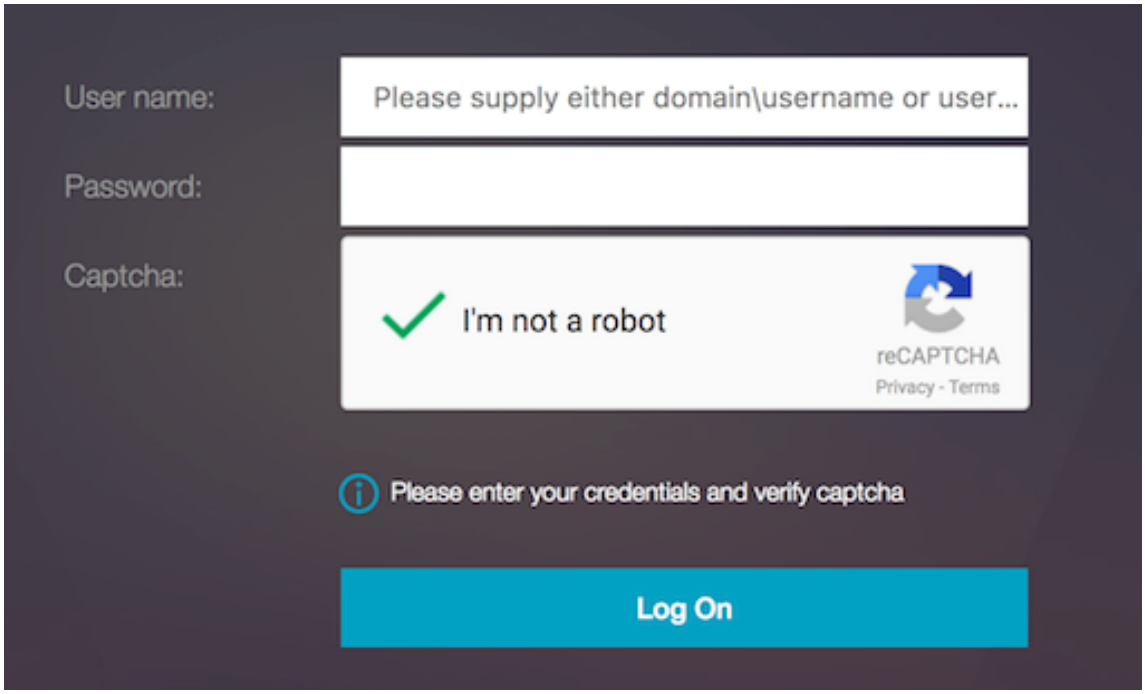
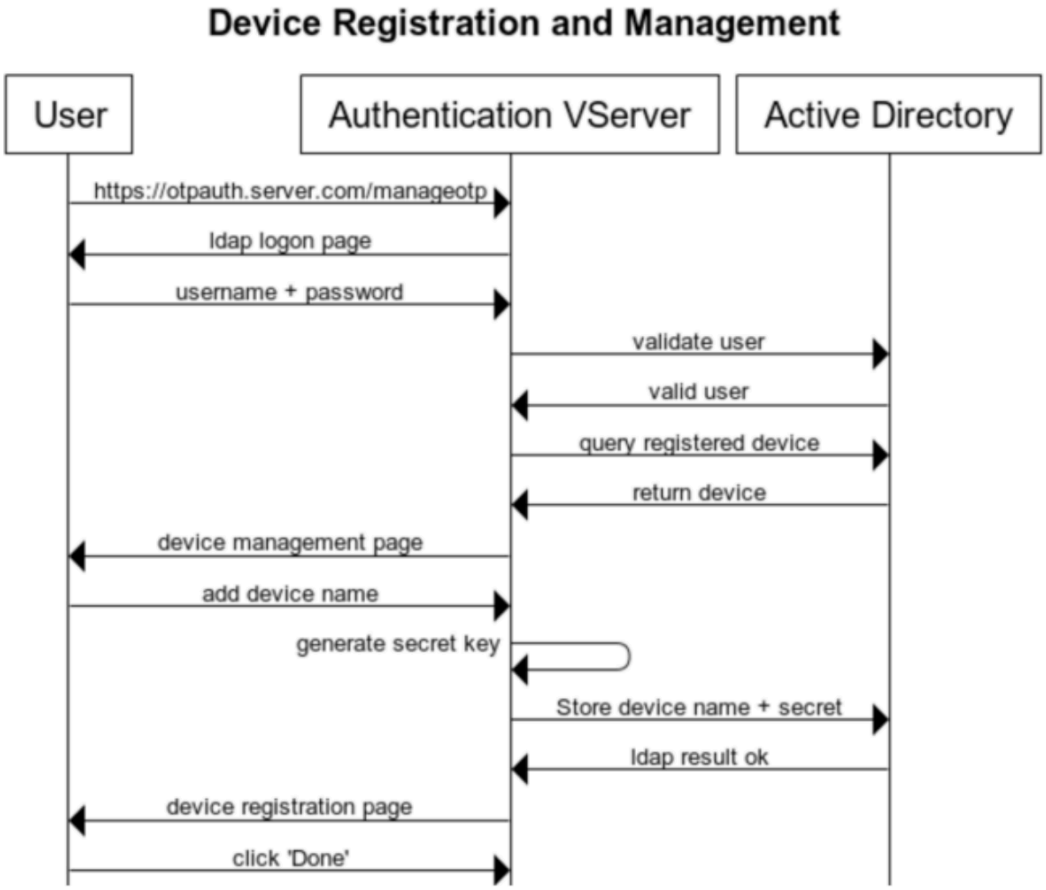
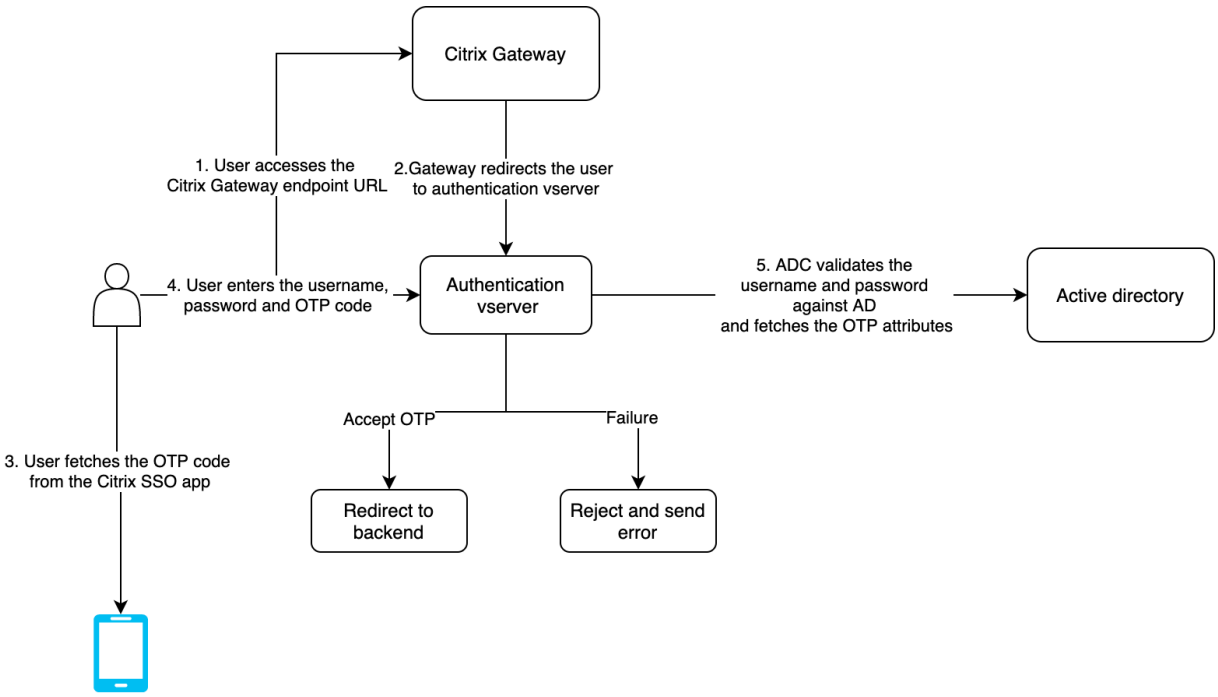
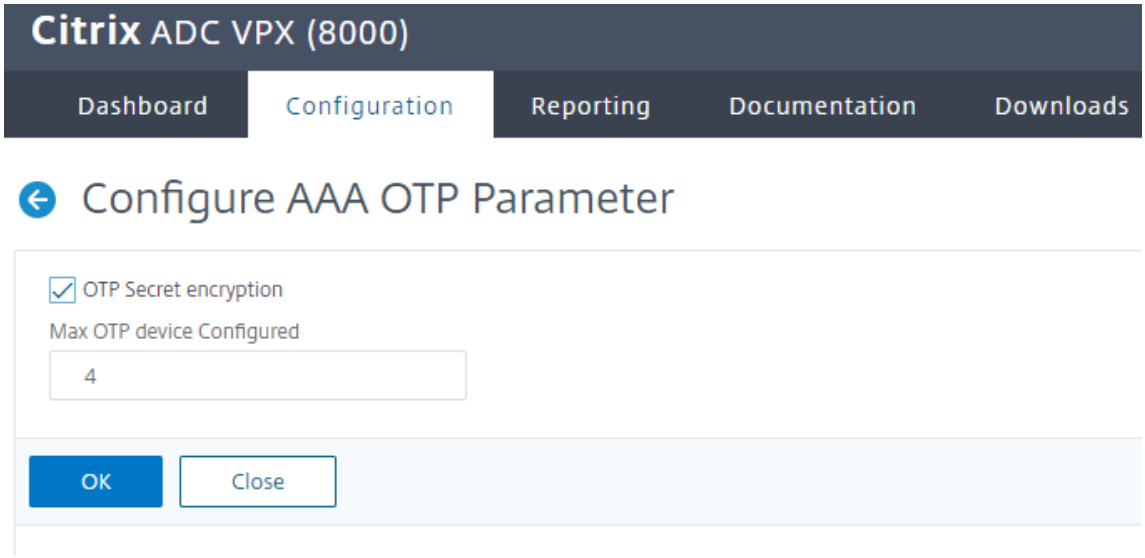
add authentication Policy ldap_policy -rule true -action ldap_action
<!--NeedCopy-->
```

Sample second factor configuration

```
add authentication policylabel set_otp -loginSchema LSCHEMA_INT
add authentication Policy set_otp -rule true -action test

bind authentication policylabel set_otp -policyName set_otp -priority 1
    -gotoPriorityExpression NEXT
bind authentication policylabel set_otp -policyName cascade_noauth -
    priority 2 -gotoPriorityExpression NEXT -nextFactor check_otp

```



```
add authentication Policy check_otp -rule "$test.valueExists(AAA.USER.SESSIONID)" -action NO_AUTHN
add authentication policylabel check_otp -loginSchema LSCHEMA_INT
bind authentication policylabel check_otp -policyName wpp -priority 1 -gotoPriorityExpression NEXT
bind authentication policylabel check_otp -policyName wpp_cascade_noauth -priority 2 -gotoPriorityExpression NEXT -nextFactor otp_verify

add authentication Policy wpp -rule true -action webAuth_POST
add authentication Policy wpp_cascade_noauth -rule true -action NO_AUTHN

add authentication Policy otp_verify -rule "AAA.LOGIN.PASSWORD.EQ($test[AAA.USER.SESSIONID])" -action NO_AUTHN
add authentication policylabel otp_verify -loginSchema onlyPassword
bind authentication policylabel otp_verify -policyName otp_verify -priority 1 -gotoPriorityExpression NEXT

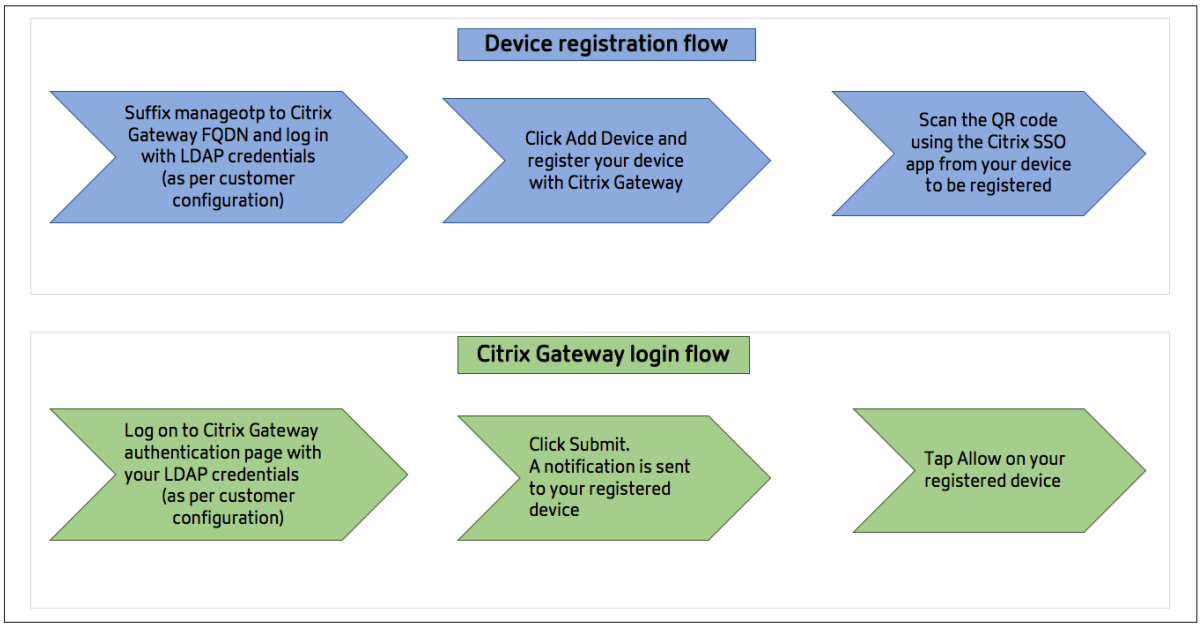
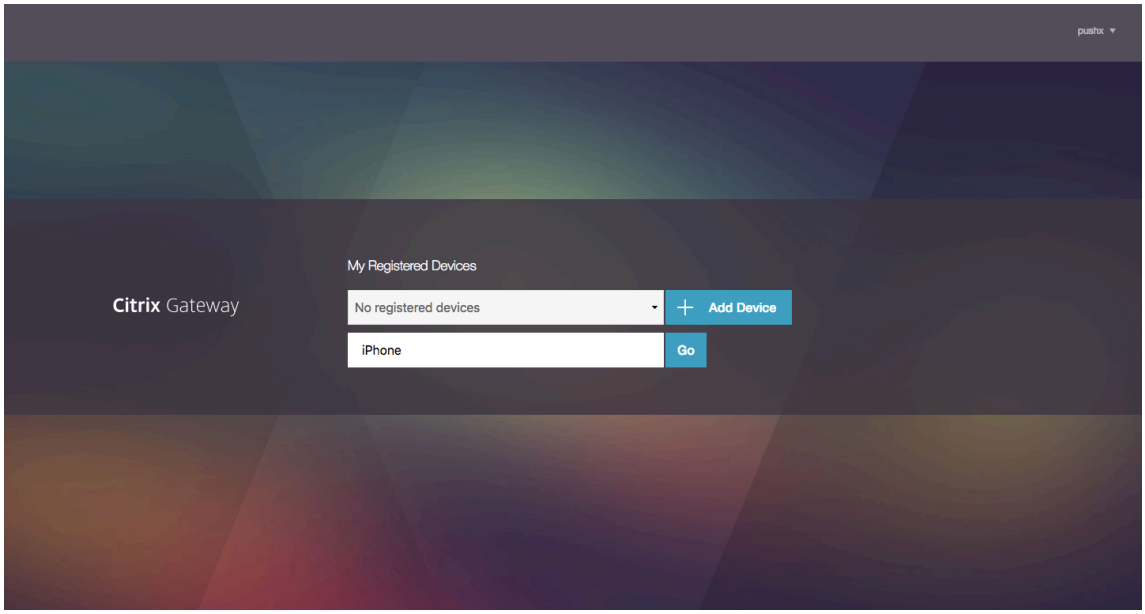
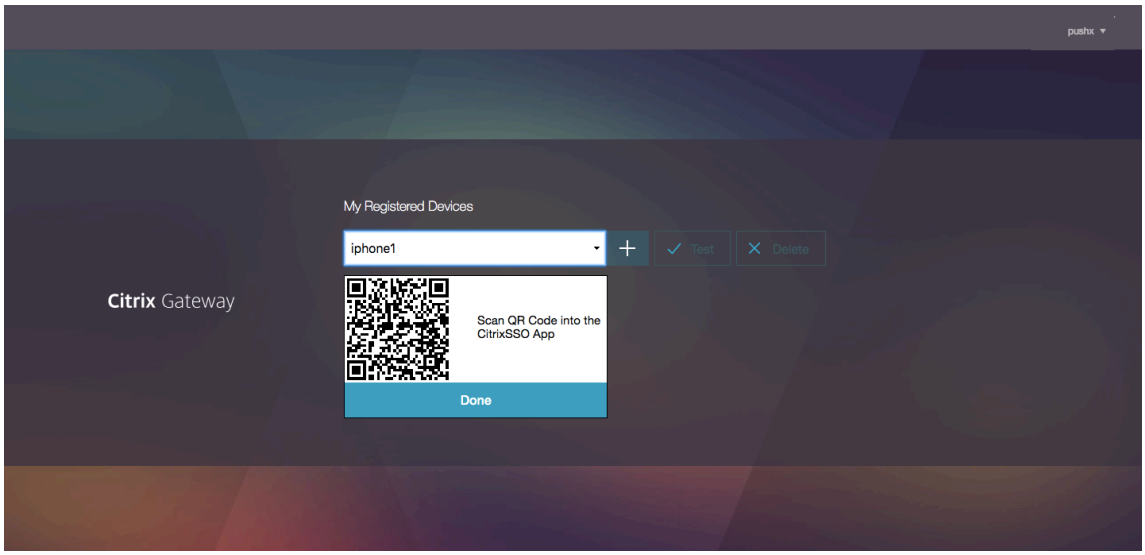
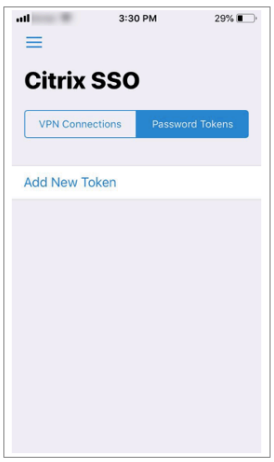
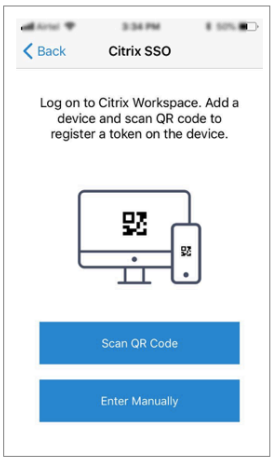
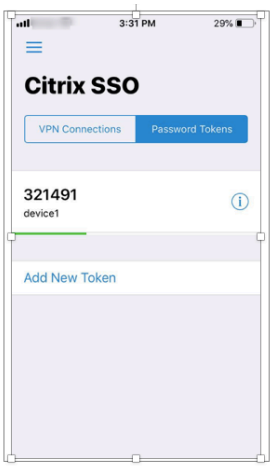
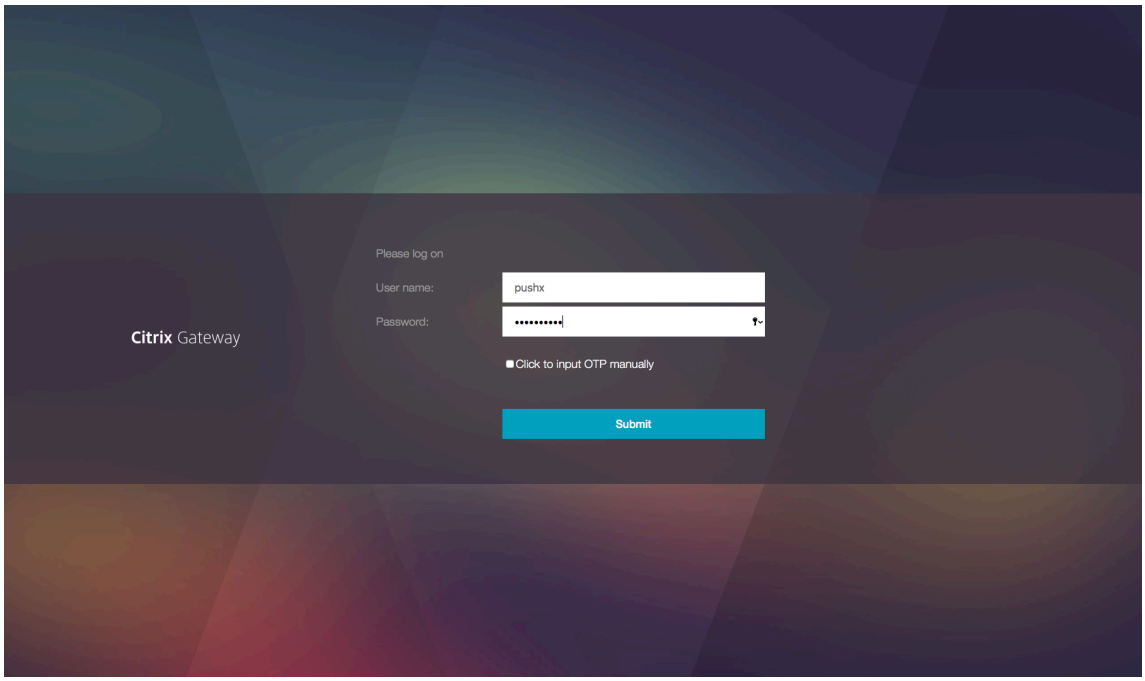
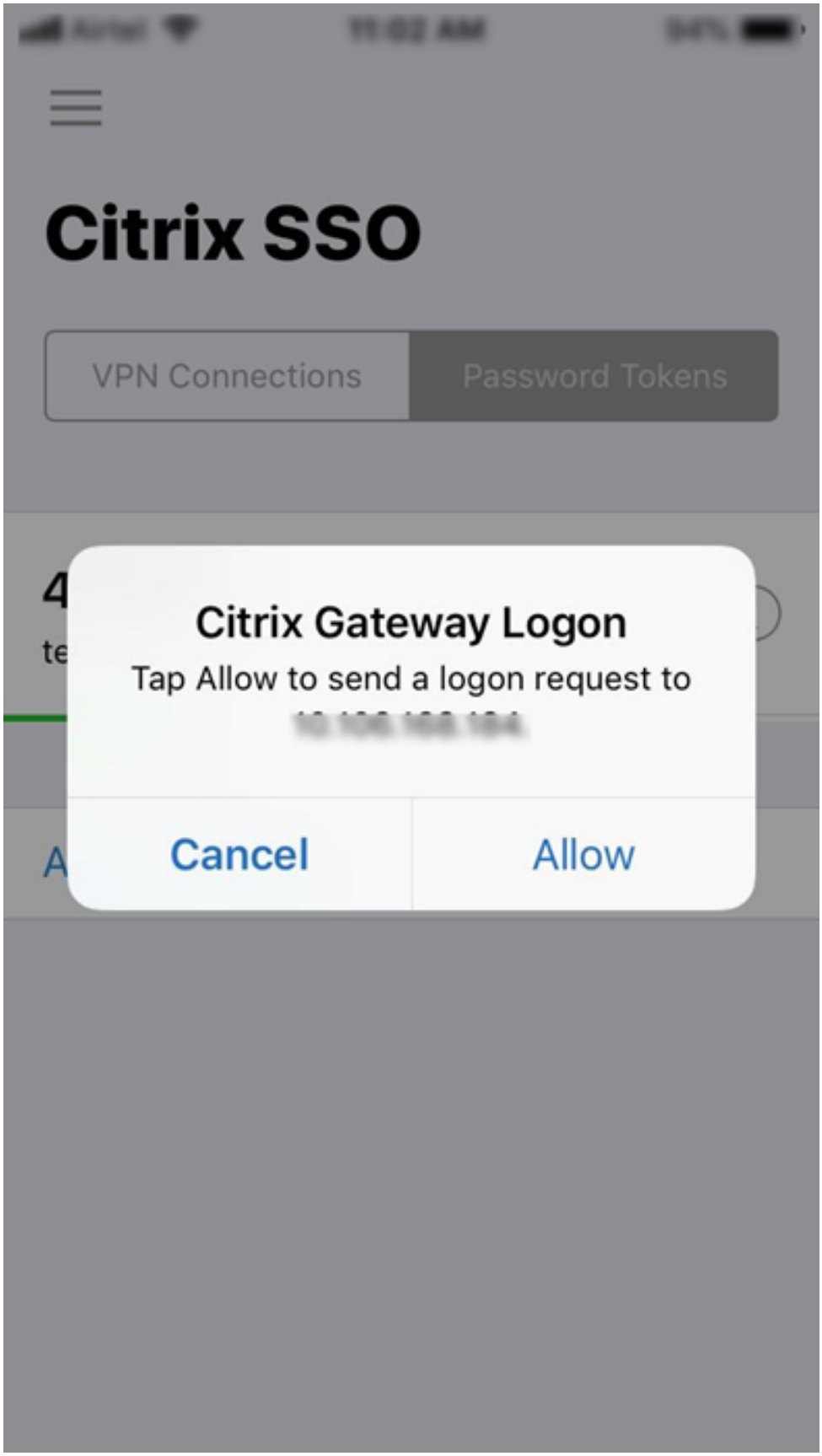
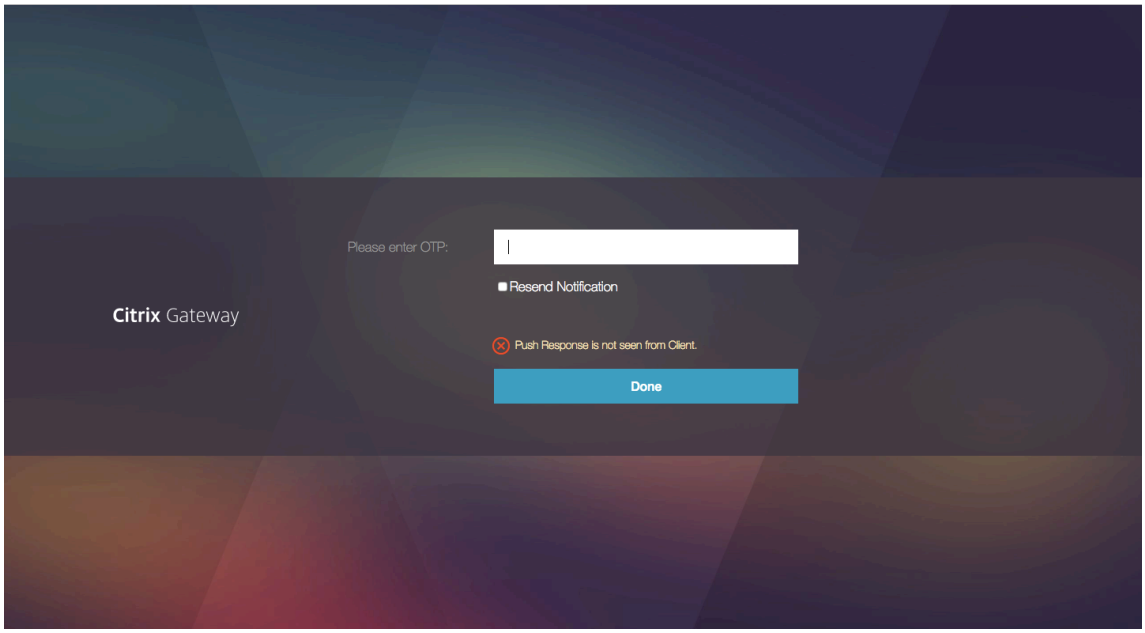
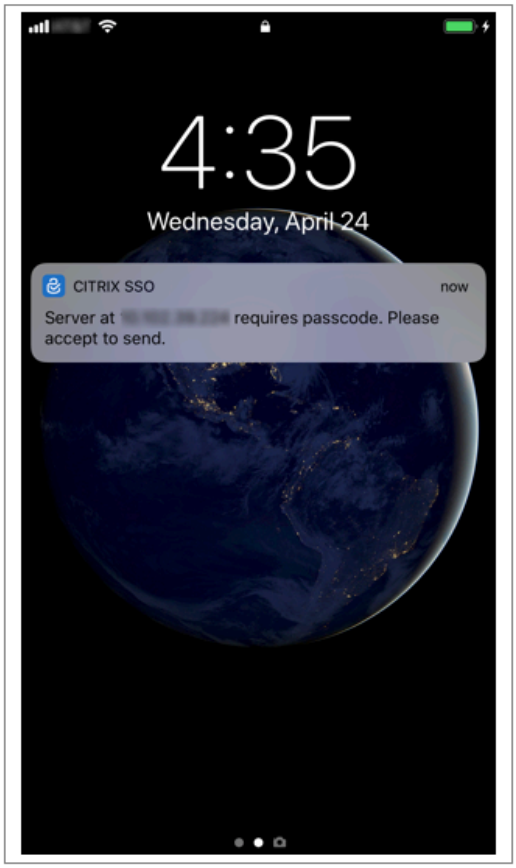
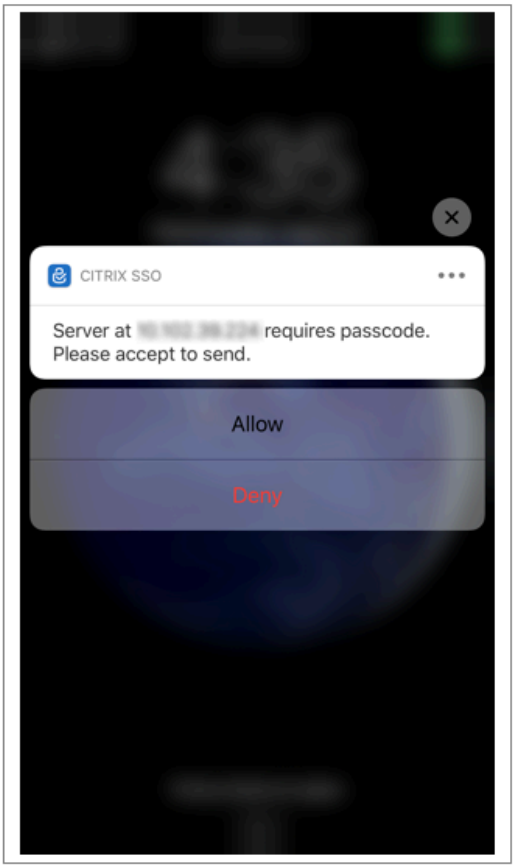
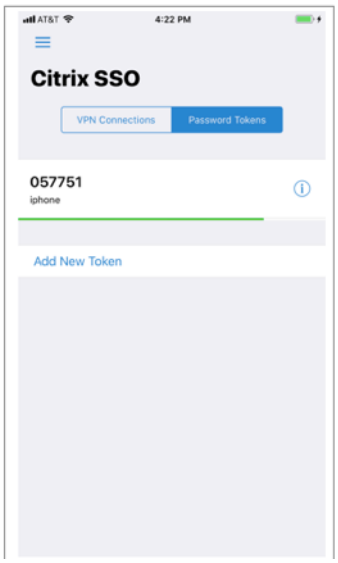
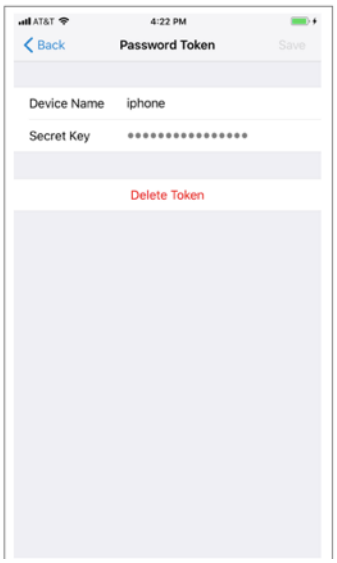
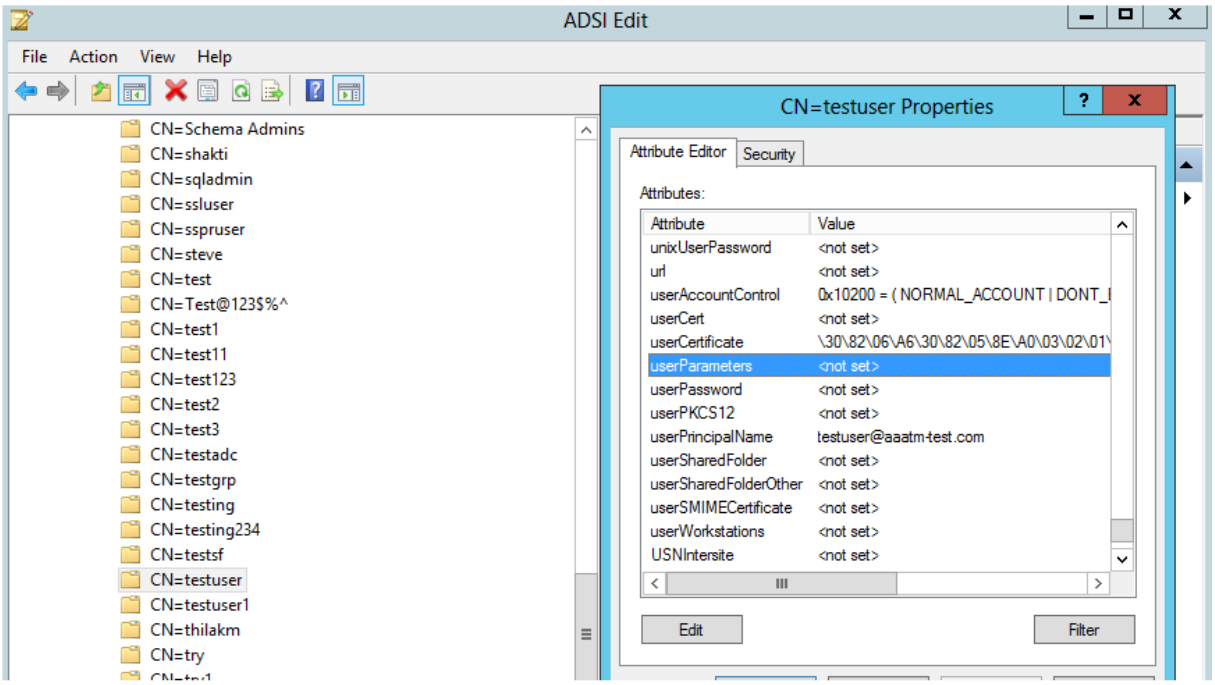
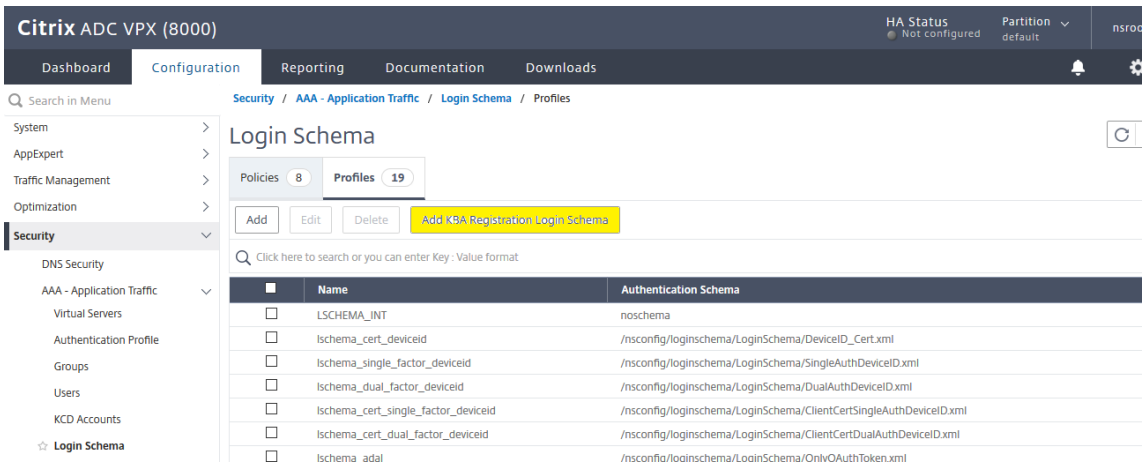
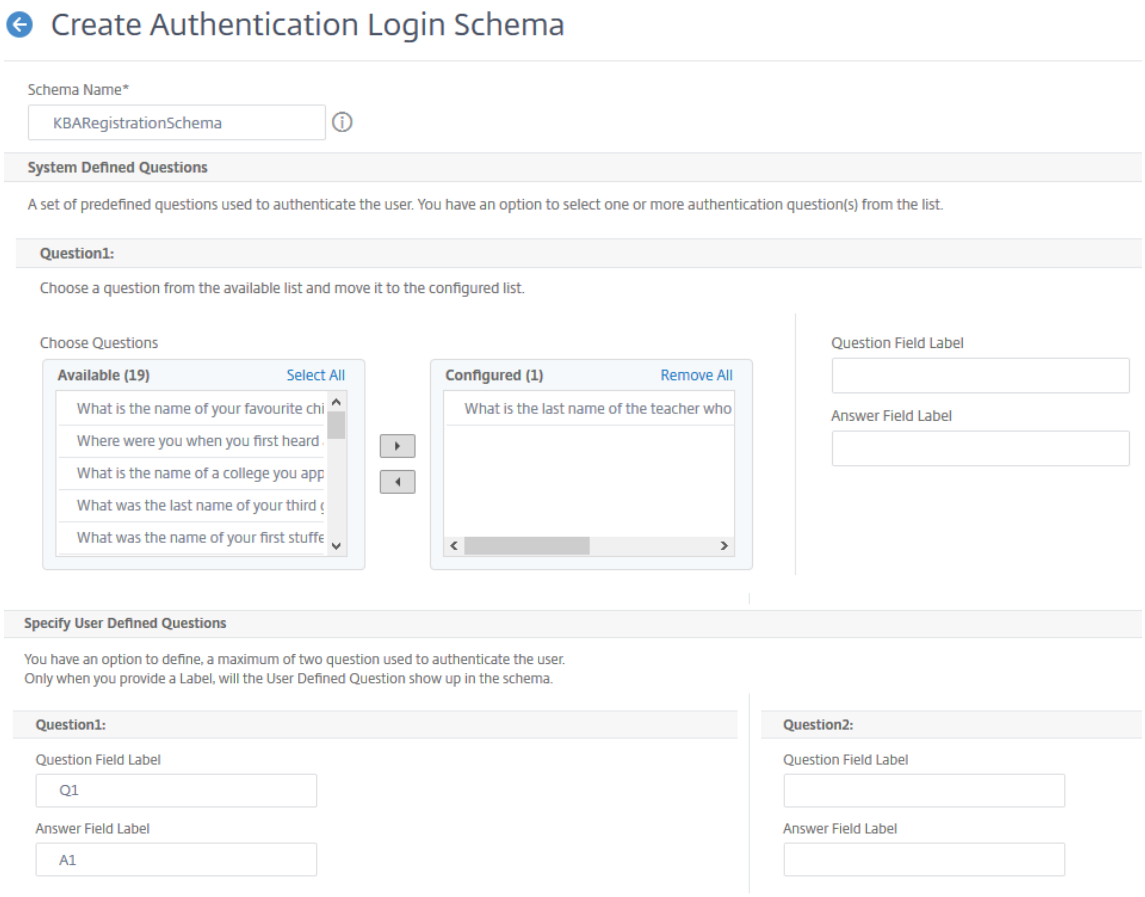
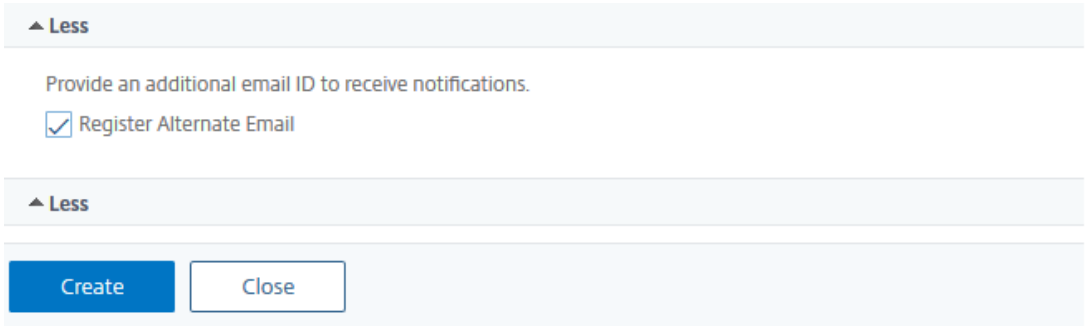
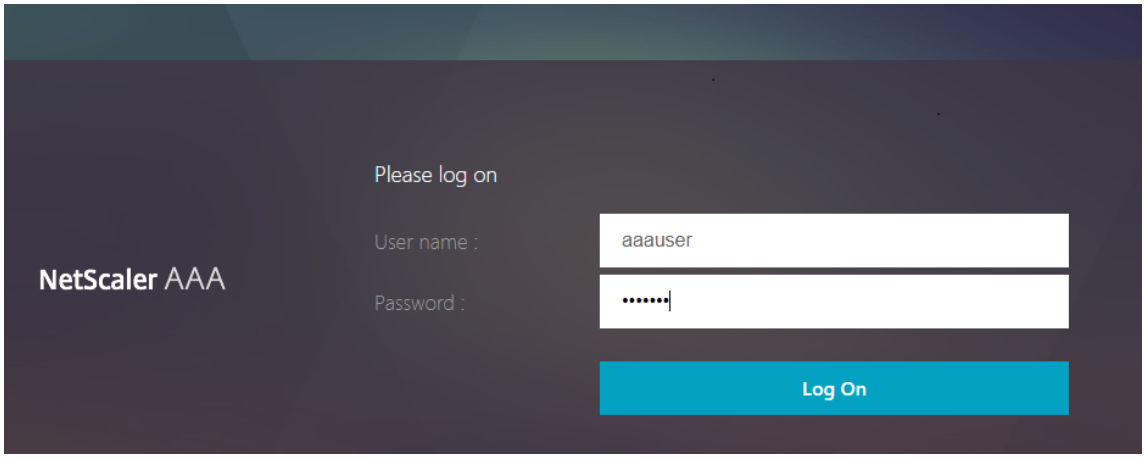
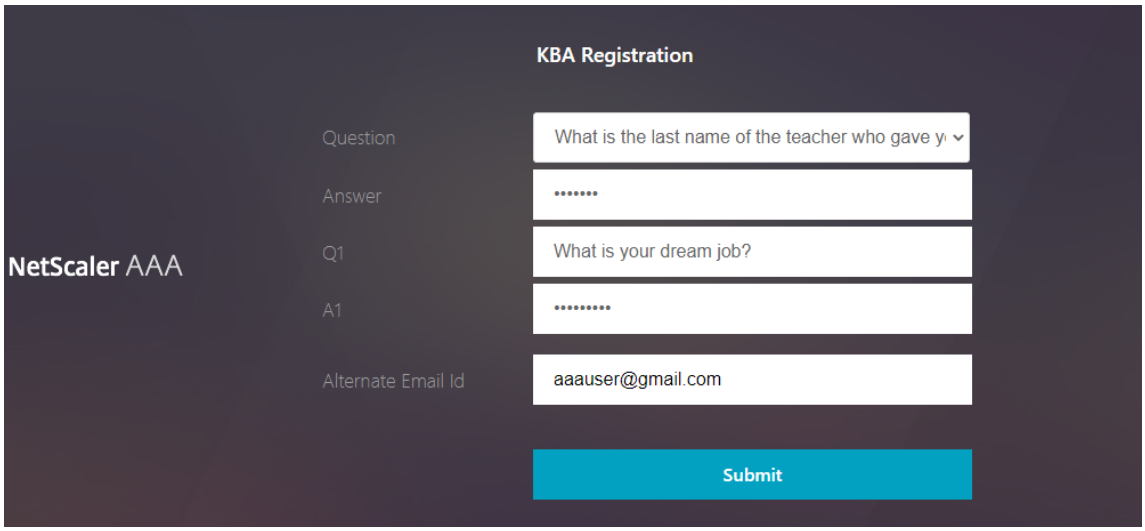
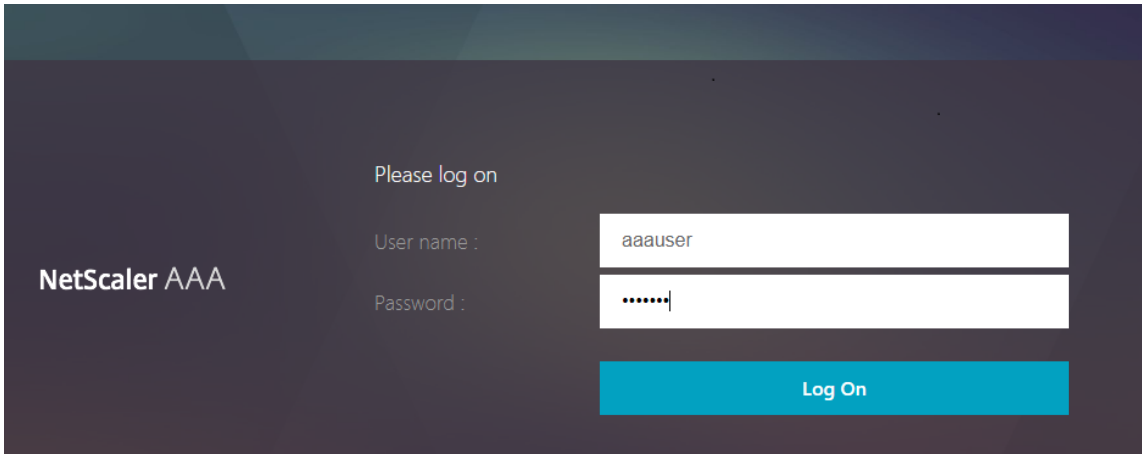
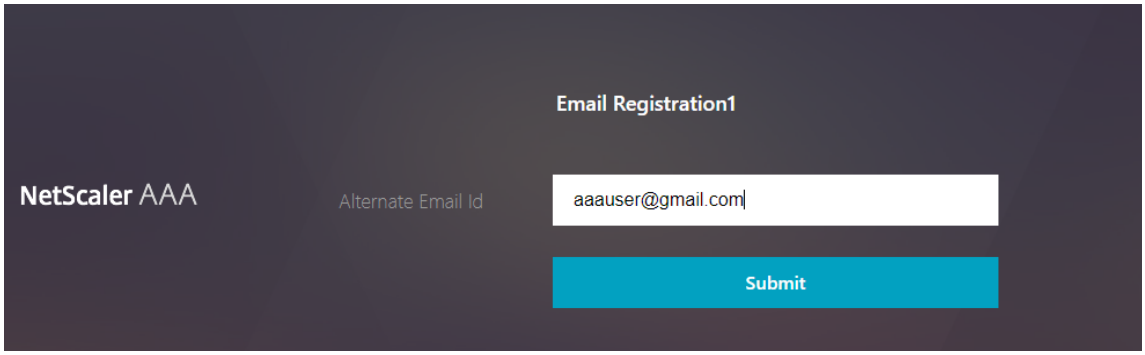
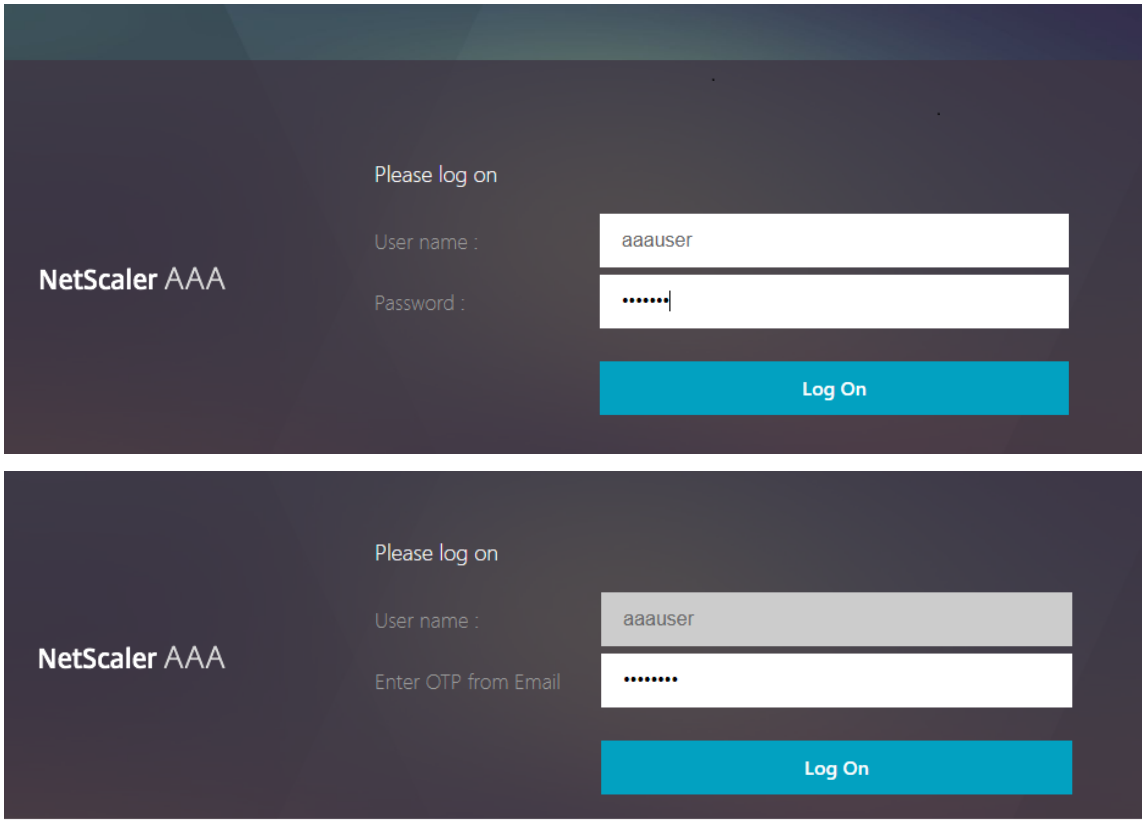
add authentication vserver avs SSL 10.106.40.121 443
bind authentication vserver avs -policy ldap_policy  -priority 1 -nextFactor set_otp -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

Forms based authentication

September 14, 2021

With Forms based authentication, a logon form is presented to the end-user. This type of authentication form supports both multifactor (nFactor) authentication and Classic authentication.



Ensure the following for the Forms based authentication to work:

- The load balancing virtual server must have authentication turned **ON**.
- 'authenticationHost' parameter must be specified to which the user must be redirected for authentication. The command for configuring the same is as follows:

```
set lb vs lb1 -authentication on  - authenticationhost aaavs-ip/
   fqdn
```

- Form based authentication is compatible with browser that supports HTML

The following steps walk through how the Forms based authentication works:

1. The client (browser) sends a GET request for a URL on the TM (load balancing/CS) virtual server.
2. The TM virtual server determines that the client has not been authenticated, and sends an HTTP 302 response to the client. The response contains a hidden script that causes the client to issue a GET request for /cgi/tm to the authentication virtual server.
3. The client sends GET /cgi/tm containing the target URL to the authentication virtual server.
4. The authentication virtual server sends out a redirect to the login page.
5. The user sends out its credentials to the authentication virtual server with a POST /doAuthentication.do. Authentication is done by the authentication virtual server.
6. If the credentials are correct, the authentication virtual server sends an HTTP 302 response to the cgi/selfauth url on the load balancing server with a one time token (OTP).
7. The load balancing server sends HTTP 302 to the client.
8. The client sends a GET request for their initial URL target URL along with a 32 byte cookie.



401 based authentication

September 14, 2021

With 401 based Authentication, the Citrix ADC appliance presents a pop-up dialog box to the end user.



Form based AAA-TM works on the redirect messages. However, some applications do not support redirects. In such applications, 401 authentication enabled AAA-TM is used.

Ensure the following for 401 Authentication Enabled AAA-TM to work:

- 'authnVsName' parameter value for the load balancing virtual server must be the name of the authentication virtual server to be used to authenticate users.
- 'authn401' parameter must be enabled. The command for configuring the same is as follows:

```
1  set lb vs lb1 -authn401 on -authnvsName <aaavs-name>
```

The following steps walk through how the 401 Authentication works:

1. User tries to access a particular URL using the load balancing virtual server.
2. The load balancing virtual server sends a 401 HTTP response back to the user indicating that authentication is required for the access.
3. The user sends its credentials to the load balancing virtual server in the authorization header.
4. The load balancing virtual server authenticates the user and then connects the user to the back end servers.



reCaptcha configuration for nFactor authentication

September 14, 2021

Citrix Gateway supports a new first class action 'captchaAction' that simplifies reCaptcha configuration. As reCaptcha is a first class action, it can be a factor of its own. You can inject reCaptcha anywhere in the nFactor flow.

Previously, you had to write custom WebAuth policies with changes to RfWeb UI as well. With the introduction of captchaAction, you do not have to modify the JavaScript.

> **Important**
>
> If reCaptcha is used along with username or password fields in the schema, submit button is disabled until reCaptcha is met.

reCaptcha configuration

reCaptcha configuration involves two parts.

1. Configuration on Google for registering reCaptcha.
2. Configuration on Citrix ADC appliance to use reCaptcha as part of login flow.

reCaptcha configuration on Google

Register a domain for reCaptcha at https://www.google.com/recaptcha/admin.

1. When you navigate to this page, the following screen appears.



> **Note**
>
> Use reCAPTCHA v2 only. Invisible reCAPTCHA is still in Beta.

2. After a domain is registered, the “SiteKey” and “SecretKey” are displayed.



> **Note**
>
> The “SiteKey” and “SecretKey” are grayed out for security reasons. “SecretKey” must be kept safe.

reCaptcha configuration on Citrix ADC appliance

reCaptcha configuration on Citrix ADC appliance can be divided into three parts:

- Display reCaptcha screen
- Post the reCaptcha response to Google server
- LDAP configuration is second factor for user logon (optional)

Display reCaptcha screen

The login form customization is done through the SingleAuthCaptcha.xml loginschema. This customization is specified at authentication virtual server and is sent to UI for rendering the login form. The built-in loginschema, SingleAuthCaptcha.xml, is at /nsconfig/loginSchema/LoginSchema directory on the Citrix ADC appliance.

> **Important**
>
> - Based on your use case and different schemas, you can modify the existing schema. For instance if you need only reCaptcha factor (without username or password) or dual authentication with reCaptcha.
> - If any custom modifications are performed or the file is renamed, Citrix recommends copying all loginSchemas from /nsconfig/loginschema/LoginSchema directory to parent directory, /nsconfig/loginschema.

To configure display of reCaptcha using CLI

- `add authentication loginSchema singleauthcaptcha -authenticationSchema /nsconfig/loginschema/SingleAuthCaptcha.xml`
- `add authentication loginSchemaPolicy singleauthcaptcha -rule true -action singleauthcaptcha`

- add authentication vserver auth SSL <IP> <Port>
- add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-key-file>
- bind ssl vserver auth -certkey vserver-cert
- bind authentication vserver auth -policy singleauthcaptcha -priority 5 -gotoPriorityExpression END

Post the reCaptcha response to Google server

After you have configured the reCaptcha that must be displayed to the users, admins post add the configuration to the Google server to verify the reCaptcha response from browser.

To verify reCaptcha response from the browser

- add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>
- add authentication policy myrecaptcha -rule **true** -action myrecaptcha
- bind authentication vserver auth -policy myrecaptcha -priority 1

The following commands are required to configure if AD authentication is desired. Else, you can ignore this step.

- add authentication ldapAction ldap-**new** -serverIP x.x.x.x -serverPort 636 -ldapBase "cn=users,dc=aaatm,dc=com"-ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup
- add authenticationpolicy ldap-**new** -rule **true** -action ldap-**new**

LDAP configuration is second factor for user logon (optional)

The LDAP authentication happens after reCaptcha, you add it to the second factor.

- add authentication policylabel second-factor
- bind authentication policylabel second-factor -policy ldap-**new** -priority 10
- bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor

Administrator needs to add appropriate virtual servers depending on whether load balancing virtual server or Citrix Gateway appliance is used for access. Administrator must configure the following command if load balancing virtual server is required:

- `add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com`

  nssp.aaatm.com – Resolves to authentication virtual server.

User validation of reCaptcha

Once you have configured all the steps mentioned in the previous sections, you must see the UI screenshots shown below.

1. Once the authentication virtual server loads the login page, the logon screen is displayed. **Log On** is disabled until reCaptcha is complete.



2. Select I'm not a robot option. The reCaptcha widget is displayed.



3. You are navigated through series of reCaptcha images, before the completion page is displayed.
4. Enter the AD credentials, select the **I'm not a robot** check box and click **Log On**. If authentication succeeds, you are redirected to the desired resource.



Notes

- If reCaptcha is used with AD authentication, Submit button for credentials is disabled until reCaptcha is complete.
- The reCaptcha happens in a factor of its own. Therefore, any subsequent validations like AD must happen in the 'nextfactor' of reCaptcha.

Native OTP support for authentication

September 14, 2021

Citrix ADC supports one-time passwords (OTPs) without having to use a third-party server. One-time password is a highly secure option for authenticating to secure servers as the number or passcode generated is random. Previously, specialized firms, such as RSA with specific devices that generate random numbers offered the OTPs. This system must be in constant communication with the client to generate a number expected by the server.

In addition to reducing capital and operating expenses, this feature enhances the administrator's control by keeping the entire configuration on the Citrix ADC appliance.

Note

Because third-party servers are no longer needed, the Citrix ADC administrator has to configure an interface to manage and validate user devices.

User must be registered with a Citrix ADC virtual server to use the OTP solution. Registration is required only once per unique device, and can be restricted to certain environments. Configuring and validation of a registered user is similar to configuring an extra authentication policy.

Advantages of having Native OTP support

- Reduces operating cost by eliminating the need to have an extra infrastructure on an authenticating server in addition to the Active Directory.
- Consolidates configuration only to Citrix ADC appliance thus offering great control to administrators.
- Eliminates the client's dependence on an extra authentication server for generating a number expected by clients.

Native OTP workflow

The native OTP solution is a two-fold process and the workflow is classified as the following:

- Device registration
- End user login

Important: You can skip the registration process if you are using third-party solutions or managing other devices apart from the Citrix ADC appliance. The final string that you add must be in the Citrix ADC specified format.

The following figure depicts the device registration flow to register a new device to receive OTP.

Note: The device registration can be done using any number of factors. The single factor (as specified in the previous figure) is used as an example to explain the device registration process.

The following figure depicts the verification of OTP through the registered device.

The following figure depicts the device registration and management flow.



The following figure depicts the end user flow for the Native OTP feature.



Prerequisites

To use the native OTP feature, make sure the following prerequisites are met.

- Citrix ADC feature release version is 12.0 build 51.24 and later.
- Advanced or Premium edition license is installed on Citrix Gateway.
- Citrix ADC is configured with management IP and the management console is accessible both using a browser and command line.
- Citrix ADC is configured with authentication, authorization, and auditing virtual server to authenticate users. For more information, see AAA
- Citrix ADC appliance is configured with Unified Gateway and the authentication, authorization, and auditing profile is assigned to the Gateway virtual server.
- Native OTP solution is restricted to nFactor authentication flow. Advanced policies are required to configure the solution. For more details, see Native OTP

Also ensure the following for Active Directory:

- A minimum attribute length of 256 characters.
- Attribute type must be 'DirectoryString' such as UserParameters. These attributes can hold string values.
- Attribute string type must be Unicode, if device name is in non-English characters.
- Citrix ADC LDAP administrator must have write access to the selected AD attribute.
- Citrix ADC appliance and client machine must be synced to a common Network Time Server.

Configure Native OTP using the GUI

The native OTP registration is not just a single factor authentication. The following sections help you to configure the single and second factor authentication.

Create Login Schema for first factor

1. Navigate to **Security AAA > Application Traffic > Login Schema**.
2. Go to **Profiles** and click **Add**.
3. On the **Create Authentication Login Schema** page, enter *lschema_single_auth_manage_otp* under the **Name** field and click **Edit** next to **noschema**.
4. Click the **LoginSchema** folder.
5. Scroll down to select **SingleAuth.xml** and click **Select**.
6. Click **Create**.
7. Click **Policies** and Click **Add**.
8. On the **Create Authentication Login Schema Policy** screen, enter the following values.

   Name: lpol_single_auth_manage_otp_by_url

   Profile: Select lschema_single_auth_manage_otp from the list.

   Rule: HTTP.REQ.COOKIE.VALUE(“NSC_TASS”).EQ(“`manageotp`”)

Configure authentication, authorization, and auditing virtual server

1. Navigate to **Security > AAA – Application Traffic > Authentication Virtual Servers**. Click to edit the existing virtual server. For more information, see AAA
2. Click the **+** icon next to **Login Schemas** under **Advanced Settings** in the right pane.
3. Select **No Login Schema**.
4. Click the arrow and select the **lpol_single_auth_manage_otp_by_url** Policy, click **Select**, and click **Bind**.
5. Scroll up and select **1 Authentication Policy** under **Advanced Authentication Policy**.
6. Right-click the **nFactor Policy** and select **Edit Binding**. Right-click the already configured nFactor Policy or refer to nFactor to create one and select Edit Binding.
7. Click the arrow under **Select Next Factor,** to select an existing configuration or click **Add** to create a new factor.

8. On the **Create Authentication PolicyLabel** screen, enter the following, and click **Continue**:

   Name: manage_otp_flow_label

   Login Schema: Lschema_Int

9. On the **Authentication PolicyLabel** screen, click **Add** to create a Policy.

   ```
   Create a policy for a normal LDAP server.
   ```

10. On the **Create Authentication Policy** screen, enter the following:

    Name: auth_pol_ldap_native_otp

11. Select the Action type as **LDAP** using the **Action Type** list.

12. In the **Action** field, click **Add** to create an action.

    ```
    Create the first LDAP action with authentication enabled to be used for
     single factor.
    ```

13. In the **Create Authentication LDAP server** page, select **Server IP** radio button, deselect the check box next to **Authentication**, enter the following values, and select **Test Connection**. The following is a sample configuration.

    Name: ldap_native_otp

    IP Address: 192.168.xx.xx

    Base DN: DC=training, DC=lab

    Administrator: Administrator@training.lab

    Password: `xxxxx`

    ```
    Create a policy for OTP .
    ```

14. On the **Create Authentication Policy** screen, enter the following:

    Name: auth_pol_ldap_otp_action

15. Select the Action type as **LDAP** using the **Action Type** list.

16. In the **Action** field, click **Add** to create an action.

    ```
    Create the second LDAP action to set OTP authenticator with OTP secret
    configuration and authentication unchecked.
    ```

17. In the **Create Authentication LDAP server** page, select **Server IP** radio button, deselect the check box next to **Authentication**, enter the following values, and select **Test Connection**. The following is a sample configuration.

    Name: ldap_otp_action

    IP Address: 192.168.xx.xx

Base DN: DC=training, DC=lab

Administrator: Administrator@training.lab

Password: xxxxx

18. Scroll down to the **Other Settings** section. Use the drop-down menu to select the following options.
Server Logon Name Attribute as **New** and type **userprincipalname**.
19. Use the drop-down menu to select **SSO Name Attribute** as **New** and type **userprincipalname**.
20. Enter “UserParameters” in the **OTP Secret** field and click **More**.
21. Enter the following Attributes.

Attribute 1 = mail
Attribute 2 = objectGUID
Attribute 3 = immutableID

22. Click **OK**.
23. On the **Create Authentication Policy** page, set the Expression to **true** and click **Create**.
24. On the **Create Authentication Policylabel** page, click **Bind**, and click **Done.**
25. On the **Policy Binding** page, click **Bind**.
26. On the **Authentication policy** page, click **Close** and click **Done**.

```
Create OTP for OTP verification.
```

27. On the **Create Authentication Policy** screen, enter the following:

Name: auth_pol_ldap_otp_verify

28. Select the Action type as **LDAP** using the **Action Type** list.
29. In the **Action** field, click **Add** to create an action.

```
Create the third LDAP action to verify OTP.
```

30. In the **Create Authentication LDAP server** page, select **Server IP** radio button, deselect the check box next to **Authentication**, enter the following values, and select **Test Connection**. The following is a sample configuration.

Name: ldap_verify_otp

IP Address: 192.168.xx.xx

Base DN: DC=training, DC=lab

Administrator: Administrator@training.lab

Password: xxxxx

31. Scroll down to the **Other Settings** section. Use the drop-down menu to select the following options.
    Server Logon Name Attribute as **New** and type **userprincipalname**.
32. Use the drop-down menu to select **SSO Name Attribute** as **New** and type **userprincipalname**.
33. Enter “UserParameters” in the **OTP Secret** field and click **More**.
34. Enter the following Attributes.

    Attribute 1 = mail
    Attribute 2 = objectGUID
    Attribute 3 = immutableID
35. Click **OK**.
36. On the **Create Authentication Policy** page, set the Expression to **true** and click **Create**.
37. On the **Create Authentication Policylabel** page, click **Bind**, and click **Done.**
38. On the **Policy Binding** page, click **Bind**.
39. On the **Authentication policy** page, click **Close** and click **Done**.

You probably don’t already have an Advanced Authentication Policy for your normal LDAP server.
Change the Action Type to LDAP.
Select your normal LDAP server, which is the one that has Authentication enabled.
Enter true as the expression. This uses Default Syntax instead of Classic Syntax.
Click Create.

> **Note**
>
> The authentication virtual server must be bound to the RFWebUI portal theme. Bind a server certificate to the server. The server IP ‘1.2.3.5’ must have a corresponding FQDN that is, otpauth.server.com, for later use.

Create login schema for second factor OTP

1. Navigate to **Security > AAA-Application Traffic > Virtual Servers**. Select the virtual server to be edited.
2. Scroll down and select **1 Login Schema**.
3. Click **Add Binding**.
4. Under the **Policy Binding** section, click **Add** to add a policy.
5. On the **Create Authentication Login Schema Policy** page, enter Name as OTP, and click **Add** to create a profile.
6. On the **Create Authentication Login Schema** page, enter Name as OTP, and click the pencil icon next to **noschema**.

7. Click the **LoginSchema** folder, select **DualAuthManageOTP.xml**, and then click **Select**.
8. Click **More** and scroll down.
9. In the **Password Credential Index** field, enter 1. This causes nFactor to save the user's password into AAA Attribute #1, which can be used later in a Traffic Policy to Single Sign-on to StoreFront. If you don't do this, then Citrix Gateway tries to use the Passcode to authenticate to StoreFront, which does not work.
10. Click **Create**.
11. In the **Rule** section, enter **True**. Click **Create**.
12. Click **Bind**.
13. Notice the two factors of authentication. Click **Close** and click **Done**.

Traffic Policy for Single Sign-on

1. Navigate to **Citrix Gateway > Policies > Traffic**
2. On the **Traffic Profiles** tab, click **Add**.
3. Enter a name for the traffic profile for OTP.
4. Scroll down, in the SSO Password Expression box, enter the following, and click **Create**. This is where we use the login schema password attribute specified for the second factor OTP.

   ```
   http.REQ.USER.ATTRIBUTE(1)
   ```

5. On **Traffic Policies** tab, click **Add**.
6. In the **Name** field, enter a name for the traffic policy.
7. In the **Request Profile** field, select the traffic profile you created.
8. In the Expression box, enter **True**. If your Citrix Gateway virtual server allows full VPN, change the expression to the following.

   ```
   http.req.method.eq(post)||http.req.method.eq(get)&& false
   ```

9. Click **Create**.

Configure content switching policy for manage OTP

The following configurations are required if you are using Unified Gateway.

1. Navigate to **Traffic Management > Content Switching > Policies**. Select the content switching policy, right click, and select **Edit**.
2. Edit the expression to evaluate the following OR statement and click **OK**:

is_vpn_url||HTTP.REQ.URL.CONTAINS("manageotp")

Configure Native OTP using the CLI

You must have the following information to configure the OTP device management page:

- IP assigned to authentication virtual server
- FQDN corresponding to the assigned IP
- Server certificate for authentication virtual server

> **Note:** Native OTP is a web-based solution only.

To configure the OTP device registration and management page

Create authentication virtual server

```
add authentication vserver authvs SSL 1.2.3.5 443
bind authentication vserver authvs -portaltheme RFWebUI
bind ssl vserver authvs -certkeyname otpauthcert
```

> **Note:** The authentication virtual server must be bound to the RFWebUI portal theme. Bind a server certificate to the server. The server IP '1.2.3.5' must have a corresponding FQDN that is, otpauth.server.com, for later use.

To create LDAP logon action

```
add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
    -  serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWO> -ldapLoginName <USER FORMAT>
```

Example:

```
add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4 -
    serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName userprincipalname
```

To add authentication policy for LDAP Logon

```
add authentication Policy auth_pol_ldap_logon -rule true -action
    ldap_logon_action
```


To present UI via LoginSchema

Show user name field and password field to users upon logon

```
add authentication loginSchema lschema_single_auth_manage_otp -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/
    SingleAuthManageOTP.xml"
```

Display device registration and management page

Citrix recommends two ways of displaying the device registration and management screen: URL or host name.

- **Using URL**

  When the URL contains '/manageotp'

  - `add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_url -rule "http.req.cookie.value("NSC_TASS").contains("manageotp")"-action lschema_single_auth_manage_otp`
  - `bind authentication vserver authvs -policy lpol_single_auth_manage_otp_by_url -priority 10 -gotoPriorityExpression END`

- **Using hostname**

  When the host name is 'alt.server.com'

  - `add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_host -rule "http.req.header("host").eq("alt.server.com")"-action lschema_single_auth_manage_otp`
  - `bind authentication vserver authvs -policy lpol_single_auth_manage_otp_by_host -priority 20 -gotoPriorityExpression END`

To configure the user login page using the CLI

You must have the following information to configure the User Logon page:

- IP for a load balancing virtual server
- Corresponding FQDN for the load balancing virtual server
- Server certificate for the load balancing virtual server

  bind ssl vserver lbvs_https -certkeyname lbvs_server_cert

Back-end service in load balancing is represented as follows:


```
add service iis_backendsso_server_com 1.2.3.210 HTTP 80
bind lb vserver lbvs_https iis_backendsso_server_com
```

To create OTP passcode validation action

```
add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
    -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
    authentication DISABLED -OTPSecret <LDAP ATTRIBUTE>`
```

Example:

```
add authentication ldapAction ldap_otp_action -serverIP 1.2.3.4 -
    serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName userprincipalname -authentication DISABLED -OTPSecret
    userParameters
```

> **Important:** The difference between the LDAP logon and OTP action is the need to disable the authentication and introduce a new parameter `OTPSecret`. Do not use the AD attribute value.

To add authentication policy for OTP passcode validation

```
add authentication Policy auth_pol_otp_validation -rule true -action
    ldap_otp_action
```

To present the two-factor authentication through LoginSchema

Add the UI for two factor authentication.

```
add authentication loginSchema lscheme_dual_factor -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/DualAuth.xml
    "
add authentication loginSchemaPolicy lpol_dual_factor -rule true -
    action lscheme_dual_factor
```

To create passcode validation factor via the policy label

Create a manage OTP flow policy label for the next factor (first factor is LDAP logon)


```
add authentication loginSchema lschema_noschema -authenticationSchema
    noschema
add authentication policylabel manage_otp_flow_label -loginSchema
    lschema_noschema
```

To bind the OTP policy to the policy label

```
bind authentication policylabel manage_otp_flow_label -policyName
    auth_pol_otp_validation -priority 10 -gotoPriorityExpression NEXT
```

To bind the UI flow

Bind the LDAP logon followed by the OTP validation with the authentication virtual server.

```
bind authentication vserver authvs -policy auth_pol_ldap_logon -
    priority 10 -nextFactor  manage_otp_flow_label -
    gotoPriorityExpression NEXT
bind authentication vserver authvs -policy lpol_dual_factor -priority
    30 -gotoPriorityExpression END
```

Register your device with Citrix ADC

1. Navigate to your Citrix ADC FQDN (first public facing IP), with a /manageotp suffix. For example, https://otpauth.server.com/manageotp Login with user credentials.
2. Click the **+** icon to add a device.
3. Enter a device name and press **Go**. A barcode appears on the screen.
4. Click **Begin Setup** and then click **Scan Barcode**.
5. Hover the device camera over the QR code. You can optionally enter the 16 digit code.

   > **Note**: The displayed QR code is valid for 3 minutes.

6. Upon successful scan, you are presented with a 6 digit time sensitive code that can be used to log in.
7. To test, click **Done** on the QR screen, then click the green check mark on the right.
8. Select your device from the drop-down menu and enter the code from Google Authenticator (must be blue, not red) and click **Go**.
9. Make sure to log out using the drop-down menu at the top right corner of the page.

Log in to Citrix ADC using the OTP

1. Navigate to your first public facing URL and enter your OTP from Google Authenticator to log on.
2. Authenticate to the Citrix ADC splash page.

Store OTP secret data in an encrypted format

September 14, 2021

Starting from Citrix ADC release 13.0 build 41.20, the OTP secret data can be stored in an encrypted format instead of plain text.

Previously, the Citrix ADC appliance stored OTP secret as a plain text in AD. Storing OTP secret in plain text poses a security threat as a malicious attacker or an admin might exploit the data by viewing the shared secret of other users.

The encryption parameter enables encryption of OTP secret in AD. When you register a new device with Citrix ADC version 13.0 build 41.20, and enable the encryption parameter, the OTP secret is stored in an encrypted format, by default. However, if the encryption parameter is disabled, the OTP secret is stored in plain text format.

For devices registered prior to 13.0 build 41.20, you must perform the following as a best practice:

1. Upgrade the 13.0 Citrix ADC appliance to 13.0 build 41.20.
2. Enable the encryption parameter on the appliance.
3. Use the OTP secret migration tool to migrate OTP secret data from plain text format to encrypted format.

For details about the OTP secret migration tool, see OTP encryption tool.

> **Important**
>
> Citrix recommends you as an admin to ensure the following criteria is met:
>
> - A new certificate must be configured to encrypt OTP secrets if you are not using KBA as part of self-service password reset feature.
>   - To bind the certificate to VPN global, you can use the following command:
>
>     ```
>     bind vpn global -userDataEncryptionKey <certificate name>
>     ```
>
> - If you are already using a certificate to encrypt KBA, you can use the same certificate to encrypt OTP secrets.

To enable OTP encryption data by using the CLI

At the command prompt, type:

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

Example

```
set aaa otpparameter -encryption ON
```

To configure the OTP encryption by using the GUI

1. Navigate to **Security > AAA – Application Traffic** and click **Change authentication AAA OTP Parameter** under **Authentication Settings** section.
2. On the **Configure AAA OTP Parameter** page, select **OTP Secret encryption**.
3. Click OK.

Configuring the number of end-user devices for receiving OTP notifications

Administrators can now configure the number of devices that an end user can register to receive OTP notification or authentication.

To configure the number of devices in OTP by using the CLI

At the command prompt, type:

```
set aaa otpparameter [-maxOTPDevices <positive_integer>]
```

Example

```
set aaa otpparameter -maxOTPDevices 4
```

To configure the number of devices by using the GUI

1. Navigate to **Security > AAA – Application Traffic**, click **Change authentication AAA OTP Parameter** under **Authentication Settings** section.
2. On the **Configure AAA OTP Parameter** page, enter the value for **Max OTP device Configured**.
3. Click **OK**.



OTP encryption tool

September 14, 2021

Starting from Citrix ADC release 13.0 build 41.20, the OTP secret data is stored in an encrypted format instead of plain text for enhanced security. Storing of OTP secret in encrypted format is automatic and does not require manual intervention.

Previously, the Citrix ADC appliance stored OTP secret as a plain text in the active directory. Storing OTP secret in a plain text format posed a security threat as a malicious attacker or an admin could exploit the data by viewing the shared secret of other users.

The OTP encryption tool provides the following advantages:

- Does not result in any data loss even if you have old devices that are using old format (plain text).
- The backward compatibility support with old Citrix Gateway versions, helps to integrate and work with the existing devices, along with the new device.
- The OTP encryption tool helps admins migrate all the OTP secret data of all users at once.

> **Note**
>
> OTP encryption tool does not encrypt or decrypt KBA registration or Email registration data.

Uses of OTP encryption tool

The OTP encryption tool can be used for the following:

- **Encryption**. Store the OTP secret in encrypted format. The tool extracts the OTP data of the devices registered with Citrix ADC, and then converts the OTP data in plain text format to encrypted format.
- **Decryption**. Revert the OTP secret to the plain text format.
- **Update certificates**. Administrators can update the certificate to a new certificate at any time. Admins can use the tool to enter the new certificate and update all the entries with the new certificate data. The certificate path must either be an absolute path or a relative path.

> **Important**
>
> - You must enable the encryption parameter in the Citrix ADC appliance to use the OTP encryption tool.
> - For devices registered with Citrix ADC prior to build 41.20, you must perform the following:
>   - Upgrade the 13.0 Citrix ADC appliance to 13.0 build 41.20.
>   - Enable the encryption parameter on the appliance.
>   - Use the OTP Secret migration tool to migrate OTP secret data from plain text format to encrypted format.

OTP secret data in plain text format

Example:

```
##@devicename=<16 or more bytes>&tag=<64bytes>&,
```

As you can see, the starting pattern for old format is always “#@” and ending pattern is always “&”. All the data between “devicename=” and end pattern, constitutes user OTP data.

OTP secret data in encrypted format

The new encrypted format of OTP data is of the following format:

Example:

```
{

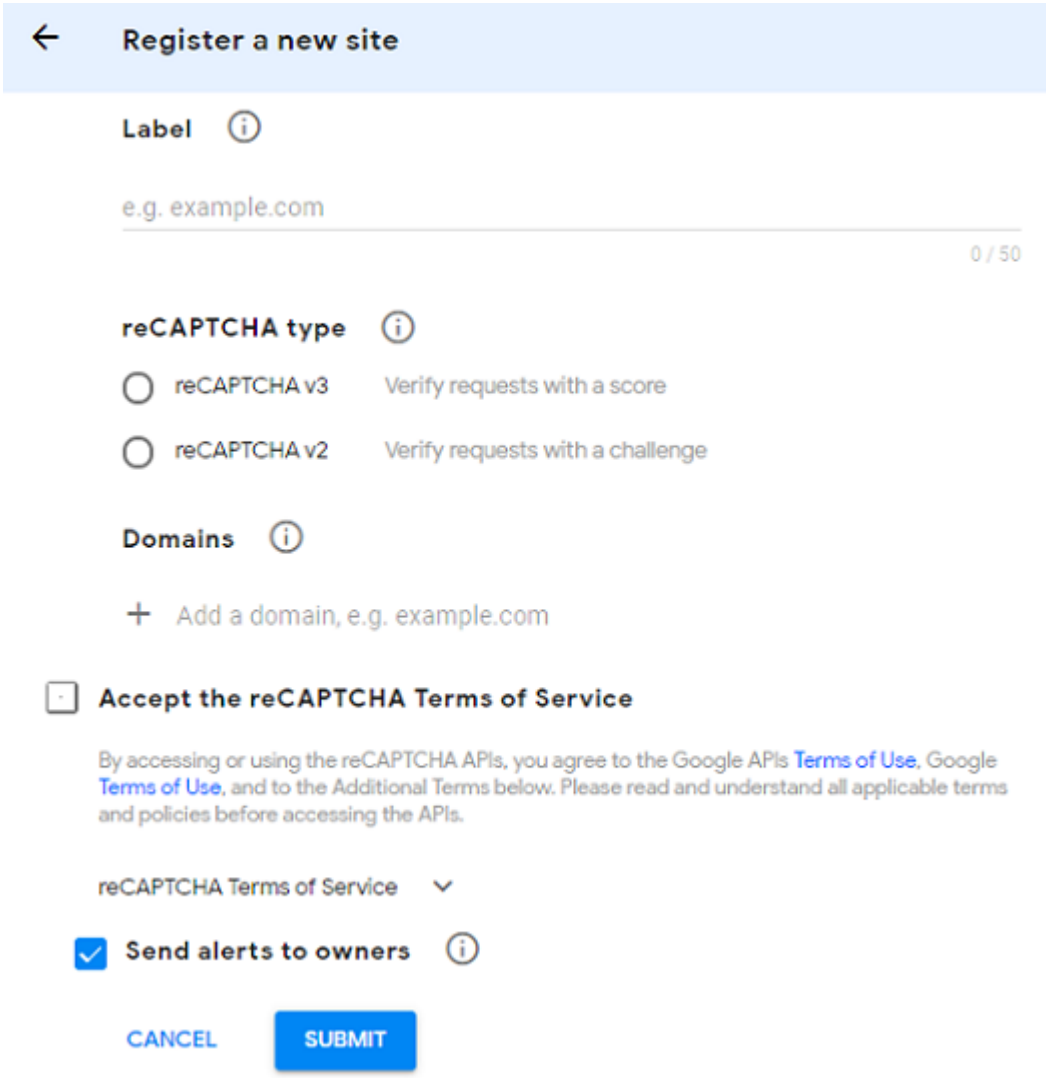
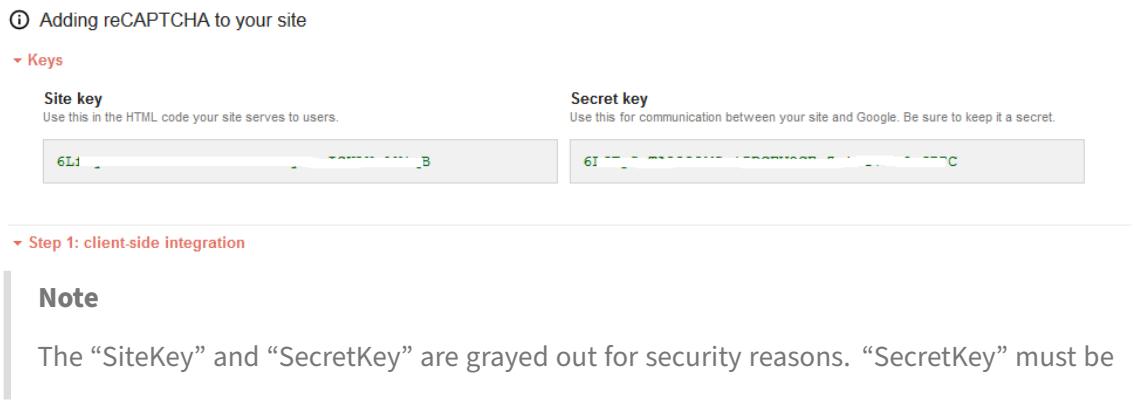
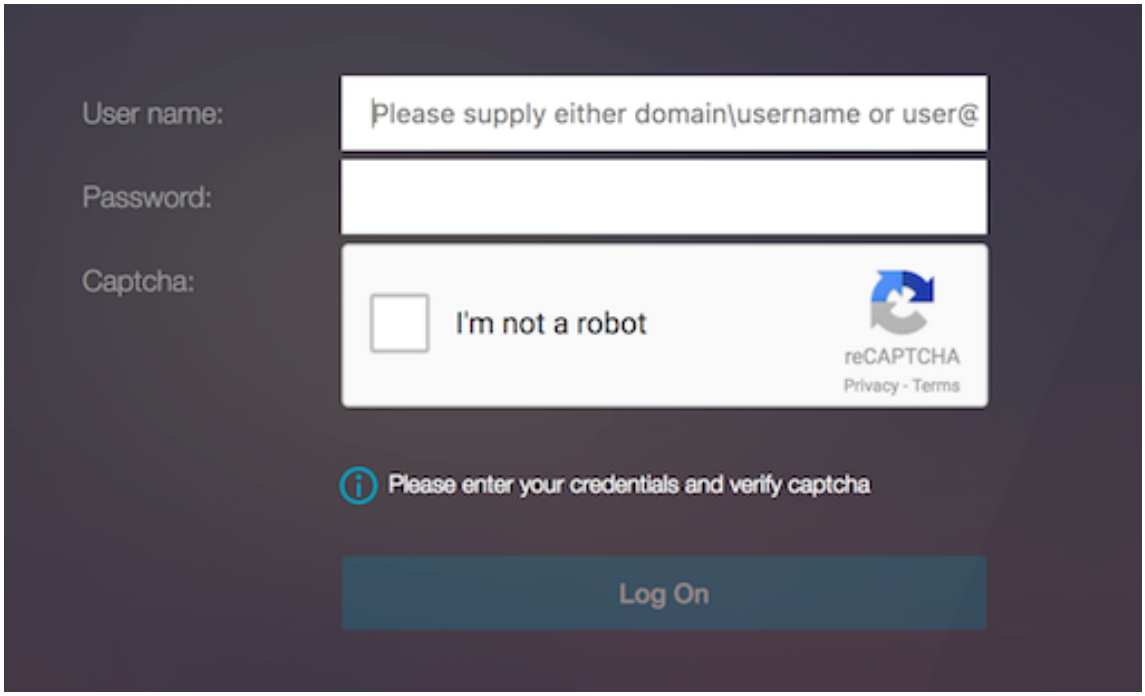
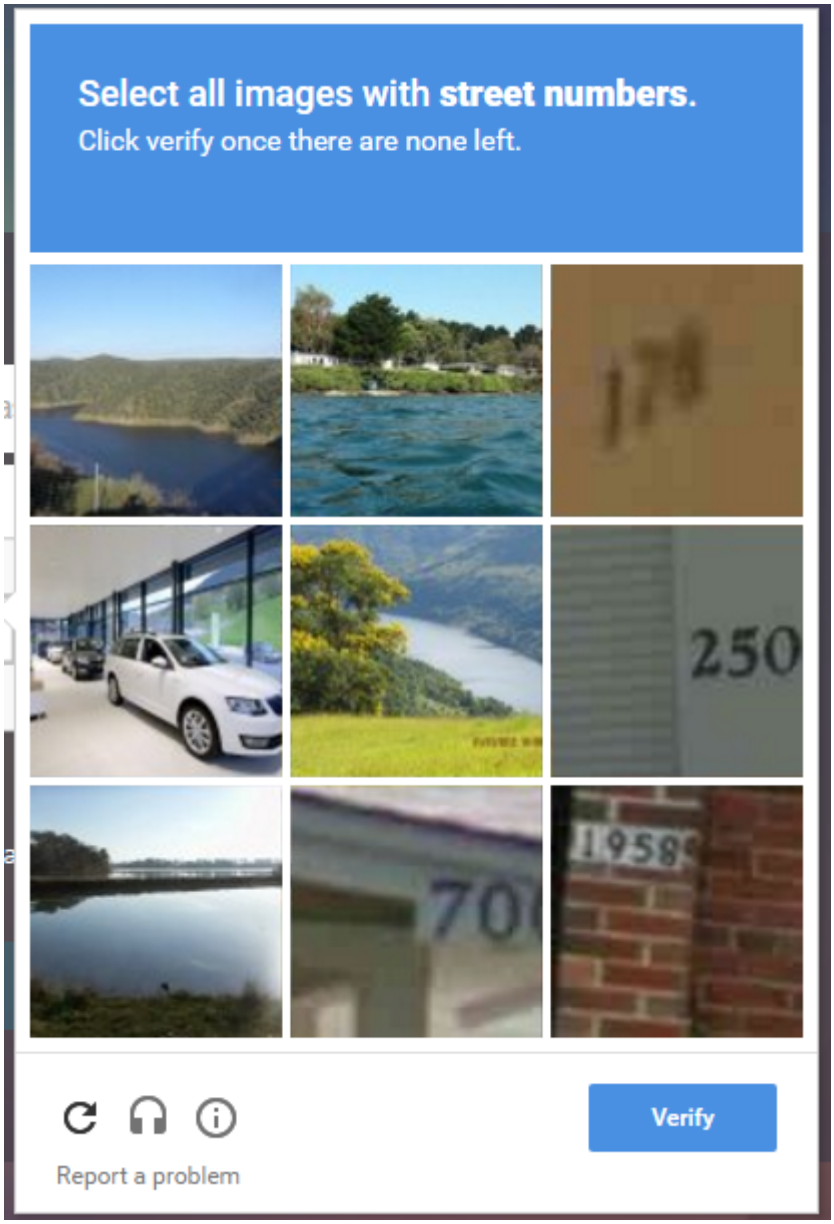
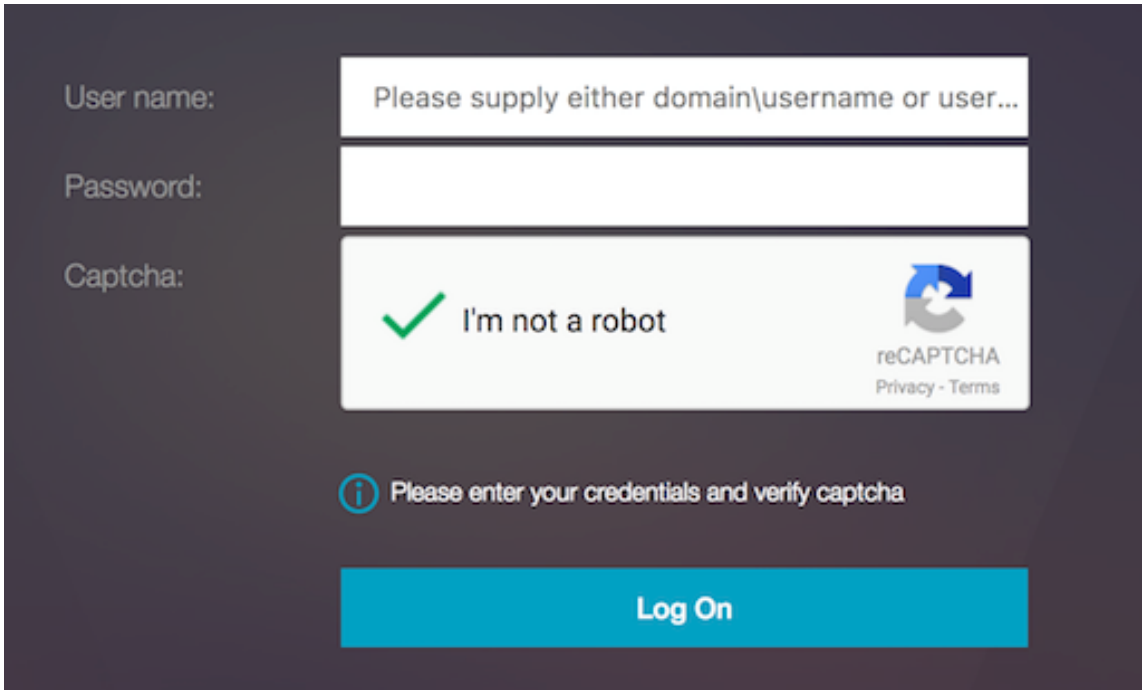
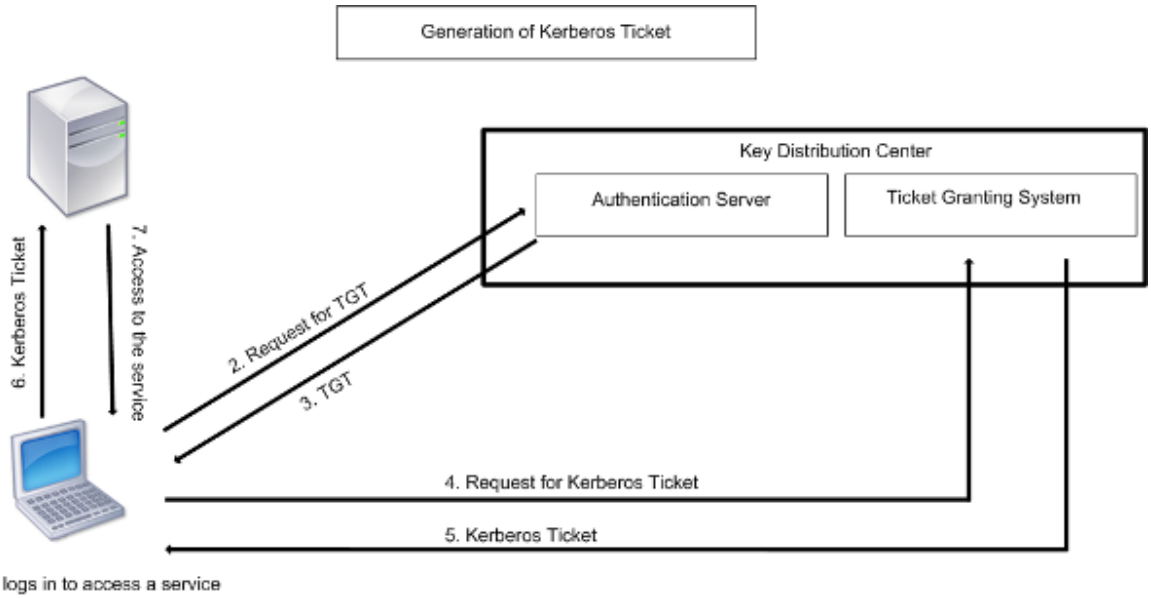
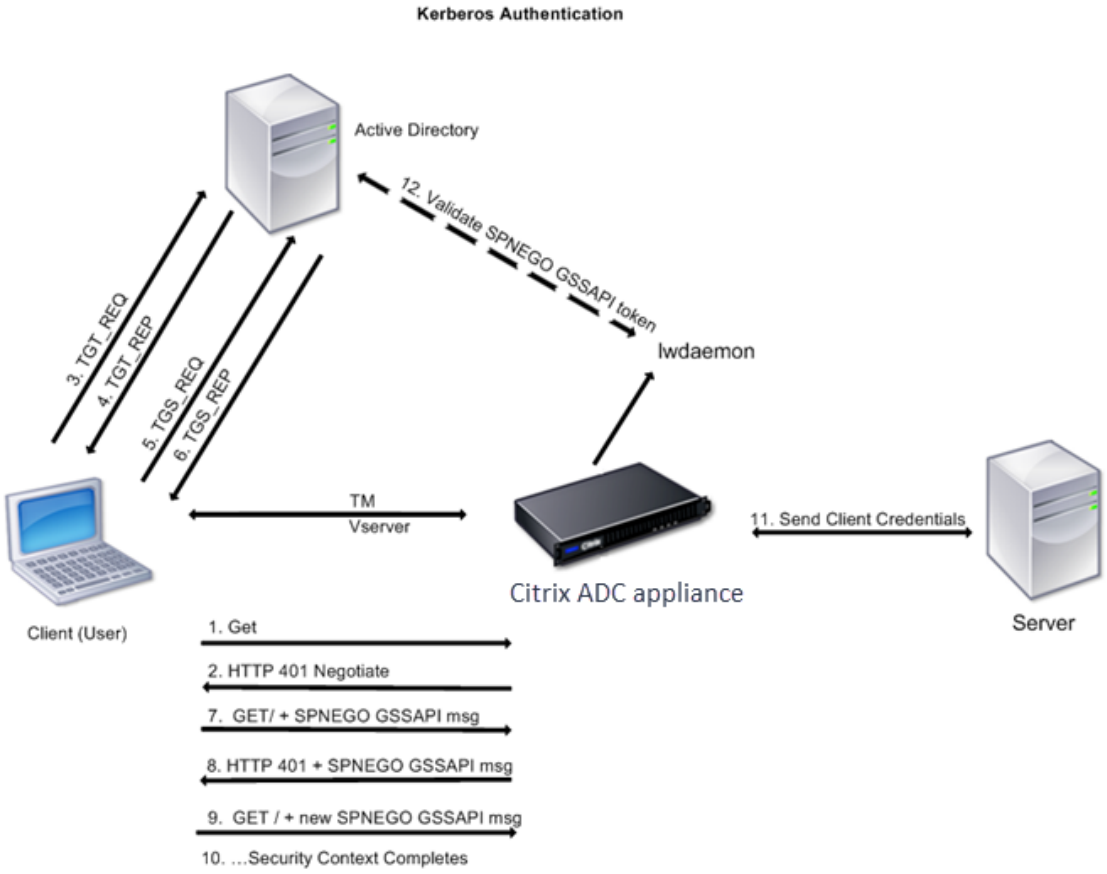
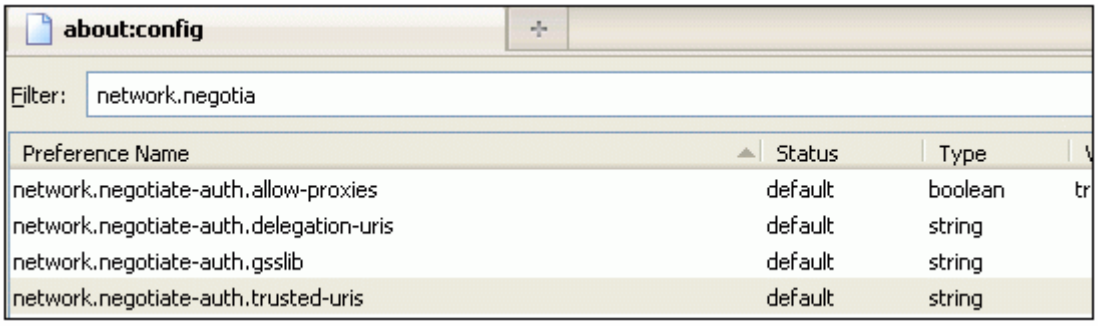
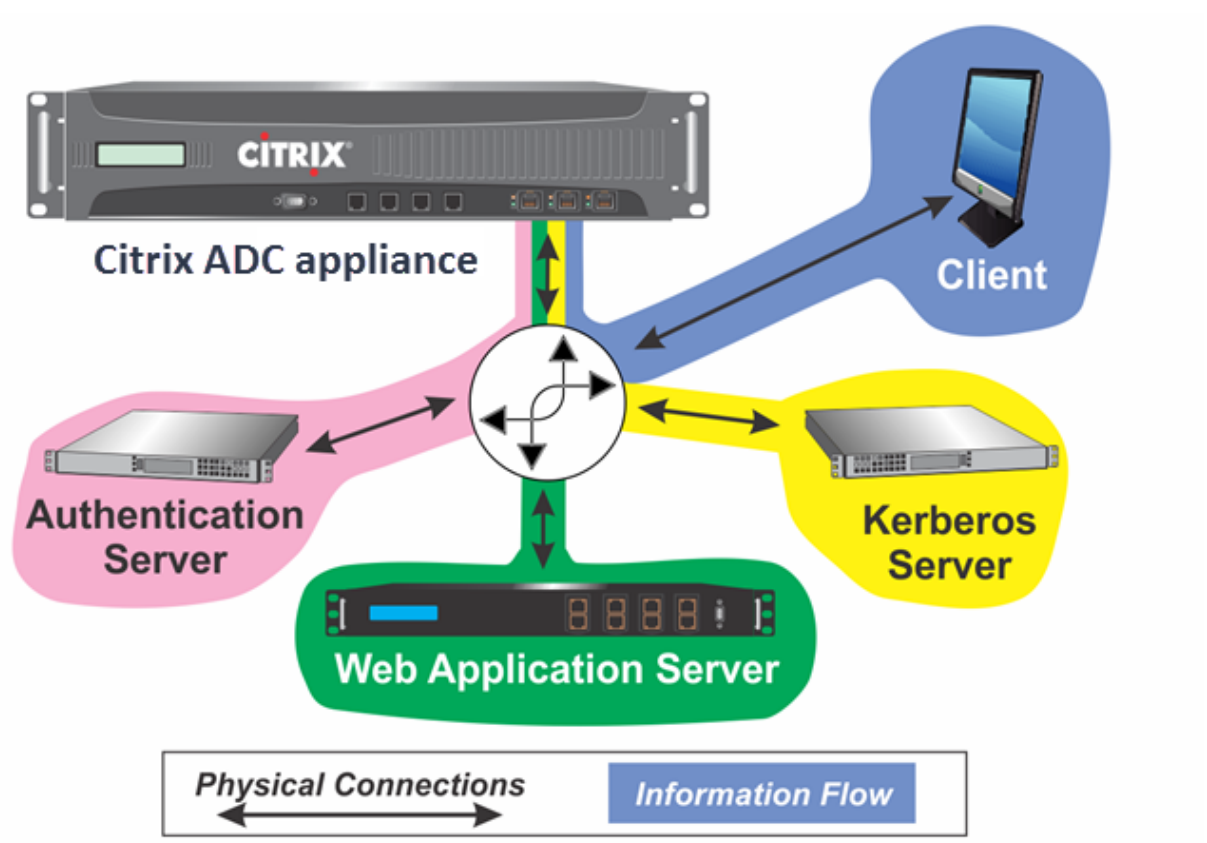
      "otpdata” : {

       “devices” : {

                        “device1” :  “value1” ,
                        “device2” :  “value2” , ⋯
                    }

          }

```



```
13       }
14
15 <!--NeedCopy-->
```

Where, value1 is base64 encoded value of kid + IV +cipher data

Cipher data is structured as following:

```
1     {
2
3       secret:<16-byte secret>,
4       tag : <64-byte tag value>
5       alg: <algorithm used> (not mandatory, default is sha1, specify
            the algorithm only if it is not default)
6       }
7
8  <!--NeedCopy-->
```

- In “devices”, you have value against each name. The value is base64encode(kid).base64encode(IV).base64en
- In standard AES algorithms, IV is always sent as first 16 or 32 bytes of cipher data. You can follow the same model.
- IV will differ for each device though key remains the same.

OTP encryption tool setup

The OTP encryption tool is located in the directory `\var\netscaler\otptool`. You must download the code from the Citrix ADC source and run the tool with the required AD credentials.

- Prerequisites for using the OTP encryption tool:
  - Install python 3.5 or higher version in the environment where this tool is run.
  - Install pip3 or later versions.
- Execute the following commands:
  - **pip install requirements.txt**. Automatically installs the requirements
  - **python main.py**. Invokes the OTP encryption tool. You must provide the required arguments as per your need for the migration of OTP secret data.
- The tool can be located at `\var\netscaler\otptool` from shell prompt.
- Run the tool with the required AD credentials.

OTP encryption tool interface

The following figure displays a sample OTP encryption tool interface. The interface contains all the arguments that must be defined for encryption/decryption/certificate upgrade. Also, a brief description of each argument is captured.

OPERATION argument

You must define the OPERATION argument to use the OTP encryption tool for encryption, decryption, or certificate upgrade.

The following table summarizes some of the scenarios in which you can use the OTP encryption tool and the corresponding OPERATION argument values.

Scenario	Operation argument value and other arguments
Convert plaintext OTP secret to encrypted format in the same attribute	Enter the OPERATION argument value as 0 and provide the same value for source and target attribute. **Example:** `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute unixhomedirectory -operation 0 -cert_path aaatm_wild_all.cert`
Convert plaintext OTP secret to encrypted format in a different attribute	Enter the OPERATION argument value as 0 and provide the corresponding values for source and target attribute. **Example:** `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 0 -cert_path aaatm_wild_all.cert`
Convert the encrypted entries back to plaintext	Enter the OPERATION argument value as 1 and provide the corresponding values for source and target attribute. **Example:** `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 1 -cert_path aaatm_wild_all.cert`

Scenario	Operation argument value and other arguments
Update the certificate to a new certificate	Enter the OPERATION argument value as 2 and provide all the previous certificate and the new certificate details in the corresponding arguments. **Example:** `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 2 -cert_path aaatm_wild_all.cert -new_cert_path aaatm_wild_all_new.cert`

CERT_PATH argument

The CERT_PATH argument specifies the certificate which is used in the Citrix ADC for encrypting the data. The user must provide this argument for all the three operations namely **Encryption**, **Decryption**, and **Update certificates**.

> **Points to note about the certificate**
>
> - The user must provide the same certificate which is bound globally in Citrix ADC for user data encryption.
> - The certificate must contain the Base64 encoded public certificate and its corresponding RSA private key in the same file.
> - The format of the certificate has to be either PEM or CERT. The certificate must adhere to X509 format.
> - Password protected certificate format and *.pfx* file are not accepted by this tool. The user must convert the PFX certificates to *.cert* before providing the certificates to the tool.

SEARCH_FILTER argument

The SEARCH_FILTER argument is used to filter the AD domains or users. The format of this search filter is same as the LDAP search filter format used in the LDAP action command in Citrix ADC.

Enabling encryption option in the Citrix ADC appliance

To encrypt the plain text format, you must enable the encryption option in the Citrix ADC appliance.

To enable OTP encryption data by using the CLI, at the command prompt, type:

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

Example:

```
set aaa otpparameter -encryption ON
```

OTP encryption tool use cases

The OTP encryption tool can be used for the following use cases.

Register new devices with Citrix ADC appliance version 13.0 build 41.20

When you register your new device with Citrix ADC appliance version 13.0 build 41.x, and if the encryption option is enabled, then the OTP data is saved in an encrypted format. You can avoid manual intervention.

If the encryption option is not enabled, the OTP data is stored in the plain text format.

Migrate OTP data for the devices registered previous to 13.0 build 41.20

You must perform the following to encrypt the OTP secret data for the devices that are registered with Citrix ADC appliance prior to 13.0 build 41.20.

- Use the conversion tool to migrate OTP data from plain text format to encrypted format.
- Enable the “Encryption” parameter on Citrix ADC appliance.
  - To enable encryption option by using the CLI:
    - `set aaa otpparameter -encryption ON`
  - To enable encryption options by using the GUI:
    - Navigate to **Security > AAA – Application Traffic** and click **Change authentication AAA OTP Parameter** under **Authentication Settings** section.
    - On the **Configure AAA OTP Parameter** page, select **OTP Secret encryption**, and click **OK**.
  - Log in with the valid AD credentials.
  - If it is required, register additional devices (optional).

Migrate encrypted data from old certificate to new certificate

If admins want to update the certificate to a new certificate, the tool provides an option to update the new certificate data entries.

To update the certificate to a new certificate by using the CLI

At the command prompt, type:

Example:

```
python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local
-search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -
target_attribute userparameters -operation 2 -cert_path aaatm_wild_all.cert
 -new_cert_path aaatm_wild_all_new.cert
```

> **Note**
>
> - The certificates must contain both private and public keys.
> - Currently, the functionality is provided only for OTP.

Re-encrypt or migrate to new certificate for devices registered after the appliance is upgraded to 13.0 build 41.20 with encryption

Admin can use the tool on the devices that are already encrypted with a certificate, and can update that certificate with a new certificate.

Convert encrypted data back to plain text format

Admin can decrypt the OTP secret and revert them to the original plain text format. The OTP encryption tool scans through all the users for OTP secret in encrypted format and converts them to decrypted format.

To update the certificate to a new certificate by using the CLI

At the command prompt, type:

Example:

```
1 python3 main.py -Host 192.0.2.1 - Port 636 -username  ldapbind_user@aaa
     .local -search_base cn=users,dc=aaa,dc=local -source_attribute
     unixhomedirectory -target_attribute userparameters -operation 1
2 <!--NeedCopy-->
```

Troubleshooting

The tool generates the following log files.

- **app.log**. Logs all the major steps of execution and information about errors, warnings, and failures.

- **unmodified_users.txt**. Contains a list of User DNs that was not upgraded from plain text to encrypted format. These logs are generated to an error in format or might be due to some other reason.

Push notification for OTP

September 14, 2021

Citrix Gateway supports push notifications for OTP. Users do not have to manually enter the OTP received on their registered devices to log in to Citrix Gateway. Admins can configure Citrix Gateway such that login notifications are sent to users' registered devices using push notification services. When users receive the notification, they have to simply tap Allow on the notification to log in to Citrix Gateway. When gateway receives acknowledgment from the user, it identifies source of the request, and sends response to that browser connection.

If the notification response is not received within the timeout period (30 seconds), users are redirected to the Citrix Gateway login page. The users can then enter the OTP manually or click **Resend Notification** to receive the notification again on the registered device.

Admins can make push notification authentication as the default authentication by using the login schemas created for push notification.

> **Important:**
>
> Push notification feature is available with a Citrix ADC Premium edition license.

Advantages of push notifications

- Push notifications provide a more secure multifactor authentication mechanism. Authentication to Citrix Gateway is not successful until the user approves the login attempt.
- Push notification is easy to administer and use. Users have to download and install the Citrix SSO mobile app that does not require any administrator assistance.
- Users do not have to copy or remember the code. They have to simply tap on the device to get authenticated.
- Users can register multiple devices.

How push notifications work

The push notification workflow can be classified into two categories:

- Device registration
- End user login



Prerequisites for using push notification

- Complete the Citrix Cloud onboarding process.
  1. Create a Citrix Cloud company account or join an existing one. For detailed processes and instructions on how to proceed, see Signing Up for Citrix Cloud.
  2. Log in to https://citrix.cloud.com, and select the customer.
  3. From Menu, select **Identity and Access Management** and then navigate to **API Access** tab to create a client for the account.
  4. Copy the ID, secret, and customer ID. The ID and secret are required to configure push service in Citrix ADC as “ClientID” and “ClientSecret” respectively.

> **Important:**
>
> - Same API credentials can be used on multiple data centers.
> - On premises Citrix ADC appliances must be able to resolve server addresses mfa.cloud.com and trust.citrixworkspacesapi.net and are accessible from the appliance. This is to ensure that there are no firewalls or IP address blocks for these servers over port 443.

- Download the Citrix SSO mobile app from App Store and Play Store for iOS devices and Android devices respectively. Push notification is supported on iOS from build 1.1.13 on Android from 2.3.5.
- Ensure the following for the Active Directory.
  - Minimum attribute length must be at least 256 characters.

- Attribute type must be 'DirectoryString' such as UserParameters. These attributes can hold string values.
- Attribute string type must be Unicode, if device name is in non-English characters.
- Citrix ADC LDAP administrator must have write access to the selected AD attribute.
- Citrix ADC and the client machine must be synchronized to a common Network Time Server.

Push notification configuration

The following are the high-level steps that must be completed to use the push notification functionality.

- The Citrix Gateway administrator must configure the interface to manage and validate users.
  1. Configure a push service.
  2. Configure Citrix Gateway for OTP management and end user login.

     Users must register their devices with gateway for logging in to Citrix Gateway.
  3. Register your device with Citrix Gateway.
  4. Log in to Citrix Gateway.

Create a push service

1. Navigate to **Security** > **AAA-Application Traffic** > **Policies** > **Authentication** > **Advanced Policies** > **Actions** > **Push Service** and click **Add**.
2. In **Name**, enter the name of the push service.
3. In **Client ID**, enter the unique identity of the relying party for communicating with Citrix Push server in cloud.
4. In **Client Secret**, enter the unique secret of the relying party for communicating with Citrix Push server in cloud.
5. In **Customer ID**, enter the customer ID or name of the account in cloud that is used to create Client ID and Client Secret pair.

> **Important**
>
> The TLS 1.2 version is required for push service. For more information, see TLS 1.2 configuration details.

Configure Citrix Gateway for OTP management and end user login

Complete the following steps for OTP management and end user login.

- Create login schema for OTP management
- Configure authentication, authorization, and auditing virtual server
- Configure VPN or load balancing virtual servers
- Configure policy label
- Create login schema for end user login

For details on configuration, see Native OTP support.

Important: For push notification, admins must explicitly configure the following:

- Create a push service.
- While creating login schema for OTP management, select the SingleAuthManageOTP.xml login schema or equivalent as per the need.
- While creating login schema for end user login, select the DualAuthOrPush.xml login schema or equivalent as per the need.

Register your device with Citrix Gateway

Users must register their devices with Citrix Gateway to use the push notification functionality.

1. In your web browser, browse to your Citrix Gateway FQDN, and suffix **/manageotp** to the FQDN.

   This loads the authentication page.
   Example: https://gateway.company.com/manageotp

2. Log in using your LDAP credentials or appropriate two-factor authentication mechanisms, as required.



3. Click **Add Device**.

4. Enter a name for your device, then click **Go**.

   A QR code is displayed on the Citrix Gateway browser page.



5. Scan this QR code using the Citrix SSO app from the device to be registered.

   Citrix SSO validates the QR code and then registers with gateway for push notifications. If there are no errors in the registration process, the token is successfully added to the password tokens page.







6. If there are no additional devices to add/manage log out using the list at the top right corner of the page.

Test one-time password authentication

1. To test the OTP, click your device from the list and then click **Test**.
2. Enter the OTP that you have received on your device and click **Go**.

   The OTP verification successful message appears.

3. Log out using the list at the top right corner of the page.

Note: You can use the OTP management portal anytime to test authentication, remove registered devices, or register more devices.

Log in to Citrix Gateway

After registering their devices with Citrix Gateway, users can use the push notification functionality for authentication.

1. Navigate to your Citrix Gateway authentication page (for example: https://gateway.company.com)

   You are prompted to enter only your LDAP credentials depending on the login schema configuration.



2. Enter your LDAP user name and password, then select **Submit**.

   A notification is sent to your registered device.

   Note: If you want to enter the OTP manually, you must select **Click** to input OTP manually and enter the OTP in the **TOTP** field.

3. Open the Citrix SSO app on your registered device and tap **Allow**.



Note:

- In case of an iOS device, you are prompted for Touch-ID/Face-ID/Passcode as an extra factor of authentication.
- The authentication server waits for push server notification response until the configured timeout period expires. After the timeout, Citrix Gateway displays the login page. The users can then enter the OTP manually or click **Resend Notification** to receive the notification again on the registered device. Based on your selected option, gateway validates the OTP that you have entered or resends the notification on your registered device.



- No notification is sent to your registered device regarding login failure.

Failure conditions

- The device registration might fail in the following cases.
  - Server certificate might not be trusted by end-user device.
  - Citrix Gateway used to register for OTP is not reachable by client.
- The notifications might fail in the following cases.
  - User device is not connected to the internet
  - Notifications on the user device are blocked
  - User does not approve the notification on the device

In these cases, the authentication server waits until the configured timeout period expires. After the timeout, the Citrix Gateway displays a login page with the options to manually enter the OTP or to resend the notification again on your registered device. Based on the selected option, further validation occurs.

Failure logs

The following are the expected logs when the OTP push service is not reachable.

- Push notification failure when user device is not connected to the internet - Push: Failed to prepare Push Request to "`client name`" for Push service.
- Device registration failure log - Push: No devices are registered to send Push Request to cloud for "`client name`".
- In case user does not accept the push - Push: Response is not seen from client, for "`user name`", checking retry options.

Citrix SSO app behavior on iOS – points to note

Notification shortcuts

Citrix SSO iOS app includes support for actionable notifications to enhance user experience. Once a notification is received on an iOS device, and if the device is locked or the Citrix SSO app is not in foreground, users can use the shortcuts built into the notification to either approve or deny login request.

To access notification shortcuts, users need to either force touch (3D touch) or long press the notification depending on the device's hardware. Selecting the Allow shortcut action sends a login request to Citrix ADC. Depending on how the authentication policy is configured on the authentication, authorization, and auditing virtual server;

- The login request might be sent in the background without any need to launch the app into foreground or unlock the device.
- The app might prompt for Touch-ID/Face-ID/Passcode as an extra factor in which case the app is launched into foreground.





Deleting password tokens from Citrix SSO

1. To delete a password token registered for push in the Citrix SSO app, users must perform the following steps:
2. Unregister (remove) the iOS/Android device on gateway. QR code for removing registration from device appears.
3. Open the Citrix SSO app and tap the info button of the password token to be deleted.
4. Tap **Delete Token** and scan the QR code.

Note:

- If the QR code is valid, the token is successfully removed from the Citrix SSO app.
- Users can tap Force Delete to delete a password token without having to scan the QR code if the device is already removed from gateway. Force deleting might result in the device continuing to receive notifications if the device has not been removed from Citrix Gateway.





Email OTP authentication

September 14, 2021

Email OTP is introduced with Citrix ADC 12.1 build 51.x. The Email OTP method enables you to authenticate using the one-time password (OTP) that is sent to the registered email address. When you try to authenticate on any service, the server sends an OTP to the registered email address of the user.

To use the Email OTP feature, you must first register your alternate email ID. An alternative email ID registration is needed so that the OTP can be sent to that mail ID since you would not be able to access the primary email ID if there was an account lockout or in the event of you forgetting the AD password.

You can use Email OTP validation without email ID registration if you have provided the alternate email ID already as part of some AD attribute. You can refer to the same attribute in the email action instead of specifying the alternate email ID in the email address section.

Prerequisites

Before you configure the Email OTP feature, review the following prerequisites:

- Citrix ADC feature release 12.1 build 51.28 and above
- Email OTP feature is available in nFactor authentication flow only
  - For more details, refer to https://support.citrix.com/pages/citrix-adc-authentication-how#nfactor
  - Supported for AAA-TM, Citrix Gateway (Browser, Native plug-in, and Receiver).

Active directory setting

- Supported version is 2016/2012 and 2008 Active Directory domain function level
- Citrix ADC ldapBind user name must have write access to the user's AD path

Email Server

- For Email OTP solution to work, ensure that the login based authentication is enabled on the SMTP server. Citrix ADC supports only Auth login based authentication for Email OTP to work.
- To ensure that the Auth login based authentication is enabled, type the following command on the SMTP server. If the login based authentication is enabled, you notice that the text AUTH LOGIN appears in **bold** in the output.

```
root@ns# telnet <IP address of the SMTP server><Port number of the server>
ehlo
root@ns# telnet 10.106.3.
Trying 10.106.  ...
Connected to 10.106.  .
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22 Nov
2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.  ]
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
For information on how to enable login based authentication, see
https://support.microfocus.com/kb/doc.php?id=7020367
```

Limitations

- This feature is supported only if authentication back-end is LDAP.
- Already registered alternate email ID cannot be seen.
- Only the alternate email ID from the KBA Registration page cannot be updated.
- KBA and Email OTP Authentication and Registration cannot be the first factors in the authentication flow. This is by design to achieve a robust authentication.
- Same AD attribute must be configured for KBA and Alternate email ID if using the same authentication LDAP action.
- For native plug-in and Receiver, registration is supported only through a browser.

Active Directory Configuration

- Email OTP uses Active Directory attribute as user data storage.
- After you register the alternate email ID, they are sent to the Citrix ADC appliance and the appliance stores it in the configured KB attribute in the AD user object.
- The alternate email ID is encrypted and stored in the configured AD attribute.

When configuring an AD attribute, consider the following:

- Attribute name length supported must be at least 128 characters.
- Attribute type must be 'DirectoryString'.
- Same AD attribute can be used for Native OTP and KBA Registration data.
- LDAP administrator must have write access to the selected AD attribute.

Using existing attributes

The attribute used in this example is 'Userparameters'. As this is an existing attribute within the AD user, you do not need to make any changes to the AD itself. However, you have to make sure that the attribute is not being used.

To ensure that the attribute is not used, navigate to **ADSI** and select user, right-click on the user, and scroll down to the attribute list. You must see the attribute value for **UserParameters** as **not set**. This indicates that the attribute is not being used at the moment.



Configuring Email OTP

Email OTP solution consists of the following two parts:

- Email Registration
- Email Validation

Email Registration

There are two ways of registering a user's alternate email ID:

1. Along with KBA Registration
2. Only Email ID Registration - This method is supported from 13.0 build 61.x and above; and 12.1 build 58.x and above.

Along With KBA Registration

KBA Registration LoginSchema

1. Navigate to **Security > AAA – Application Traffic > Login Schema > Profiles** and click **Add KBA Registration LoginSchema**.



2. Configure KBA Registration Authentication Schema. This loginschema once generated shows all the Questions configured for the end user during the registration process. In **Email Registration** section, check the Register Alternate Email option to Register user's alternate email ID.



3. In the Email Registration section, check **Register Alternate Email** to register an alternate email ID.



Do the following configuration using the CLI command prompt after the aforementioned KBA Registration schema is created successfully.

1. Bind Portal Theme and Certificate to VPN global.

```
bind authentication vserver authvs -portaltheme RfWebUI
```



```
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> **Note:**
>
> Preceding certificate binding is required to encrypt the user data (KB Q&A and alternate email ID registered) stored in the AD attribute

2. Create an LDAP Authentication policy.

```
add authentication ldapAction ldap -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL
add authentication Policy ldap -rule true -action ldap
<!--NeedCopy-->
```

3. Create a KBA Registration Loginschema and PolicyLabel.

```
add authentication loginSchema Registrationschema -
    authenticationSchema /nsconfig/loginschema/LoginSchema/
    KBARegistrationSchema.xml [This is the authentication schema
    created in the previous section.]
add authentication policylabel Registrationfactor -loginSchema
    Registrationschema
add authentication ldapAction ldap_registration -serverIP
    10.102.2.2 -serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -
    ldapBindDn administrator@aaatm-test.com -ldapBindDnPassword
    freebsd -ldapLoginName samAccountName -secType SSL  -
    KBAttribute userParameters -alternateEmailAttr userParameters
add authentication Policy ldap_registration -rule true -action
    ldap_registration
bind authentication policylabel Registrationfactor -policyName
    ldap_registration -priority 1 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

4. Bind Authentication policy to authentication virtual server.

```
bind authentication vserver authvs - policy ldap -priority 1 -
    nextFactor Registrationfactor -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

5. Once you have configured all the steps mentioned in the previous sections, you must see the following GUI screen. Upon accessing via a URL for example, https://lb1.server.com/ you are presented with an initial login page that only requires the LDAP logon credential.



6. After login with a valid credential, you see the user registration page as follows.



7. Click **Submit** for user registration to be successful and session to be created.

Only Email ID Registration

Do the following configuration using the CLI command prompt after the aforemention KBA registration schema is created successfully:

1. Bind Portal Theme and Certificate to VPN global.

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> **Note:**
>
> Preceding Cert binding is required to encrypt the userdata (KB Q&A and alternate mail ID

> Registered) stored in AD attribute.

2. Create an LDAP authentication policy.

```
add authentication ldapAction ldap -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL
add authentication Policy ldap -rule true -action ldap
<!--NeedCopy-->
```

3. Create an LDAP authentication policy for Email Registration.

```
add authentication ldapAction ldap_email_registration -serverIP
    10.102.2.2 -serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -
    ldapBindDn administrator@aaatm-test.com -ldapBindDnPassword
    freebsd -ldapLoginName samAccountName -secType SSL -KBAttribute
     userParameters -alternateEmailAttr userParameters
add authentication Policy ldap_email_registration -rule true -
    action ldap_email_registration
<!--NeedCopy-->
```

4. Create an Email Registration Loginschema and PolicyLabel.

```
add authentication loginSchema onlyEmailRegistration -
    authenticationSchema /nsconfig/loginschema/LoginSchema/
    AltEmailRegister.xml
add authentication policylabel email_Registration_factor -
    loginSchema onlyEmailRegistration
bind authentication policylabel email_Registration_factor -
    policyName ldap_email_registration -priority 1 -
    gotoPriorityExpression NEXT
<!--NeedCopy-->
```

5. Bind Authentication policy to authentication virtual server.

```
bind authentication vserver authvs  - policy ldap -priority 1 -
    nextFactor email_Registration_factor -gotoPriorityExpression
    NEXT
<!--NeedCopy-->
```

6. Once you have configured all the steps mentioned in the previous sections, you must see the following GUI screen. Upon accessing via URL for example, https://lb1.server.com/ you are presented with an initial login page that only requires LDAP logon credential followed by an alternate email registration page.





Email Validation

Do the following steps for Email validaition.

1. Bind Portal Theme and Certificate to VPN global

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> **Note:**
>
> Preceding certificate binding is required to decrypt the userdata (KB Q&A and alternate email ID registered) stored in AD attribute.

2. Create an LDAP Authentication policy. LDAP must be a prior factor to email validation factor because you need the user's email ID or alternate email ID for Email OTP Validation

```
add authentication ldapAction ldap1 -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -   ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL -KBAttribute
    userParameters -alternateEmailAttr userParameters
```



```
2 add authentication Policy ldap1 -rule true -action ldap1
3 <!--NeedCopy-->
```

3. Create an Email authentication policy

```
1 add authentication emailAction email -userName sqladmin@aaa.com -
     password freebsd-encrypted -encryptmethod ENCMTHD_3 -serverURL
     "smtps://10.2.3.3:25" -content "OTP is $code" -
     defaultAuthenticationGroup emailgrp -emailAddress "aaa.user.
     attribute(\"alternate_mail\")"
2 add authentication Policy email -rule true  - action email
3 <!--NeedCopy-->
```

> In the previously mentioned command, **email address** is the alternate email ID user provided during KBA Registration.

4. Create an Email OTP validation policyLabel.

```
1 add authentication policylabel email_Validation_factor
2 bind authentication policylabel email_Validation_factor -
     policyName email -priority 1 -gotoPriorityExpression NEXT
3 <!--NeedCopy-->
```

5. Bind Authentication policy to authentication virtual server

```
1 bind authentication vserver authvs  -  policy ldap1 -priority 1 -
     nextFactor email_Validation_factor -gotoPriorityExpression NEXT
2 <!--NeedCopy-->
```

6. Once you have configured all the steps mentioned in the previous sections, you must see the following GUI screen for EMAIL OTP Validation. Upon accessing via URL for example, https://lb1.server.com/ you are presented with an initial login page that only requires the LDAP logon credential followed by the EMAIL OTP Validation page.

> **Note:**
>
> In the LDAP policy it is important to configure alternateEmailAttr to be able to query the user's email id from the AD attribute.



Troubleshooting

Before analyzing the log, it is better to set the log level to debug as follows.

```
set syslogparams -loglevel DEBUG
<!--NeedCopy-->
```

Registration – Successful Scenario

The following entries indicate a successful user registration.

```
"ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1"
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  "
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw=="

<!--NeedCopy-->
```


Registration – Failed Scenario

On the user login page, you see the following error message, “Cannot Complete your request”. This indicates that certkey to be bounded to VPN global for encrypting the user data is missing.

```
Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:4 6 GMT
    0-PPE-1 : default SSLVPN Message 696 0 :  "Encrypt UserData: No
    Encryption cert is bound to vpn global"
Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:46 GMT  0-
    PPE-1 : default SSLVPN Message 697 0 :  "KBA Register: Alternate
    email id Encrypted blob length is ZERO aaauser"
<!--NeedCopy-->
```

Email Validation – Successful Scenario

The following entries indicates a successful Email OTP Validation.

```
"NFactor: Successfully completed email auth, nextfactor is pwd_reset"
<!--NeedCopy-->
```

Email Validation – Failed Scenario

On the user login page, “Cannot Complete your request” error message is displayed. This indicates login based authentication is not enabled on the email server and the same needs to be enabled.

```
" /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp
    [100]: void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID:
    8]SMTP Configuration is Secure..
/home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[108]:
    void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID: 8]
    First login succeeded
Wed Mar  4 17:16:28 2020
/home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/naaad.c[697]: main
    0-0: timer 2 firing...
/home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[127]:
    void ThreadWorker_SendMailJob(SMTPJob*) 0-0: [POCO-ERROR][JobID: 8]
    Poco SMTP Mail Dispatch  Failed. SMTP TYPE:1, SMTPException:
    Exception occurs. SMTP Exception: The mail service does not support
    LOGIN authentication: 250-smtprelay.citrix.com Hello [10.9.154.239]
250-SIZE 62914560
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-8BITMIME
```



```
250-BINARYMIME
250 CHUNKING
<!--NeedCopy-->
```

reCaptcha configuration for nFactor authentication

September 14, 2021

Citrix Gateway supports a new first class action 'captchaAction' that simplifies reCaptcha configuration. As reCaptcha is a first class action, it can be a factor of its own. You can inject reCaptcha anywhere in the nFactor flow.

Previously, you had to write custom WebAuth policies with changes to RfWeb UI as well. With the introduction of captchaAction, you do not have to modify the JavaScript.

> **Important**
>
> If reCaptcha is used along with username or password fields in the schema, submit button is disabled until reCaptcha is met.

reCaptcha configuration

reCaptcha configuration involves two parts.

1. Configuration on Google for registering reCaptcha.
2. Configuration on Citrix ADC appliance to use reCaptcha as part of login flow.

reCaptcha configuration on Google

Register a domain for reCaptcha at https://www.google.com/recaptcha/admin.

1. When you navigate to this page, the following screen appears.



> **Note**
>
> Use reCAPTCHA v2 only. Invisible reCAPTCHA is still in Beta.

2. After a domain is registered, the “SiteKey” and “SecretKey” are displayed.



> **Note**
>
> The “SiteKey” and “SecretKey” are grayed out for security reasons. “SecretKey” must be

> kept safe.

reCaptcha configuration on Citrix ADC appliance

reCaptcha configuration on Citrix ADC appliance can be divided into three parts:

- Display reCaptcha screen
- Post the reCaptcha response to Google server
- LDAP configuration is second factor for user logon (optional)

Display reCaptcha screen

The login form customization is done through the SingleAuthCaptcha.xml loginschema. This customization is specified at authentication virtual server and is sent to UI for rendering the login form. The built-in loginschema, SingleAuthCaptcha.xml, is at /nsconfig/loginSchema/LoginSchema directory on the Citrix ADC appliance.

> **Important**
>
> - Based on your use case and different schemas, you can modify the existing schema. For instance if you need only reCaptcha factor (without username or password) or dual authentication with reCaptcha.
> - If any custom modifications are performed or the file is renamed, Citrix recommends copying all loginSchemas from /nsconfig/loginschema/LoginSchema directory to parent directory, /nsconfig/loginschema.

To configure display of reCaptcha using CLI

- `add authentication loginSchema singleauthcaptcha -authenticationSchema /nsconfig/loginschema/SingleAuthCaptcha.xml`
- `add authentication loginSchemaPolicy singleauthcaptcha -rule true -action singleauthcaptcha`
- `add authentication vserver auth SSL <IP> <Port>`
- `add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-key-file>`
- `bind ssl vserver auth -certkey vserver-cert`
- `bind authentication vserver auth -policy singleauthcaptcha -priority 5 -gotoPriorityExpression END`

Post the reCaptcha response to Google server

After you have configured the reCaptcha that must be displayed to the users, admins post add the configuration to the Google server to verify the reCaptcha response from browser.

To verify reCaptcha response from the browser

- `add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>`
- `add authentication policy myrecaptcha -rule true -action myrecaptcha`
- `bind authentication vserver auth -policy myrecaptcha -priority 1`

The following commands are required to configure if AD authentication is desired. Else, you can ignore this step.

- `add authentication ldapAction ldap-new -serverIP x.x.x.x -serverPort 636 -ldapBase "cn=users,dc=aaatm,dc=com"-ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup`
- `add authenticationpolicy ldap-new -rule true -action ldap-new`

LDAP configuration is second factor for user logon (optional)

The LDAP authentication happens after reCaptcha, you add it to the second factor.

- `add authentication policylabel second-factor`
- `bind authentication policylabel second-factor -policy ldap-new -priority 10`
- `bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor`

Administrator needs to add appropriate virtual servers depending on whether load balancing virtual server or Citrix Gateway appliance is used for access. Administrator must configure the following command if load balancing virtual server is required:

- `add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com`

  nssp.aaatm.com – Resolves to authentication virtual server.

User validation of reCaptcha

Once you have configured all the steps mentioned in the previous sections, you must see the UI screenshots shown below.

1. Once the authentication virtual server loads the login page, the logon screen is displayed. **Log On** is disabled until reCaptcha is complete.



2. Select I’m not a robot option. The reCaptcha widget is displayed.



3. You are navigated through series of reCaptcha images, before the completion page is displayed.
4. Enter the AD credentials, select the **I'm not a robot** check box and click **Log On**. If authentication succeeds, you are redirected to the desired resource.



Notes

- If reCaptcha is used with AD authentication, Submit button for credentials is disabled until reCaptcha is complete.
- The reCaptcha happens in a factor of its own. Therefore, any subsequent validations like AD must happen in the 'nextfactor' of reCaptcha.

Authentication, authorization, and auditing configuration for commonly used protocols

September 14, 2021

Configuring the Citrix ADC appliance for authentication, authorization, and auditing needs a specific setup on the Citrix ADC appliance and clients' browsers. The configuration varies with the protocol used for authentication, authorization, and auditing.

For more information about configuring the Citrix ADC appliance for Kerberos authentication, see Handling Authentication, Authorization and Auditing with Kerberos/NTLM.

Handling authentication, authorization and auditing with Kerberos/NTLM

September 14, 2021

Kerberos, a computer network authentication protocol, provides secure communication over the Internet. Designed primarily for client-server applications, it provides for mutual authentication by which the client and server can each ensure the other's authenticity. Kerberos uses a trusted third party, referred to as Key Distribution Center (KDC). A KDC consists of an Authentication Server (AS), which authenticates a user, and a Ticket Granting Server (TGS).

Each entity on the network (client or server) has a secret key that is known only to itself and the KDC. The knowledge of this key implies authenticity of the entity. For communication between two entities on the network, the KDC generates a session key, referred to as the Kerberos ticket or service ticket. The client makes a request to the AS for credentials for a specific server. The client then receives a ticket, referred to as Ticket Granting Ticket (TGT). The client then contacts the TGS, using the TGT it received from the AS to prove its identity, and asks for a service. If the client is eligible for the service, the TGS issues a Kerberos ticket to the client. The client then contacts the server hosting the service (referred to as the service server), using the Kerberos ticket to prove that it is authorized to receive the service. The Kerberos ticket has a configurable lifetime. The client authenticates itself with the AS only once. If it contacts the physical server multiple times, it reuses the AS ticket.

The following figure shows the basic functioning of the Kerberos protocol.

Figure 1. **Functioning of Kerberos**



Kerberos authentication has the following advantages:

- Faster authentication. When a physical server gets a Kerberos ticket from a client, the server has enough information to authenticate the client directly. It does not have to contact a domain controller for client authentication, and therefore the authentication process is faster.
- Mutual authentication. When the KDC issues a Kerberos ticket to a client and the client uses the ticket to access a service, only authenticated servers can decrypt the Kerberos ticket. If the virtual server on the Citrix ADC appliance is able to decrypt the Kerberos ticket, you can conclude that both the virtual server and client are authenticated. Thus, the authentication of the server happens along with the authentication of the client.
- Single sign-on between Windows and other operating systems that support Kerberos.

Kerberos authentication may have the following disadvantages:

- Kerberos has strict time requirements; the clocks of the involved hosts must be synchronized with the Kerberos server clock to ensure that the authentication does not fail. You can mitigate this disadvantage by using the Network Time Protocol daemons to keep the host clocks synchronized. Kerberos tickets have an availability period, which you can configure.
- Kerberos needs the central server to be available continuously. When the Kerberos server is down, no one can log on. You can mitigate this risk by using multiple Kerberos servers and fallback authentication mechanisms.
- Because all the authentication is controlled by a centralized KDC, any compromise in this infrastructure, such as the user's password for a local workstation being stolen, can allow an attacker to impersonate any user. You can mitigate this risk to some extent by using only a desktop machine or laptop that you trust, or by enforcing preauthentication by means of a hardware-token.

To use Kerberos authentication, you must configure it on the Citrix ADC appliance and on each client.

Optimizing Kerberos authentication on authentication, authorization, and auditing

The Citrix ADC appliance now optimizes and improves the system performance while Kerberos authentication. The authentication, authorization, and auditing daemon remembers the outstanding Kerberos request for the same user to avoid load on Key Distribution Center (KDC), which will avoid duplicate requests.

How Citrix ADC implements Kerberos for client authentication

September 14, 2021

> **Important**
>
> Kerberos/NTLM authentication is supported only in the NetScaler 9.3 nCore release or later, and it can be used only for authentication, authorization, and auditing traffic management virtual

> servers.

Citrix ADC handles the components involved in Kerberos authentication in the following way:

Key Distribution Center (KDC)

In the Windows 2000 Server or later versions, the Domain Controller and KDC are part of the Windows Server. If the Windows Server is UP and running, it indicates that the Domain Controller and KDC are configured. The KDC is also the Active Directory server.

> **Note**
>
> All Kerberos interactions are validated with the Windows Kerberos Domain Controller.

Authentication service and protocol negotiation

A Citrix ADC appliance supports Kerberos authentication on the authentication, authorization, and auditing traffic management authentication virtual servers. If the Kerberos authentication fails, the Citrix ADC uses the NTLM authentication.

By default, Windows 2000 Server and later Windows Server versions use Kerberos for authentication, authorization, and auditing. If you create an authentication policy with NEGOTIATE as the authentication type, the Citrix ADC attempts to use the Kerberos protocol for authentication, authorization, and auditing and if the client's browser fails to receive a Kerberos ticket, the Citrix ADC uses the NTLM authentication. This process is referred to as negotiation.

The client may fail to receive a Kerberos ticket in any of the following cases:

- Kerberos is not supported on the client.
- Kerberos is not enabled on the client.
- The client is in a domain other than that of the KDC.
- The Access Directory on the KDC is not accessible to the client.

For Kerberos/NTLM authentication, the Citrix ADC does not use the data that is present locally on the Citrix ADC appliance.

Authorization

The traffic management virtual server can be a load balancing virtual server or a content switching virtual server.

Auditing

The Citrix ADC appliance supports auditing of Kerberos authentication with the following audit logging:

- Complete audit trail of the traffic management end-user activity
- SYSLOG and high performance TCP logging
- Complete audit trail of system administrators
- All system events
- Scriptable log format

Supported Environment

Kerberos authentication does not need any specific environment on the Citrix ADC. The client (browser) must provide support for Kerberos authentication.

High Availability

In a high availability setup, only the active Citrix ADC joins the domain. In case of a failover, the Citrix ADC lwagent daemon joins the secondary Citrix ADC appliance to the domain. No specific configuration is required for this functionality.

Kerberos authentication process

The following figure shows a typical process for Kerberos authentication in the Citrix ADC environment.

Figure 1. Kerberos Authentication Process on Citrix ADC



The Kerberos authentication occurs in the following stages:

Client authenticates itself to the KDC

1. The Citrix ADC appliance receives a request from a client.
2. The traffic management (load balancing or content switching) virtual server on the Citrix ADC appliance sends a challenge to the client.
3. To respond to the challenge, the client gets a Kerberos ticket.
   - The client sends the Authentication Server of the KDC a request for a ticket-granting ticket (TGT) and receives the TGT. (See 3, 4 in the figure, Kerberos Authentication Process.)
   - The client sends the TGT to the Ticket Granting Server of the KDC and receives a Kerberos ticket. (See 5, 6 in the figure, Kerberos Authentication Process.)

> **Note**
>
> The above authentication process is not necessary if the client already has a Kerberos ticket whose lifetime has not expired. In addition, clients such as Web Services, .NET, or J2EE, which support SPNEGO, get a Kerberos ticket for the target server, create an SPNEGO token, and insert the token in the HTTP header when they send an HTTP request. They do not go through the client authentication process.

Client requests a service.

1. The client sends the Kerberos ticket containing the SPNEGO token and the HTTP request to the traffic management virtual server on the Citrix ADC. The SPNEGO token has the necessary GSSAPI data.
2. The Citrix ADC appliance establishes a security context between the client and the Citrix ADC. If the Citrix ADC cannot accept the data provided in the Kerberos ticket, the client is asked to get a different ticket. This cycle repeats till the GSSAPI data is acceptable and the security context is established. The traffic management virtual server on the Citrix ADC acts as an HTTP proxy between the client and the physical server.

Citrix ADC appliance completes the authentication.

1. After the security context is complete, the traffic management virtual server validates the SPNEGO token.
2. From the valid SPNEGO token, the virtual server extracts the user ID and GSS credentials, and passes them to the authentication daemon.
3. A successful authentication completes the Kerberos authentication.

Configuring kerberos authentication on the Citrix ADC appliance

September 14, 2021

This topic provides the detailed steps to configure Kerberos authentication on the Citrix ADC appliance by using the CLI and the GUI.

Configuring Kerberos authentication on the CLI

1. Enable the authentication, authorization, and auditing feature to ensure the authentication of traffic on the appliance.

   ns-cli-prompt> **enable ns feature** AAA

2. Add the keytab file to the Citrix ADC appliance. A keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. A single keytab file contains authentication details for all the services that are bound to the traffic management virtual server on the Citrix ADC appliance.

   First generate the keytab file on the Active Directory server and then transfer it to the Citrix ADC appliance.

   - Log on to the Active Directory server and add a user for Kerberos authentication. For example, to add a user named “Kerb-SVC-Account”:

net user Kerb-SVC-Account freebsd!@#456 /add

> **Note**
>
> In the **User Properties** section, ensure that the “Change password at next logon option” is not selected and the “Password does not expire” option is selected.

- Map the HTTP service to the above user and export the keytab file. For example, run the following command on the Active Directory server:

  ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456 /mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL

  > **Note**
  >
  > You can map more than one service if authentication is required for more than one service. If you want to map more services, repeat the above command for every service. You can give the same name or different names for the output file.

- Transfer the keytab file to the Citrix ADC appliance by using the unix **ftp** command or any other file transfer utility of your choice.

3. The Citrix ADC appliance must obtain the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, Citrix recommends configuring the Citrix ADC with a DNS server.

   ns-cli-prompt> **add dns nameserver** <ip-address>

   > **Note**
   >
   > Alternatively, you can add static host entries or use any other means so that the Citrix ADC appliance can resolve the FQDN name of the domain controller to an IP address.

4. Configure the authentication action and then associate it to an authentication policy.

   - Configure the negotiate action.

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

     Note: For domain user and domain name configuration, go to client and use the klist command as shown in the following example:

     Client: username @ AAA.LOCAL

     Server: HTTP/onprem_idp.aaa.local @ AAA.LOCAL

     add authentication negotiateAction <name> -domain -domainUser <HTTP/onprem_idp.aaa.local>

- Configure the negotiate policy and associate the negotiate action to this policy.

  ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. Create an authentication virtual server and associate the negotiate policy with it.
   - Create an authentication virtual server.

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - Bind the negotiate policy to the authentication virtual server.

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   > **Note**
   >
   > Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations by doing the following:
   - Access the traffic management virtual server, using the FQDN. For example, Sample
   - View the details of the session on the CLI.

     ns-cli-prompt> **show aaa session**

Configuring Kerberos authentication on the GUI

1. Enable the authentication, authorization, and auditing feature.

   Navigate to **System** > **Settings**, click **Configure Basic Features** and enable the authentication, authorization, and auditing feature.

2. Add the keytab file as detailed in step 2 of the CLI procedure mentioned above.
3. Add a DNS server.

   Navigate to **Traffic Management > DNS > Name Servers**, and specify the IP address for the DNS server.

4. Configure the **Negotiate** action and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, and create a policy with **Negotiate** as the action type. Click **ADD** to create a new authentication negotiate server or click **Edit** to configure the existing details.

5. Bind the negotiate policy to the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the **Negotiate** policy with the authentication virtual server.

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and specify the relevant authentication settings.

   > **Note**
   >
   > Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations as detailed in step 7 of the CLI procedure mentioned above.

Configure kerberos authentication on a client

September 14, 2021

Kerberos support must be configured on the browser to use Kerberos for authentication. You can use any Kerberos-compliant browser. Instructions for configuring Kerberos support on Internet Explorer and Mozilla Firefox follow. For other browsers, see the documentation of the browser.

To configure Internet Explorer for Kerberos authentication

1. In the **Tools** menu select **Internet Options**.
2. On the **Security** tab, click **Local Intranet**, and then click **Sites**.
3. In the **Local Intranet** dialog box, make sure that the Automatically detect intranet network option is selected, and then click **Advanced**.
4. In the **Local Intranet** dialog box, add the web sites of the domains of the traffic management virtual server on the Citrix ADC appliance. The specified sites become local intranet sites.
5. Click **Close** or **OK** to close the dialog boxes.

To configure Mozilla Firefox for Kerberos authentication

1. Make sure that you have Kerberos properly configured on your computer.
2. Type about:config in the URL bar.
3. In the filter text box, type network.negotiate.
4. Change network.negotiate-auth.delegation-uris to the domain that you want to add.
5. Change network.negotiate-auth.trusted-uris to the domain that you want to add.

> **Note:** If you are running Windows, you also need to enter sspi in the filter text box and change the network.auth.use-sspi option to False.

Offload Kerberos authentication from physical servers

September 14, 2021

The Citrix ADC appliance can offload authentication tasks from servers. Instead of the physical servers authenticating the requests from clients, the Citrix ADC authenticates all the client requests before it forwards them to any of the physical servers bound to it. The user authentication is based on Active Directory tokens.

There is no authentication between the Citrix ADC and the physical server, and the authentication offload is transparent to the end users. After the initial logon to a Windows computer, the end user does not have to enter any additional authentication information in a pop-up or on a logon page.

In the current Citrix ADC appliance release, Kerberos authentication is available only for authentication, authorization, and auditing traffic management virtual servers. Kerberos authentication is not supported for SSL VPN in the Citrix Gateway Advanced Edition appliance or for Citrix ADC appliance management.

Kerberos authentication requires configuration on the Citrix ADC appliance and on client browsers.

To configure Kerberos authentication on the Citrix ADC appliance

1. Create a user account on Active Directory. When creating a user account, verify the following options in the User Properties section:
    - Make sure that you do not select the Change password at next logon option.
    - Be sure to select the Password does not expire option.
2. On the AD server, at the CLI command prompt, type:
    - ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

    > **Note**
    >
    > Be sure to type the above command on a single line. The output of the above command is written into the C:\kerbtabfile.txt file.
3. Upload the kerbtabfile.txt file to the /etc directory of the Citrix ADC appliance by using a Secure Copy (SCP) client.
4. Run the following command to add a DNS server to the Citrix ADC appliance.

- add dns nameserver 1.2.3.4

The Citrix ADC appliance cannot process Kerberos requests without the DNS server. Be sure to use the same DNS server that is used in the Microsoft Windows domain.

5. Switch to the command line interface of Citrix ADC.
6. Run the following command to create a Kerberos authentication server:
   - add authentication negotiateAction KerberosServer -domain “crete.lab.net” -domainUser kerbuser -domainUserPasswd Citrix1 -keytab /var/mykcd.keytab

   > **Note**
   >
   > If keytab is not available, you can specify the parameters: domain, domainUser, and -domainUserPasswd.

7. Run the following command to create a negotiation policy:
   - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`
8. Run the following command to create an authentication virtual server.
   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`
9. Run the following command to bind the Kerberos policy to the authentication virtual server:
   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`
10. Run the following command to bind an SSL certificate to the authentication virtual server. You can use one of the test certificates, which you can install from the GUI Citrix ADC appliance. Run the following command to use the ServerTestCert sample certificate.
    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`
11. Create an HTTP load balancing virtual server with the IP address, 192.168.17.200.

    Ensure that you create a virtual server from the command line interface for NetScaler 9.3 releases if they are older than 9.3.47.8.
12. Run the following command to configure an authentication virtual server:
    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`
13. Enter the host name Example in the address bar of the Web browser.

    The Web browser displays an authentication dialog box because the Kerberos authentication is not set up in the browser.

> **Note**
>
> Kerberos authentication requires a specific configuration on the client. Ensure that the client can resolve the hostname, which results in the Web browser connecting to an HTTP virtual server.

14. Configure Kerberos on the Web browser of the client computer.
    - For configuring on Internet Explorer, see Configuring Internet Explorer for Kerberos authentication.
    - For configuring on Mozilla Firefox, see Configuring Internet Explorer for Kerberos authentication.
15. Verify whether you can access the backend physical server without authentication.

To configure Internet Explorer for Kerberos authentication

1. Select **Internet Options** from the **Tools** menu.
2. Activate the **Security** tab.
3. Select **Local Intranet** from the Select a zone to view change security settings section.
4. Click **Sites**.
5. Click **Advanced**.
6. Specify the URL, Example and click **Add**.
7. Restart **Internet Explorer**.

To configure Mozilla Firefox for Kerberos authentication

1. Enter about:config in the address bar of the browser.
2. Click the warning disclaimer.
3. Type **Network.Negotiate-auth.trusted-uris** in the **Filter** box.
4. Double click **Network.Negotiate-auth.trusted-uris.** A sample screen is shown below.



5. In the Enter String Value dialog box, specify www.crete.lab.net.
6. Restart Firefox.

Single sign-on types

September 14, 2021

Citrix ADC Authentication, authorization, and auditing features supports the following single sign-on types.

- **Citrix ADC kerberos single sign-on:** Citrix ADC appliances now support single sign-on (SSO) using the Kerberos 5 protocol. Users log on to a proxy, the Application Delivery Controller (ADC), which then provides access to protected resources. For details, see Citrix ADC kerberos single sign-on.
- **SSO for Basic, Digest, and NTLM authentication:** Single Sign-On (SSO) configuration in Citrix ADC and Citrix Gateway can be enabled at global level and also per traffic level. By default the SSO configuration is OFF and an administrator can enable the SSO per traffic or globally. From a security point of view, Citrix recommends administrators to turn SSO globally OFF and enable per traffic basis. This enhancement is to make SSO configuration more secure by disabling certain type of SSO methods globally. For details, see SSO for Basic, Digest, and NTLM authentication.

Citrix ADC kerberos single sign-on

September 14, 2021

Citrix ADC appliances now support single sign-on (SSO) using the Kerberos 5 protocol. Users log on to a proxy, the Application Delivery Controller (ADC), which then provides access to protected resources.

The Citrix ADC Kerberos SSO implementation requires the user's password for SSO methods that rely on basic, NTLM, or forms-based authentication. The user's password is not required for Kerberos SSO, although if Kerberos SSO fails and the Citrix ADC appliance has the user's password, it uses the password to attempt NTLM SSO.

If the user's password is available, the KCD account is configured with a realm, and no delegated user information is present, the Citrix AD Kerberos SSO engine impersonates the user to obtain access to authorized resources. Impersonation is also called unconstrained delegation.

The Citrix ADC Kerberos SSO engine can also be configured to use a delegated account to obtain access to protected resources on the user's behalf. This configuration requires delegated user credentials, a keytab, or a delegated user certificate and matching CA certificate. Configuration that uses a delegated account is called constrained delegation.

An overview of Citrix ADC kerberos SSO

September 14, 2021

To use the Citrix ADC Kerberos SSO feature, users first authenticate with Kerberos or a supported third-party authentication server. Once authenticated, the user requests access to a protected web application. The web server responds with a request for proof that the user is authorized to access that web application. The user's browser contacts the Kerberos server, which verifies that the user is authorized to access that resource, and then provides the user's browser with a service ticket that provides proof. The browser resends the user's request to the web application server with the service ticket attached. The web application server verifies the service ticket, and then allows the user to access the application.

Authentication, authorization, and auditing traffic management implements this process as shown in the following diagram. The diagram illustrates the flow of information through the Citrix ADC appliance and authentication, authorization, and auditing traffic management, on a secure network with LDAP authentication and Kerberos authorization. Authentication, authorization, and auditing traffic management environments that use other types of authentication have essentially the same information flow, although they might differ in some details.

Figure 1. A Secure Network with LDAP and Kerberos



The authentication, authorization, and auditing traffic management with authentication and authorization in a Kerberos environment requires that the following actions take place.

1. The client sends a request for a resource to the traffic management virtual server on the Citrix ADC appliance.
2. The traffic management virtual server passes the request to the authentication virtual server, which authenticates the client and then passes the request back to the traffic management virtual server.
3. The traffic management virtual server sends the client's request to the web application server.
4. The web application server responds to the traffic management virtual server with a 401 Unauthorized message that requests Kerberos authentication, with fallback to NTLM authentication if the client does not support Kerberos.
5. The traffic management virtual server contacts the Kerberos SSO daemon.
6. The Kerberos SSO daemon contacts the Kerberos server and obtains a ticket-granting ticket (TGT) allowing it to request service tickets authorizing access to protected applications.
7. The Kerberos SSO daemon obtains a service ticket for the user and sends that ticket to the traffic management virtual server.
8. The traffic management virtual server attaches the ticket to the user's initial request and sends the modified request back to the web application server.
9. The web application server responds with a 200 OK message.

These steps are transparent to the client, which just sends a request and receives the requested resource.

Integration of Citrix ADC Kerberos SSO with authentication methods

All authentication, authorization, and auditing traffic management authentication mechanisms support Citrix ADC Kerberos SSO. Authentication, authorization, and auditing traffic management supports the Kerberos SSO mechanism with the Kerberos, CAC (Smart Card) and SAML authentication mechanisms with any form of client authentication to the Citrix ADC appliance. It also supports the HTTP-Basic, HTTP-Digest, Forms-based, and NTLM (versions 1 and 2) SSO mechanisms if the client uses either HTTP-Basic or Forms-Based authentication to log on to the Citrix ADC appliance.

The following table shows each supported client-side authentication method, and the supported server-side authentication method for that client-side method.

Table 1. Supported Authentication Methods

	Basic/Digest/NTLM	Kerberos Constrained Delegation	User Impersonation
CAC (Smart Card): at SSL/T LS Layer		X	X
Forms-Based (LDAP/RADIUS/TA-CACS)	X	X	X
HTTP Basic (LDAP/RA-DIUS/TACACS)	X	X	X
Kerberos		X	
NT LM v1/v2		X	X
SAML		X	
SAML Two-Factor	X	X	X
Certificate Two-Factor	X	X	X

Set up Citrix ADC SSO

September 14, 2021

You can configure Citrix ADC SSO to work in one of two ways: by impersonation or by delegation. SSO by impersonation is a simpler configuration than SSO by delegation, and is therefore usually preferable when your configuration allows it. To configure Citrix ADC SSO by impersonation, you must have the user's user name and password.

To configure Citrix ADC SSO by delegation, you must have the delegated user's credentials in one of the following formats: the user's user name and password, the keytab configuration that includes the user name and an encrypted password, or the delegated user certificate and the matching CA certificate.

Prerequisites for configuring Citrix ADC SSO

Before you configure Citrix ADC SSO, you need to have your Citrix ADC appliance fully configured to manage traffic to and authentication for your web application servers. Therefore, you must configure either load balancing or content switching, and then authentication, authorization, and auditing, for these web application servers. You should also verify routing between the appliance, your LDAP server, and your Kerberos server.

If your network is not already configured in this manner, perform the following configuration tasks:

- Configure a server and service for each web application server.
- Configure a traffic management virtual server to handle traffic to and from your web application server.

Following are brief instructions and examples for performing each of these tasks from the Citrix ADC command line. For further assistance, see Setting up an Authentication Virtual Server.

To create a server and service by using the CLI

For Citrix ADC SSO to obtain a TGS (service ticket) for a service, either the FQDN assigned to the server entity on the Citrix ADC appliance must match the FQDN of the web application server, or the server entity name must match the NetBios name of the web application server. You can take either of the following approaches:

- Configure the Citrix ADC server entity by specifying the FQDN of the web application server.
- Configure the Citrix ADC server entity by specifying the IP address of the web application server, and assign the server entity the same name as the NetBios name of the web application server.

At the command prompt, type the following commands:

```
1 -  add server name <serverFQDN>
2
3 -  add service name serverName serviceType port
4 <!--NeedCopy-->
```

For the variables, substitute the following values:

- **serverName**. A name for the Citrix ADC appliance to use to refer to this server.
- **serverFQDN**. The FQDN of the server. If the server has no domain assigned to it, use the server's IP address and make sure that the server entity name matches the NetBios name of the web application server.
- **serviceName**. A name for the Citrix ADC appliance to use to refer to this service.
- **type**. The protocol used by the service, either HTTP or MSSQLSVC.
- **port**. The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example:

The following examples add server and service entries on the Citrix ADC appliance for the web application server was1.example.com. The first example uses the FQDN of the web application server; the second uses the IP address.

To add the server and service using the web application server FQDN, was1.example.com, you would type the following commands:


```
add server was1 was1.example.com
add service was1service was1 HTTP 80
<!--NeedCopy-->
```

To add the server and service using the web application server IP and NetBios name, where the web application server IP is 10.237.64.87 and its NetBios name is WAS1, you would type the following commands:

```
add server WAS1 10.237.64.87
add service was1service WAS1 HTTP 8
<!--NeedCopy-->
```

To create a traffic management virtual server by using the CLI

The traffic management virtual server manages traffic between the client and the web application server. You can use either a load balancing or a content switching virtual server as the traffic management server. The SSO configuration is the same for either type.

To create a load balancing virtual server, at the command prompt, type the following command:

```
add lb vserver <vserverName> <type> <IP> <port>
<!--NeedCopy-->
```

For the variables, substitute the following values:

- **vserverName**—A name for the Citrix ADC appliance to use to refer to this virtual server.
- **type**—The protocol used by the service, either HTTP or MSSQLSVC.
- **IP**—The IP address assigned to the virtual server. This would normally be an IANA-reserved, non-public IP address on your LAN.
- **port**—The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example:

To add a load balancing virtual server called tmvserver1 to a configuration that manages HTTP traffic on port 80, assigning it a LAN IP address of 10.217.28.20 and then binding the load balancing virtual server to the wasservice1 service, you would type the following commands:

```
add lb vserver tmvserver1 HTTP 10.217.28.20 80
bind lb vserver tmvserv1 wasservice1
<!--NeedCopy-->
```


To create an authentication virtual server by using the CLI

The authentication virtual server manages authentication traffic between the client and the authentication (LDAP) server. To create an authentication virtual server, at the command prompt type the following commands:

```
add authentication vserver <authvserverName> SSL <IP> 443
<!--NeedCopy-->
```

For the variables, substitute the following values:

- **authvserverName** —A name for the Citrix ADC appliance to use to refer to this authentication virtual server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the authentication virtual server is added by using the rename authentication vserver command.
- **IP**—The IP address assigned to the authentication virtual server. As with the traffic management virtual server, this address would normally be an IANA-reserved, non-public IP on your LAN.
- **domain**—The domain assigned to the virtual server. This would usually be the domain of your network. It is customary, though not required, to enter the domain in all capitals when configuring the authentication virtual server.

Example:

To add an authentication virtual server called authverver1 to your configuration and assign it the LAN IP 10.217.28.21 and the domain EXAMPLE.COM, you would type the following commands:

```
add authentication vserver authvserver1 SSL 10.217.28.21 443
<!--NeedCopy-->
```

To configure a traffic management virtual server to use an authentication profile

The authentication virtual server can be configured to handle authentication for a single domain or for multiple domains. If it is configured to support authentication for multiple domains, you must also specify the domain for Citrix ADC SSO by creating an authentication profile, and then configuring the traffic management virtual server to use that authentication profile.

> **Note**
>
> The traffic management virtual server can be either a load balancing (lb) or content switching (cs) virtual server. The following instructions assume that you are using a load balancing virtual

> server. To configure a content switching virtual server, simply substitute set cs vserver for set lb vserver. The procedure is otherwise the same.

To create the authentication profile, and then configure the authentication profile on a traffic management virtual server, type the following commands:

```
-  add authentication authnProfile <authnProfileName> {
 -authvserverName <string> }
  {
 -authenticationHost <string> }
  {
 -authenticationDomain <string> }

-  set lb vserver \<vserverName\> -authnProfile <authnprofileName>
<!--NeedCopy-->
```

For the variables, substitute the following values:

- **authnprofileName**—A name for the authentication profile. Must begin with a letter, number, or the underscore character (_), and must consist of from one to thirty-one alphanumeric or hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.
- **authvserverName**—The name of the authentication virtual server that this profile uses for authentication.
- **authenticationHost**—Host name of the authentication virtual server.
- **authenticationDomain**—Domain for which Citrix ADC SSO handles authentication. Required if the authentication virtual server performs authentication for more than one domain, so that the correct domain is included when the Citrix ADC appliance sets the traffic management virtual server cookie.

Example:

To create an authentication profile named authnProfile1 for authentication of the example.com domain, and to configure the load balancing virtual server vserver1 to use the authentication profile authnProfile1, you would type the following commands:

```
add authentication authnProfile authnProfile1 -authnvsName
   authvsesrver1
     -authenticationHost authvsesrver1 -authenticationDomain example.
        com
set lb vserver vserver1 -authnProfile authnProfile1
<!--NeedCopy-->
```


Configuring SSO

September 14, 2021

Configuring Citrix ADC SSO to authenticate by impersonation is simpler than configuring than SSO to authenticate by delegation, and is therefore preferable when your configuration allows it. You create a KCD account. You can use the user's password.

If you do not have the user's password, you can configure Citrix ADC SSO to authenticate by delegation. Although more complex than configuring SSO to authenticate by impersonation, the delegation method provides flexibility in that a user's credentials might not be available to the Citrix ADC appliance in all circumstances.

For either impersonation or delegation, you must also enable integrated authentication on the web application server.

Enable integrated authentication on the web application server

To set up Citrix ADC Kerberos SSO on each web application server that Kerberos SSO manages, use the configuration interface on that server to configure the server to require authentication. Select Kerberos (negotiate) authentication by preference, with fallback to NTLM for clients that do not support Kerberos.

Following are instructions for configuring the Microsoft Internet Information Server (IIS) to require authentication. If your web application server uses software other than IIS, consult the documentation for that web server software for instructions.

To configure Microsoft IIS to use integrated authentication

1. Log on to the IIS server and open **Internet Information Services Manager.**
2. Select the website for which you want to enable integrated authentication. To enable integrated authentication for all IIS web servers managed by IISM, configure authentication settings for the Default website. To enable integrated authentication for individual services (such as Exchange, Exadmin, ExchWeb, and Public), configure these authentication settings for each service individually.
3. Open the **Properties** dialog box for the default website or for the individual service, and click the **Directory Security** tab.
4. Beside **Authentication** and **Access Control**, select **Edit.**
5. Disable anonymous access.
6. Enable Integrated Windows authentication (only). Enabling integrated Windows authentication must automatically set protocol negotiation for the web server to Negotiate, NTLM, which spec-

ifies Kerberos authentication with fallback to NTLM for non-Kerberos capable devices. If this option is not automatically selected, manually set protocol negotiation to Negotiate, NTLM.

Set up SSO by impersonation

You can configure the KCD account for Citrix ADC SSO by impersonation. In this configuration, the Citrix ADC appliance obtains the user's user name and password when the user authenticates to the authentication server and uses those credentials to impersonate the user to obtain a ticket-granting ticket (TGT). If the user's name is in UPN format, the appliance obtains the user's realm from UPN. Otherwise, it obtains the user's name and realm by extracting it from the SSO domain used during initial authentication, or from the session profile.

> **Note**
>
> You cannot add a user name with domain if the user name is already added without domain. If the user name with domain is added first followed by the same user name without domain, then the Citrix ADC appliance adds the user name to the user list.

When configuring the KCD account, you must set the realm parameter to the realm of the service that the user is accessing. The same realm is also used as the user's realm if the user's realm cannot be obtained from authentication with the Citrix ADC appliance or from the session profile.

To create the KCD account for SSO by impersonation with a password

At the command prompt, type the following command:

```
add aaa kcdaccount <accountname> -realmStr <realm>

<!--NeedCopy-->
```

For the variables, substitute the following values:

- **accountname**. The KCD account name.
- **realm**. The domain assigned to the Citrix ADC SSO.

Example

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following command:

```
add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab

<!--NeedCopy-->
```

For information on configuring Kerberos impersonation through the Citrix ADC GUI, see Citrix Support.

Configure SSO by delegation

To configure SSO by Delegation, you need to perform the following tasks:

- If you are configuring delegation by delegated user certificate, install the matching CA certificates on the Citrix ADC appliance and add them to the Citrix ADC configuration.
- Create the KCD account on the appliance. The appliance uses this account to obtain service tickets for your protected applications.
- Configure the Active Directory server.

> **Note**
>
> For more information on creating a KCD account and configuring on the NetScaler appliance, refer to the following topics:
>
> - Handling authentication, authorization and auditing with Kerberos/NTLM
> - How Citrix ADC implements Kerberos for client authentication
> - Configuring kerberos authentication on the Citrix ADC appliance

Installing the client CA certificate on the Citrix ADC appliance

If you are configuring Citrix ADC SSO with a client certificate, you must copy the matching CA certificate for the client certificate domain (the client CA certificate) to the Citrix ADC appliance, and then install the CA certificate. To copy the client CA certificate, use the file transfer program of your choice to transfer the certificate and private-key file to the Citrix ADC appliance, and store the files in /nsconfig/ssl.

To install the client CA certificate on the Citrix ADC appliance

At the command prompt, type the following command:

```
1 add ssl certKey <certkeyName> -cert <cert> [(-key <key> [-password]) |
     -fipsKey <fipsKey>][-inform ( DER | PEM )][-expiryMonitor ( ENABLED
     | DISABLED | UNSET ) [-notificationPeriod <positive_integer>]] [-
     bundle ( YES | NO )]
2
3 <!--NeedCopy-->
```

For the variables, substitute the following values:

- **certkeyName**. A name for the client CA certificate. Must begin with an ASCII alphanumeric or underscore (_) character, and must consist of from one to thirty-one characters. Allowed characters include the ASCII alphanumerics, underscore, hash (#), period(.), space, colon (:), at (@),

equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created. If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my cert” or ‘my cert’).

- **cert**. Full path name and file name of the X509 certificate file used to form the certificate-key pair. The certificate file must be stored on the Citrix ADC appliance, in the /nsconfig/ssl/ directory.
- **key**. Full path name and file name of the file that contains the private key to the X509 certificate file. The key file must be stored on the Citrix ADC appliance in the /nsconfig/ssl/ directory.
- **password**. If a private key is specified, the passphrase used to encrypt the private key. Use this option to load encrypted private keys in PEM format.
- **fipsKey**. Name of the FIPS key that was created inside the Hardware Security Module (HSM) of a FIPS appliance, or a key that was imported into the HSM.

  > **Note**
  >
  > You can specify either a key or a fipsKey, but not both.

- **inform**. Format of the certificate and private-key files, either PEM or DER.
- **passplain**. Pass phrase used to encrypt the private key. Required when adding an encrypted private-key in PEM format.
- **expiryMonitor**. Configure the Citrix ADC appliance to issue an alert when the certificate is about to expire. Possible values: ENABLED, DISABLED, UNSET.
- **notificationPeriod**. If expiryMonitor is ENABLED, number of days before the certificate expires to issue an alert.
- **bundle**. Parse the certificate chain as a single file after linking the server certificate to its issuer’s certificate within the file. Possible values: YES, NO.

Example

The following example adds the specified delegated user certificate customer-cert.pem to the Citrix ADC configuration along with the key customer-key.pem, and sets the password, certificate format, expiration monitor, and notification period.

To add the delegated user certificate, you would type the following commands:

```
1 add ssl certKey customer -cert "/nsconfig/ssl/customer-cert.pem"
2 -key "/nsconfig/ssl/customer-key.pem" -password "dontUseDefaultPWs!"
3 -inform PEM -expiryMonitor ENABLED [-notificationPeriod 14]
4
5 <!--NeedCopy-->
```


Creating the KCD account

If you are configuring Citrix ADC SSO by delegation, you can configure the KCD account to use the user's log-on name and password, to use the user's log-on name and keytab, or to use the user's client certificate. If you configure SSO with user name and password, the Citrix ADC appliance uses the delegated user account to obtain a Ticket Granting Ticket (TGT), and then uses the TGT to obtain service tickets for the specific services that each user requests. If you configure SSO with keytab file, the Citrix ADC appliance uses the delegated user account and keytab information. If you configure SSO with a delegated user certificate, the Citrix ADC appliance uses the delegated user certificate.

To create the KCD account for SSO by delegation with a password

At the command prompt, type the following commands:

```
add aaa kcdAccount <kcdAccount> {
 -keytab <string> }
  {
 -realmStr <string> }
  {
 -delegatedUser <string> }
  {
 -kcdPassword  }
  {
 -usercert <string> }
  {
 -cacert <string> }
  [-userRealm <string>]
[-enterpriseRealm <string>] [-serviceSPN <string>]
<!--NeedCopy-->
```

For the variables, substitute the following values:

- **kcdAccount** - A name for the KCD account.This is a mandatory argument. Maximum Length: 31
- **keytab** - The path to the keytab file. If specified other parameters in this command need not be given. Maximum Length: 127
- **realmStr** - The realm of Kerberos. Maximum Length: 255
- **delegatedUser** - Username that can perform kerberos constrained delegation. Maximum Length: 255
- **kcdPassword** - Password for Delegated User. Maximum Length: 31
- **usercert** - SSL Cert (including private key) for Delegated User. Maximum Length: 255
- **cacert** - CA Cert for UserCert or when doing PKINIT backchannel. Maximum Length: 255

- **userRealm** - Realm of the user. Maximum Length: 255
- **enterpriseRealm** - Enterprise Realm of the user. This should be given only in certain KDC deployments where KDC expects Enterprise username instead of Principal Name. Maximum Length: 255
- **serviceSPN** - Service SPN. When specified, this will be used to fetch kerberos tickets. If not specified, Citrix ADC will construct SPN using service fqdn. Maximum Length: 255

Example (UPN Format)

To add a KCD account named kcdaccount1 to the Citrix ADC appliance configuration with a password of password1 and a realm of EXAMPLE.COM, specifying the delegated user account in UPN format (as root), you would type the following commands:

```
1 add aaa kcdaccount kcdaccount1 -delegatedUser root
2 -kcdPassword password1 -realmStr EXAMPLE.COM
3
4 <!--NeedCopy-->
```

Example (SPN Format)

To add a KCD account named kcdaccount1 to the Citrix ADC appliance configuration with a password of password1 and a realm of EXAMPLE.COM, specifying the delegated user account in SPN format, you would type the following commands:

```
1 add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM
2 -delegatedUser "host/kcdvserver.example.com" -kcdPassword password1
3
4 <!--NeedCopy-->
```

Creating the KCD account for SSO by delegation with a keytab

If you plan to use a keytab file for authentication, first create the keytab. You can create the keytab file manually by logging on to the AD server and using the ktpass utility, or you can use the Citrix ADC configuration utility to create a batch script, and then run that script on the AD server to generate the keytab file. Next, use FTP or another file transfer program to transfer the keytab file to the Citrix ADC appliance and place it in the /nsconfig/krb directory. Finally, configure the KCD account for Citrix ADC SSO by delegation and provide the path and file name of the keytab file to the Citrix ADC appliance.

To create the keytab file manually

Log on to the AD server command line and, at the command prompt, type the following command:

“‘ktpass princ ptype KRB5_NT_PRINCIPAL mapuser pass -out


```
For the variables, substitute the following values:

-  **SPN**. The service principal name for the KCD service account.
-  **DOMAIN**. The domain of the Active Directory server.
-  **username**. The KSA account user name.
-  **password**. The KSA account password.
-  **path**. The full path name of the directory in which to store the
   keytab file after it is generated.

##### To use the Citrix ADC configuration utility to create a script to
     generate the keytab file

1.  Navigate to **Security > AAA - Application Traffic.**
1.  In the data pane, under **Kerberos Constrained Delegation**, click
   **Batch** file to generate Keytab.
1.  In the **Generate KCD (Kerberos Constrained Delegation) Keytab
   Script** dialog box, set the following parameters:
    -  **Domain User Name**. The KSA account user name.
    -  **Domain Password**. The KSA account password.
    -  **Service Principal**. The service principal name for the KSA.
    -  **Output File Name**. The full path and file name to which to
       save the keytab file on the AD server.
1.  Clear the **Create Domain User Account** check box.
1.  Click **Generate Script**.
1.  Log on to the Active Directory server and open a command line
   window.
1.  Copy the script from the **Generated Script** window and paste it
   directly into the Active Directory server command-line window. The
   keytab is generated and stored in the directory under the file name
   that you specified as **Output File Name**.
1.  Use the file transfer utility of your choice to copy the keytab
   file from the Active Directory server to the Citrix ADC appliance
   and place it in the /nsconfig/krb directory.

##### To create the KCD account

At the command prompt, type the following command:
```

add aaa kcdaccount –keytab

```
Example

To add a KCD account named kcdccount1, and use the keytab named
   kcdvserver.keytab, you would type the following commands:
```


add aaa kcdaccount kcdaccount1 –keytab kcdvserver.keytab

```
#### To create the KCD account for SSO by delegation with a delegated
    user cert

At the command prompt, type the following command:
```

add aaa kcdaccount -realmStr -delegatedUser -usercert -cacert

```
For the variables, substitute the following values:

-  **accountname**. A name for the KCD account.
-  **realmStr**. The realm for the KCD account, usually the domain for
    which SSO is configured.
-  **delegatedUser**. The delegated user name, in SPN format.
-  **usercert**. The full path and name of the delegated user
    certificate file on the Citrix ADC appliance. The delegated user
    certificate must contain both the client certificate and the private
     key, and must be in PEM format. If you use smart card
    authentication, you might must create a smart card certificate
    template to allow certificates to be imported with the private key.
-  **cacert**. The full path to and name of the CA certificate file on
    the Citrix ADC appliance.

Example

To add a KCD account named kcdccount1, and use the keytab named
    kcdvserver.keytab, you would type the following command:
```

add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
-delegatedUser “host/kcdvserver.example.com” -usercert /certs/usercert
-cacert /cacerts/cacert

“‘

Setting up Active Directory for Citrix ADC SSO

When you configure SSO by delegation, in addition to creating the KCDAccount on the Citrix ADC appliance, you must also create a matching Kerberos Service Account (KSA) on your LDAP active directory server, and configure the server for SSO. To create the KSA, use the account creation process on the active directory server. To configure SSO on the active directory server, open the properties window for the KSA. In the **Delegation** tab, enable the following options: Trust this user for delegation to specified services only and Use any Authentication protocol. (The Kerberos only option does not work,

because it does not enable protocol transition or constrained delegation.) Finally, add the services that Citrix ADC SSO manages.

> **Note**
>
> If the Delegation tab is not visible in the KSA account properties dialog box, before you can configure the KSA as described, you must use the Microsoft
> setspn command-line tool to configure the active directory server so that the tab is visible.

To configure delegation for the Kerberos service account

1. In the LDAP account configuration dialog box for the Kerberos service account that you created, click the **Delegation** tab.
2. Choose “Trust this user for delegation to the specified services only”.
3. Under “Trust this user for delegation to the specified services only,” choose “Use any authentication protocol”.
4. Under “Services to which this account can present delegated credentials,” click **Add.**
5. In the **Add Services** dialog box, click **Users** or **Computers**, choose the server that hosts the resources to be assigned to the service account, and then click **OK.**

   > **Note**
   >
   > - Constrained delegation does not support services hosted in domains other than the domain assigned to the account, even though Kerberos might have a trust relationship with other domains.
   > - Use the following command to create the setspn if a new user is created in active directory: setspn -A host/kcdvserver.example.com example\kcdtest

6. Back in the **Add Services** dialog box, in the Available Services list, chooses the services assigned to the service account. Citrix ADC SSO supports the HTTP and MSSQLSVC services.
7. Click **OK.**

Points to note when advanced encryptions is used to configure KCD account

> - **Sample configuration when keytab is used:** add kcdaccount lbvs_keytab_aes256 -keytab “/nsconfig/krb/kcd2_aes256.keytab”
> - **Use the following command when keytab has multiple encryption types.** The command additionally captures domain user parameters: add kcdaccount lbvs_keytab_aes256 -keytab “/nsconfig/krb/kcd2_aes256.keytab” –domainUser “HTTP/lbvs.aaa.local”
> - **Use the following commands when user credential are used:** add kcdaccount kslb2_user -realmStr AAA.LOCAL -delegatedUser lbvs -kcdPassword <password>

> • Ensure that the correct **domainUser** information is provided. You can look for the user logon name in AD.

Generate the KCD keytab script

September 14, 2021

The KCD Keytab Script dialog box generates the keytab script, which in turn generates the keytab file necessary to configure KCD on the Citrix ADC.

To generate the KCD keytab script by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic.**
2. In the details pane, under **Kerberos Constrained Delegation**, click Batch file to generate keytab.
3. In the Generate KCD (Kerberos Constrained Delegation) **Keytab Script** dialog box, fill out the fields as described below.
   - **Domain User Name:** The name of the domain user.
   - **Domain Password:** The password for the domain user.
   - **Service Principal:** The service principal.
   - **Output File Name:** A filename for the KCD script file.
   - **Create Domain User Account:** Select this check box to create the specified domain user account.
4. Click **Generate Script** to generate the script. The script is generated, and appears in the **Generated Script** text box below the **Generate Script** button.
5. Copy the script, and save it as a file on your AD domain controller. You must now run this script on the domain controller to generate the keytab file, and then copy the keytab file to the /nsconfig/krb/ directory on the Citrix ADC appliance.
6. Click **OK.**

Enable SSO for Basic, Digest, and NTLM authentication

September 14, 2021

From Citrix ADC feature release 13.0 build 64.35 and above, the following SSO types are disabled globally.

- Basic authentication
- Digest Access authentication

- NTLM without Negotiate NTLM2 Key or Negotiate Sign

Single Sign-On (SSO) configuration in Citrix ADC and Citrix Gateway can be enabled at global level and also per traffic level. By default the SSO configuration is **OFF** and an administrator can enable the SSO per traffic or globally. From a security point of view, Citrix recommends administrators to turn SSO globally **OFF** and enable per traffic basis. This enhancement is to make SSO configuration more secure by disabling certain type of SSO methods globally.

> StoreFront SSO configuration is impacted (disabled) only for 13.0 build 64.35. The configuration will not be impacted in the future 13.0 builds.

Non-impacted SSO types

The following SSO types are not impacted with this enhancement.

- Kerberos authentication
- SAML authentication
- Form based authentication
- OAuth bearer authentication
- NTLM with Negotiate NTLM2 Key or Negotiate Sign

Impacted SSO configurations

Following are the impacted (disabled) SSO configurations.

Global configurations

```
set tmsessionparam -SSO ON
set vpnparameter -SSO ON
add tmsessionaction tm_act -SSO ON
add vpn sessionaction tm_act -SSO ON
```

> You can enable/disable SSO as a whole and cannot modify individual SSO types.

Security measures to be applied

As part of the security measures, security sensitive SSO types are disabled in the global configuration but are allowed only through a Traffic action configuration.
So, if a back-end server expects Basic, Digest, or NTLM without Negotiate NTLM2 Key or Negotiate Sign, the administrator can allow SSO only through the following configuration.

Traffic Action

```
add vpn trafficaction tf_act  http -SSO ON
add tm trafficaction tf_act -SSO ON
```

Traffic Policy

```
add tm trafficpolicy <name> <rule> tf_act
add vpn trafficpolicy <name> <rule> tf-act
```

> Administrator must have an appropriate rule configured for traffic policy to make sure SSO is enabled for only trusted back-end server.

AAA-TM

Scenarios based on global configuration:

```
set tmsessionparam -SSO ON
```

Workaround:

```
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol true tf_act
```

Bind the following traffic policy to all LB virtual server where SSO is expected:

```
bind lb vserver <LB VS Name> -policy tf_pol -priority 65345
```

Scenarios based on session Policy configuration:

```
add tmsessionaction tm_act -SSO ON
add tmsession policy <name> <rule> tm_act
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol <same rule as session Policy> tf_act
```

Points of note:

- Citrix ADC AAA user/group for the preceding session policy must be replaced by traffic policy.
- Bind the following policy to the load balancing virtual servers for the preceding session policy, `bind lb vserver [LB VS Name] -policy tf_pol -priority 65345`
- If a traffic policy with other priority is configured, the preceding command does not serve good.

The following section deals with scenarios based on conflict with multiple traffic policies associated with a traffic:

For a particular TM traffic, only one TM traffic policy is applied. Because of global setting of SSO feature changes, applying an additional TM traffic policy with low priority might not be applicable in case a TM traffic policy with high priority (that does not have required SSO configuration) is already applied. The following section describes method to ensure that such cases are handled.

Consider that the following three traffic policies with higher priority are applied to load balancing (LB) virtual server:

```
add tm trafficaction tf_act1 <Addition config>
add tm trafficaction tf_act2 <Addition config>
add tm trafficaction tf_act3 <Addition config>

add tm trafficpolicy tf_pol1 <rule1> tf_act1
add tm trafficpolicy tf_pol2 <rule2> tf_act2
add tm trafficpolicy tf_pol3 <rule3> tf_act3

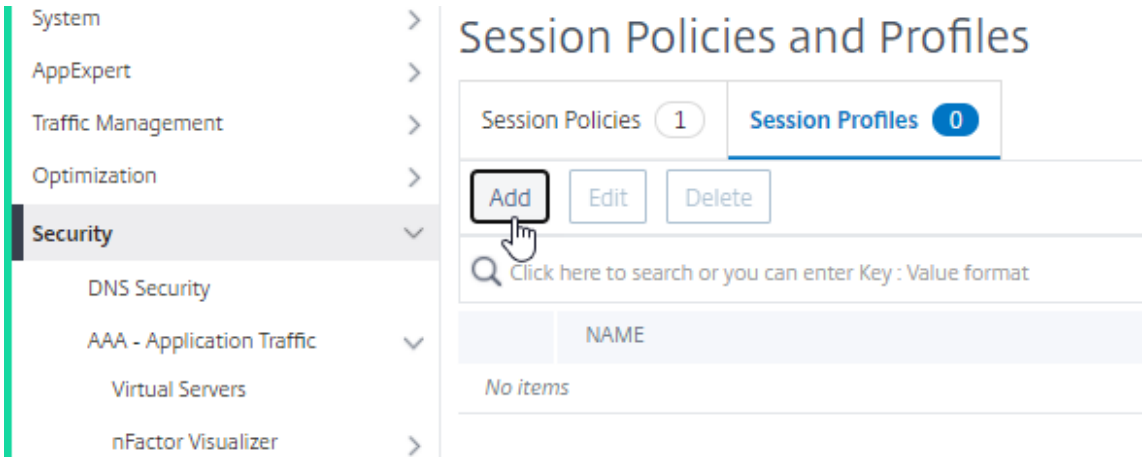
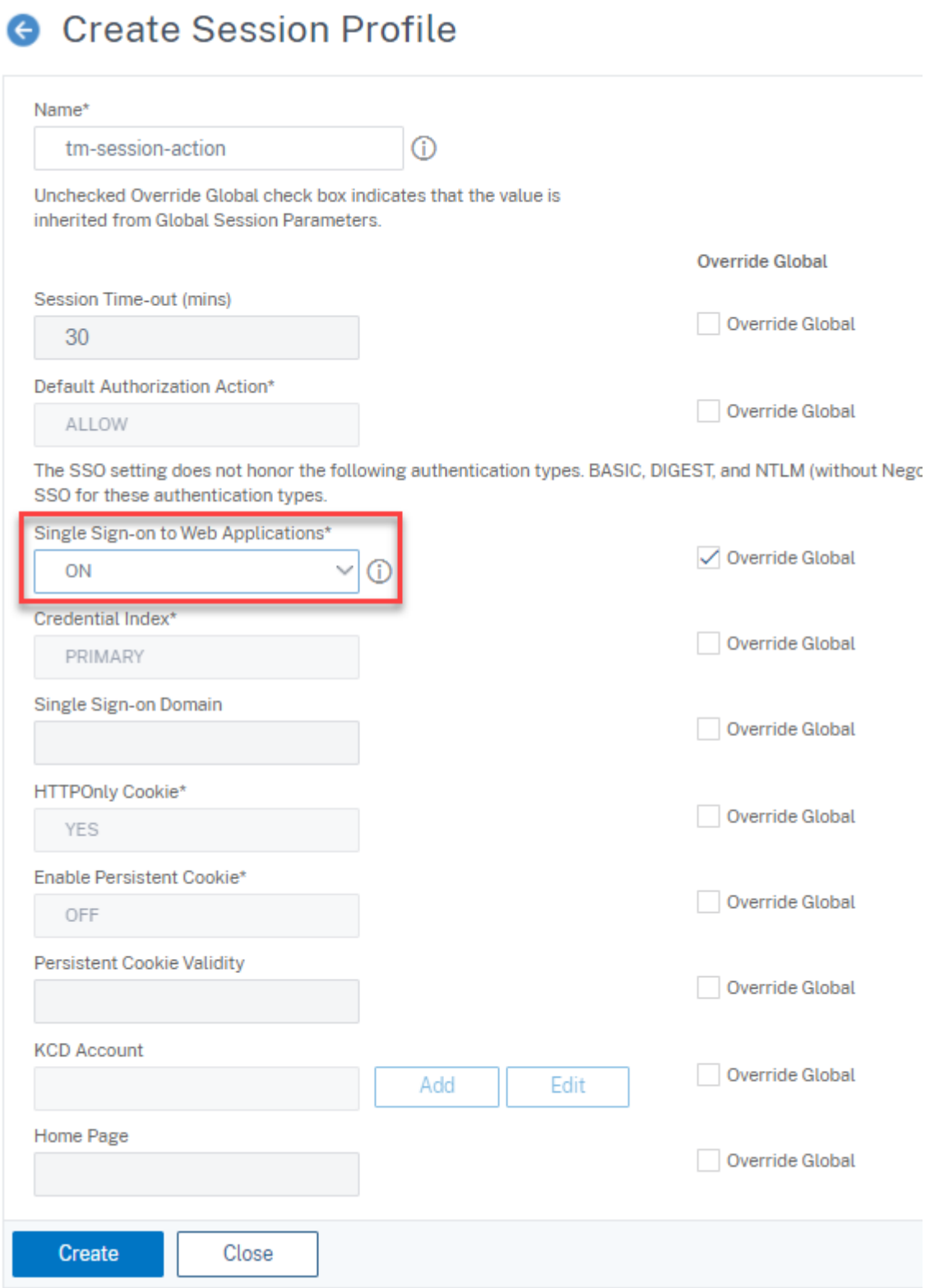
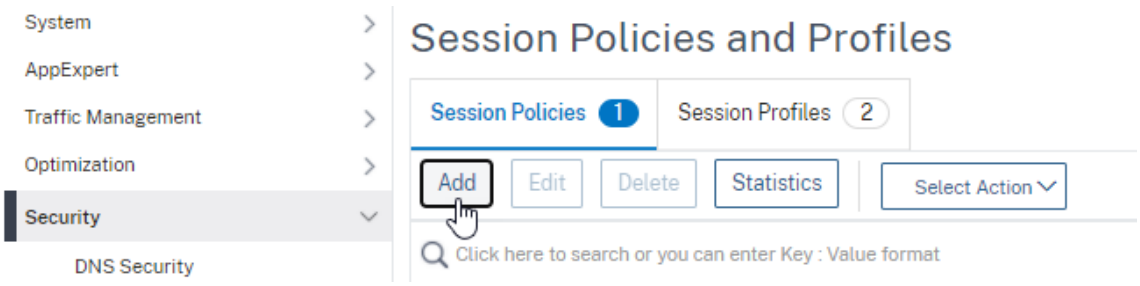
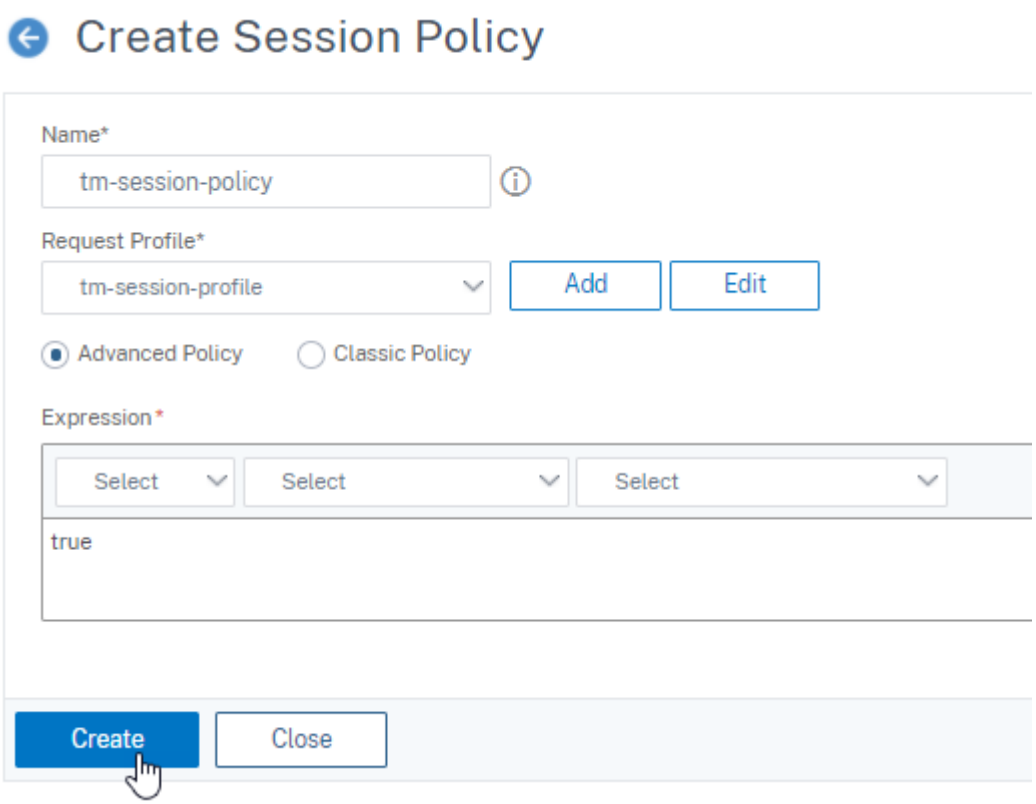
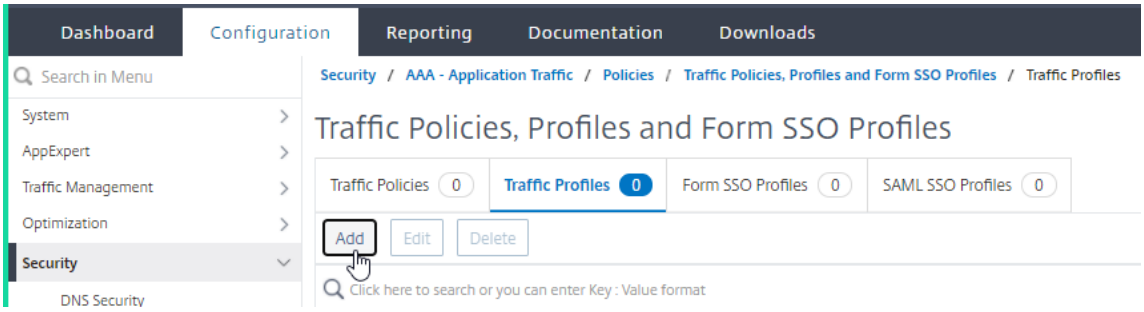
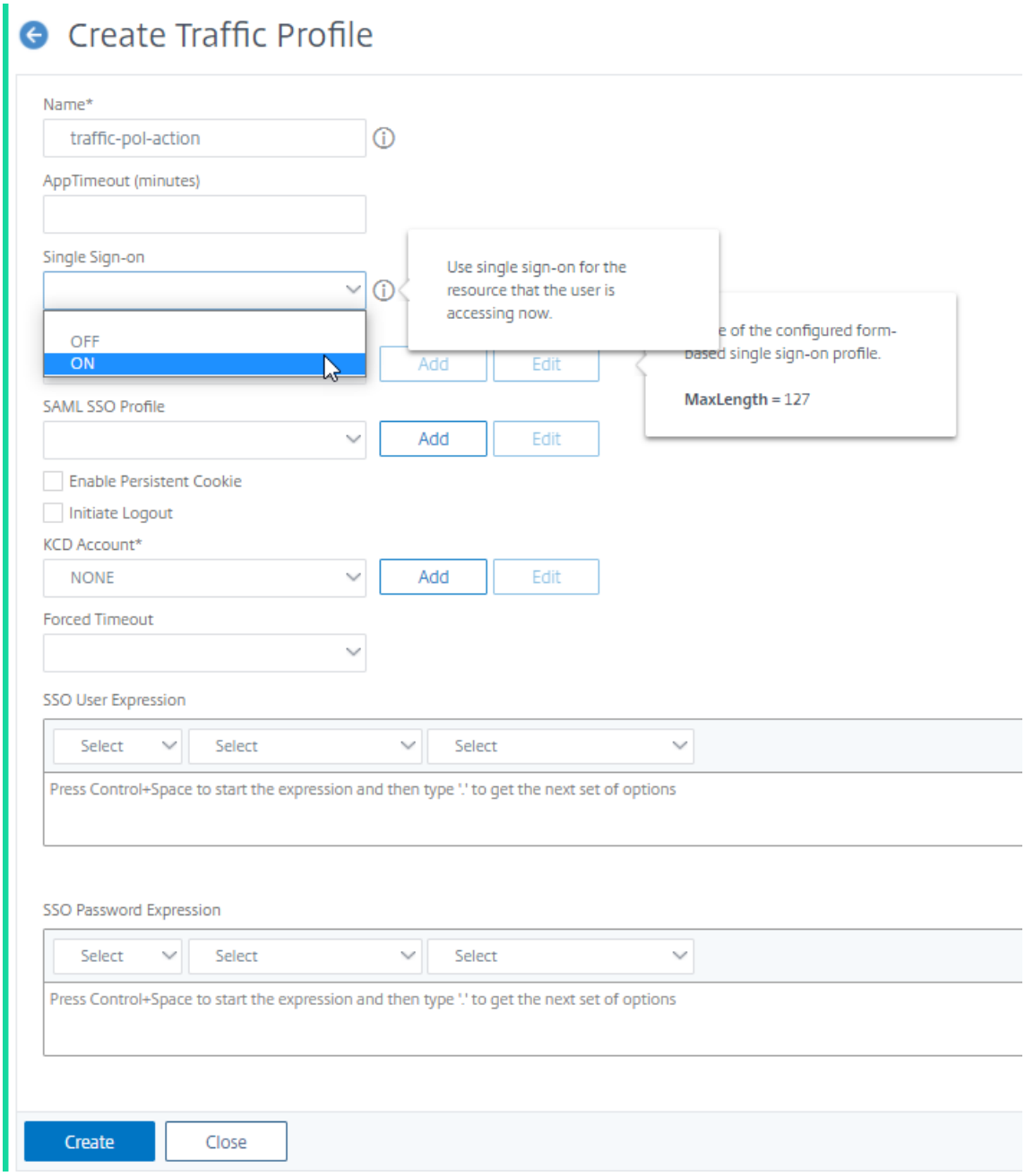
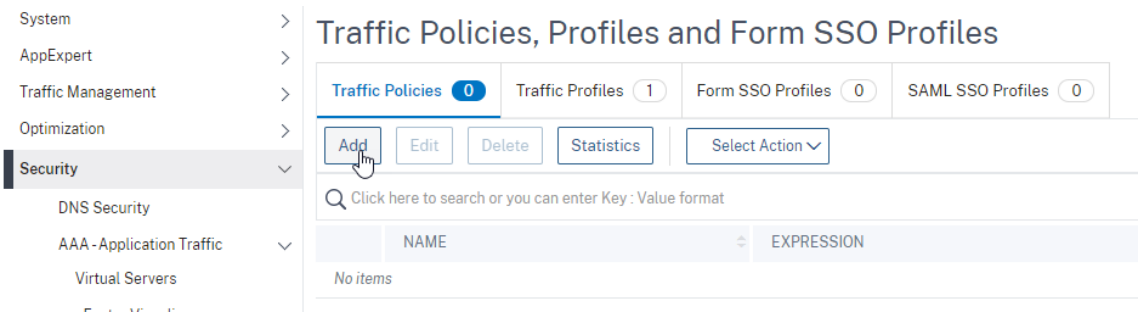
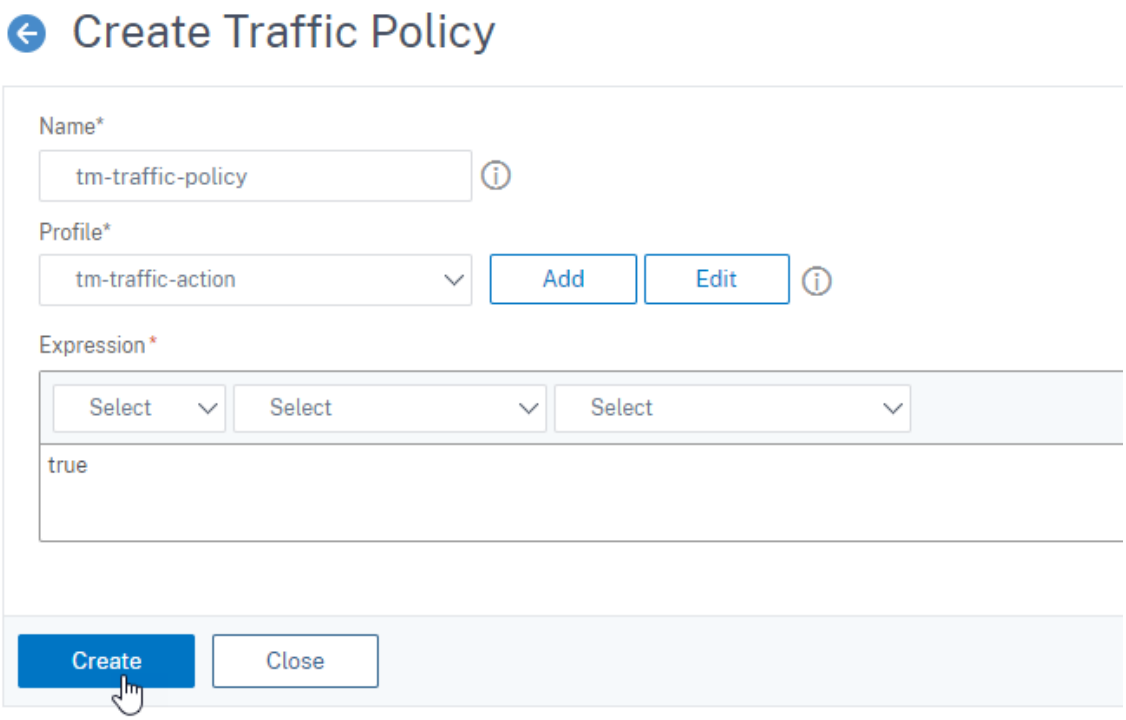
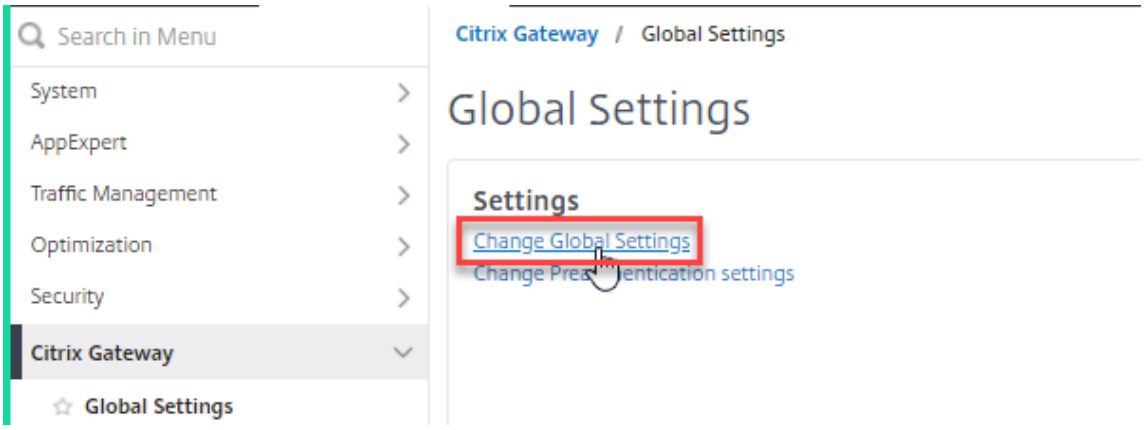
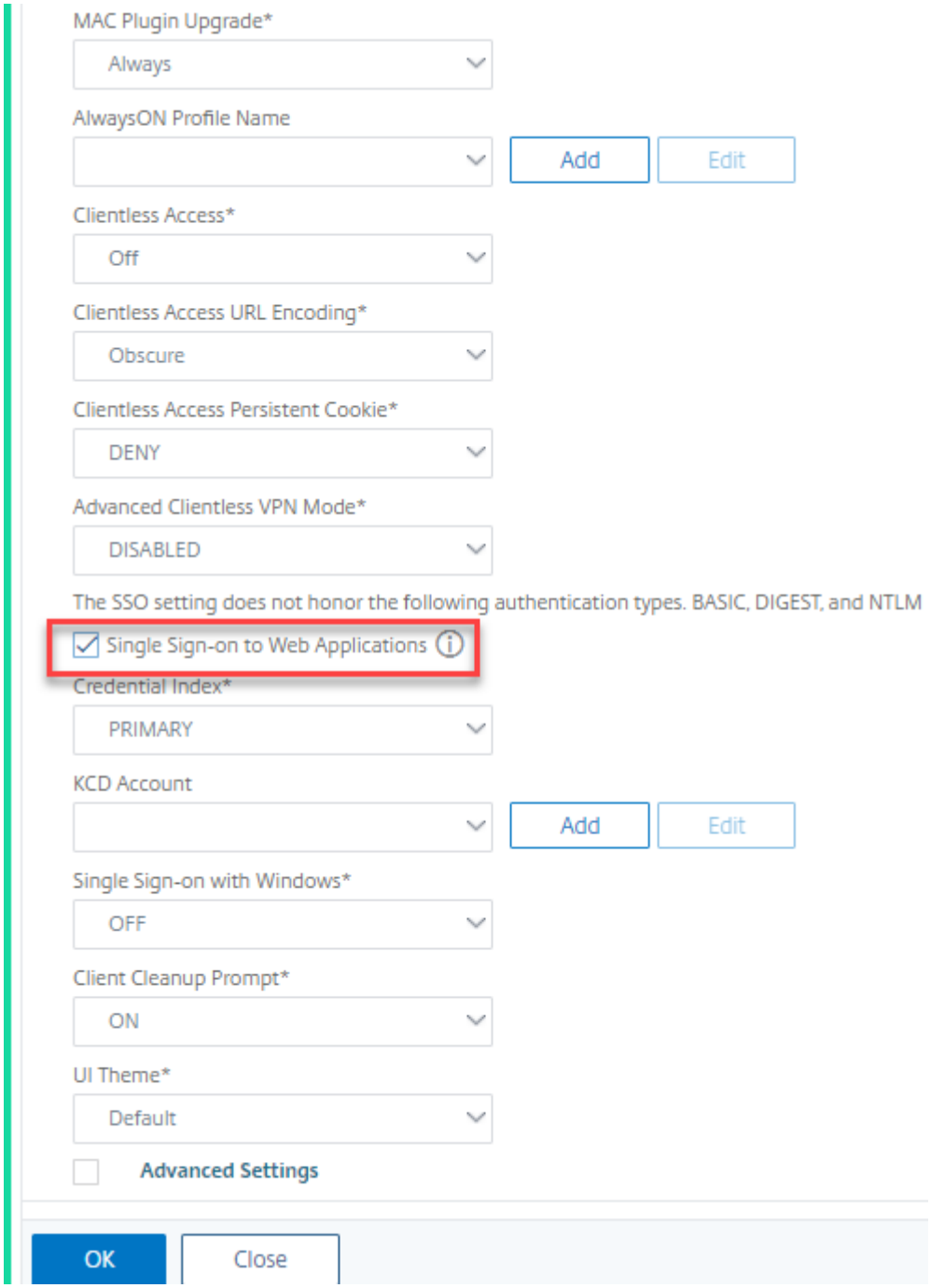
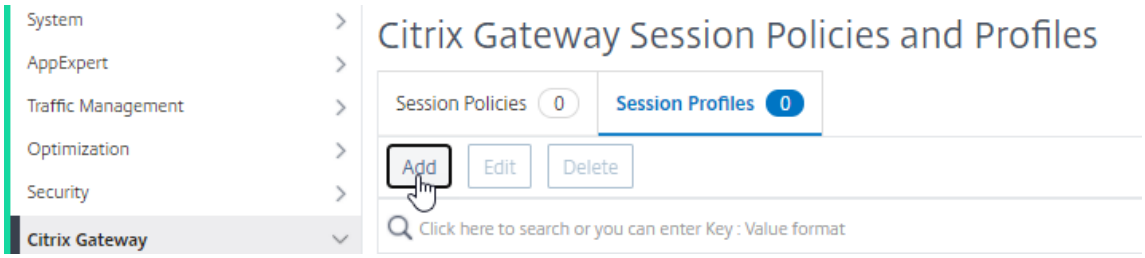
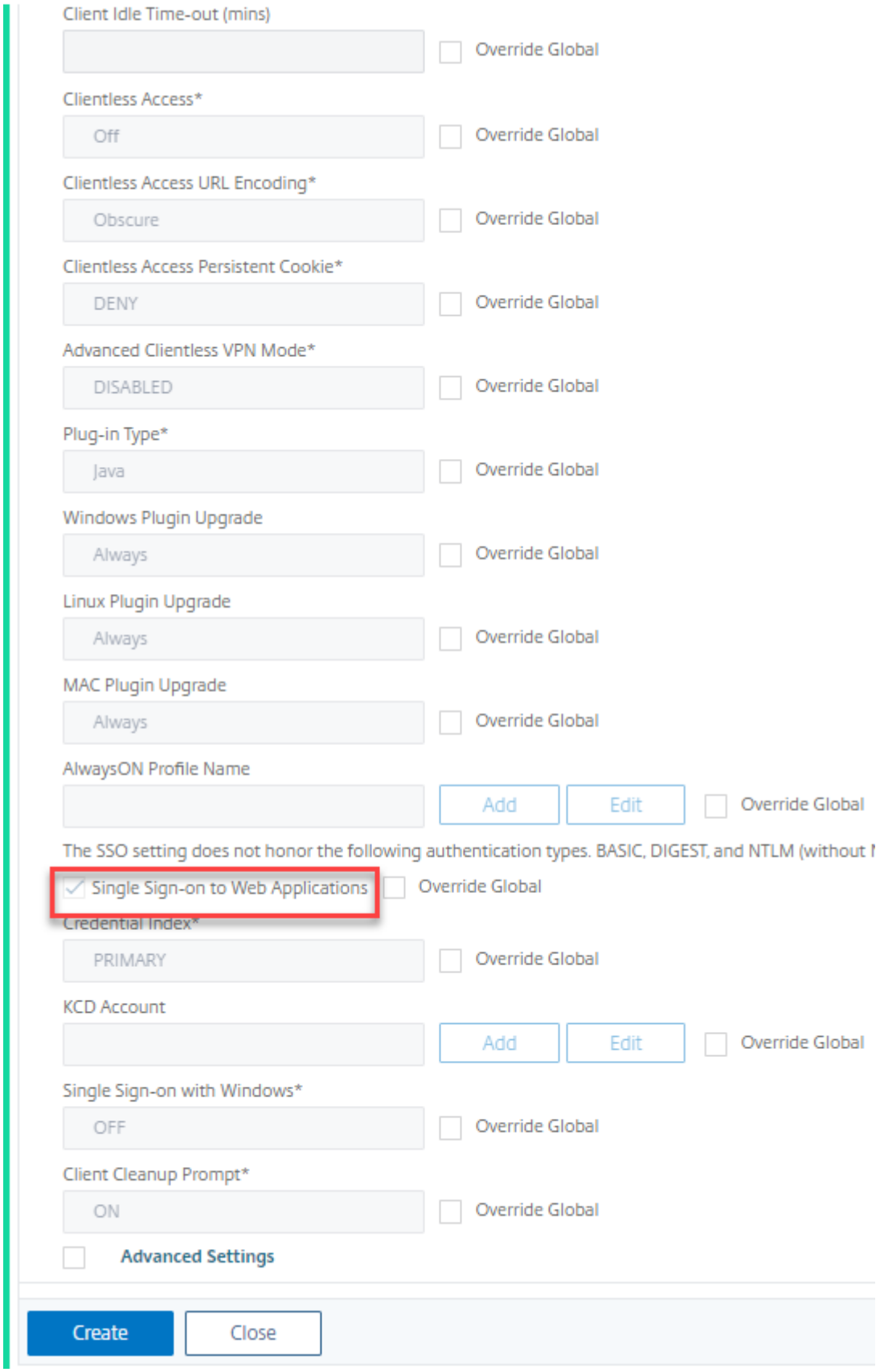
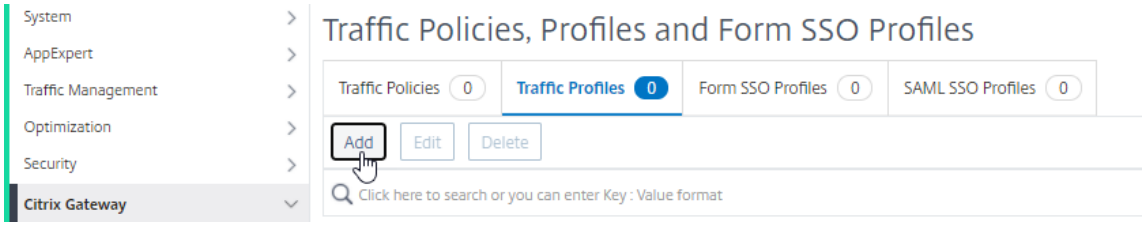
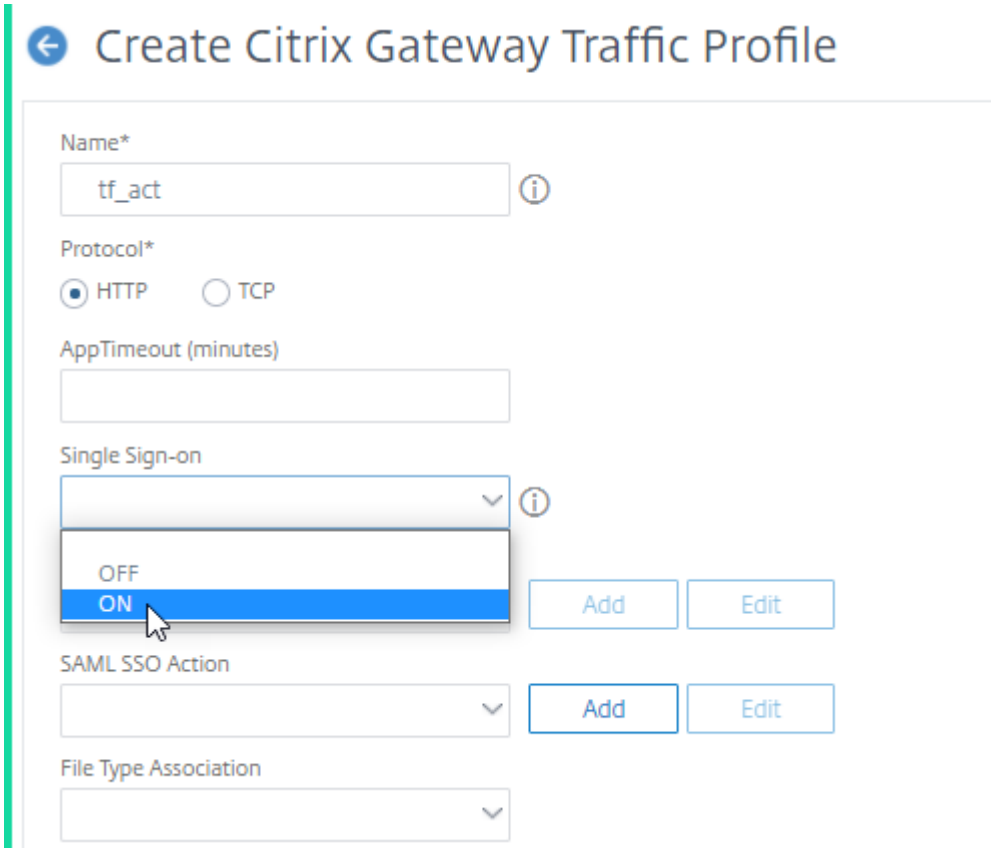
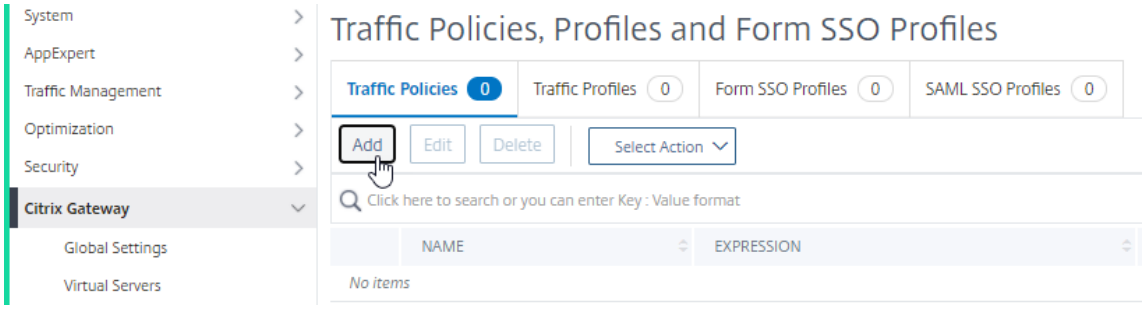
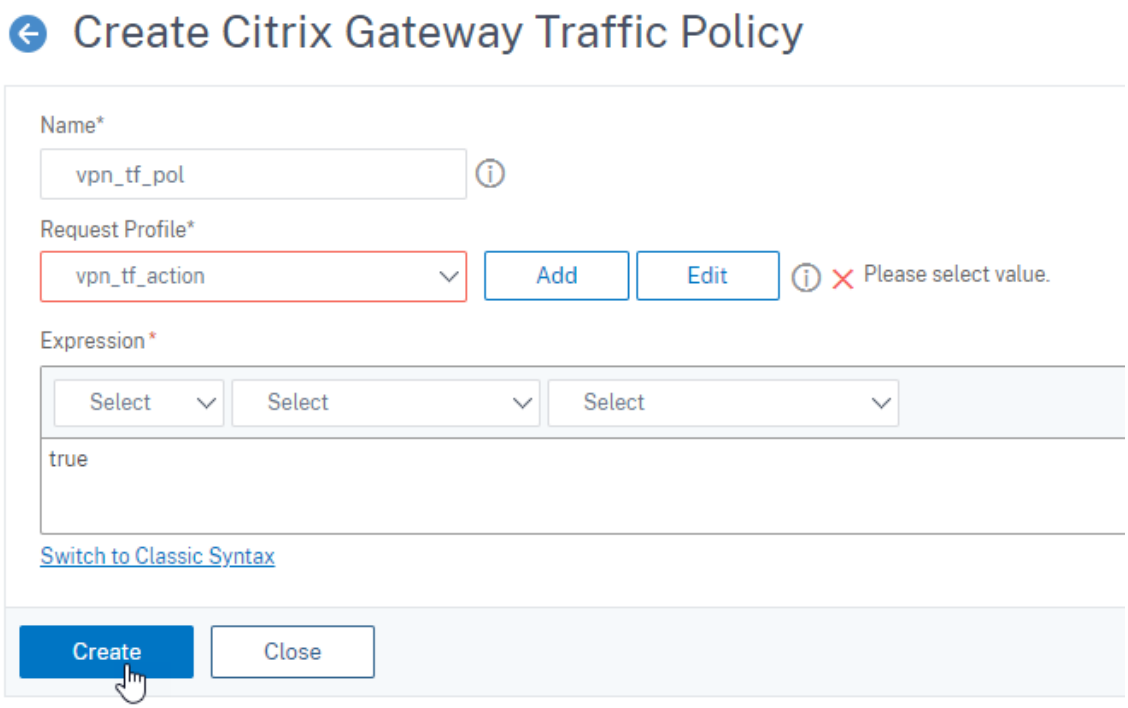
bind lb vserver <LB VS Name> -policy tf_pol1 -priority 100
bind lb vserver <LB VS Name> -policy tf_pol2 -priority 200
bind lb vserver <LB VS Name> -policy tf_pol3 -priority 300
```

Error prone method - To resolve the Global SSO configuration, you add the following configuration:

```
add tm trafficaction tf_act_default -SSO ON
add tm trafficpolicy tf_pol_default true tf_act_default

bind lb vserver <LB VS Name> -policy tf_pol_default -priority 65345
```

Note: The preceding modification can break SSO for traffic which hits <tf_pol1/tf_pol2/tf_pol3> as for these traffic, traffic policy is not be applied.

Correct method - To mitigate this, the SSO property must be applied individually for each of the corresponding traffic actions:

For example, in the preceding scenario, for SSO to happen for the traffic hitting tf_pol1/tf_pol3, the following configuration must be applied along with .

```
add tm trafficaction tf_act1 <Addition config> -SSO ON
add tm trafficaction tf_act3 <Addition config> -SSO ON
```

Citrix Gateway cases

Scenarios based on global configuration:

```
set vpnparameter -SSO ON
```


Workaround:

```
add vpn trafficaction vpn_tf_act http  -SSO ON
add vpn trafficpolicy vpn_tf_pol  true vpn_tf_act
```

Bind the following traffic policy to all VPN virtual server where SSO is expected:

```
bind vpn vserver vpn_vs -policy vpn_tf_pol -priority 65345
```

Scenarios based on session Policy configuration:

```
add vpn sessionaction vpn_sess_act -SSO ON
add vpnsession policy <name> <rule> vpn_sess_act
```

Points to note:

- Citrix ADC AAA user/group for the preceding session policy must be replaced by traffic policy.
- Bind the following policy to the LB virtual servers for the preceding session policy, `bind lb virtual server [LB VS Name] -policy tf_pol -priority 65345`.
- If a traffic policy with other priority is configured, the preceding command does not serve good. The following section deals with scenarios based on conflict with multiple traffic policies associated with traffic.

Functional scenarios based on conflict with multiple traffic policies associated with a traffic:

For a particular Citrix Gateway traffic, only one VPN traffic policy is applied. Because of global setting of SSO feature changes, applying an additional VPN traffic policy with low priority might not be applicable if there are other VPN traffic policies with high priority that does not have a required SSO configuration.

The following section describes method to ensure such cases are handled:

Consider there are three traffic policies with higher priority applied to a VPN virtual server:

```
add vpn trafficaction tf_act1 <Addition config>
add vpn trafficaction tf_act2 <Addition config>
add vpn trafficaction tf_act3 <Addition config>

add vpn trafficpolicy tf_pol1 <rule1> tf_act1
add vpn trafficpolicy tf_pol2 <rule2> tf_act2
add vpn trafficpolicy tf_pol3 <rule3> tf_act3

bind vpn vserver <VPN VS Name> -policy tf_pol1 -priority 100
bind vpn vserver <VPN VS Name> -policy tf_pol2 -priority 200
bind vpn vserver <VPN VS Name> -policy tf_pol3 -priority 300
```


Error prone method: To resolve the Global SSO configuration, you add the following configuration:

```
add vpn trafficaction tf_act_default -SSO ON
add vpn trafficpolicy tf_pol_default true tf_act_default

bind vpn vserver <VPN VS Name> -policy tf_pol_default -priority 65345
```

Note: The preceding modification can break SSO for the traffic that hits <tf_pol1/tf_pol2/tf_pol3> as for these traffic, traffic policy is not applied.

Correct method: To mitigate this, the SSO property must be applied individually for each of the corresponding traffic actions.

For example in the preceding scenario, for SSO to happen for traffic hitting tf_pol1/tf_pol3, the following configuration must be applied along with .

```
add vpn trafficaction tf_act1 [Additional config] -SSO ON

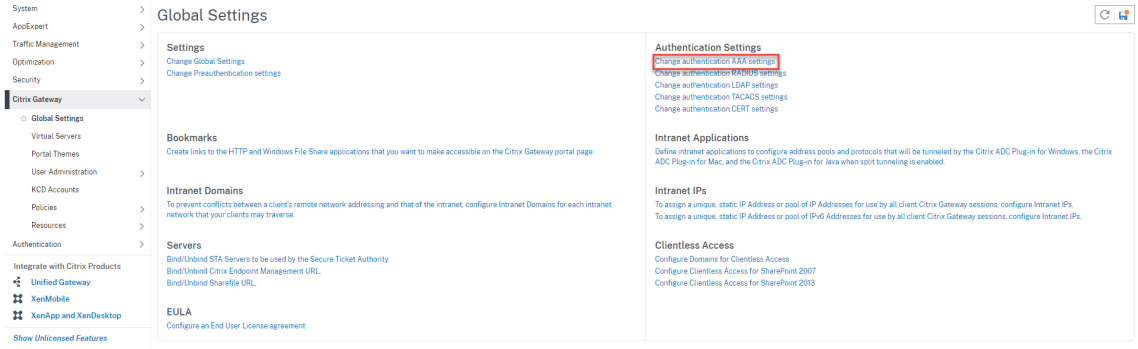
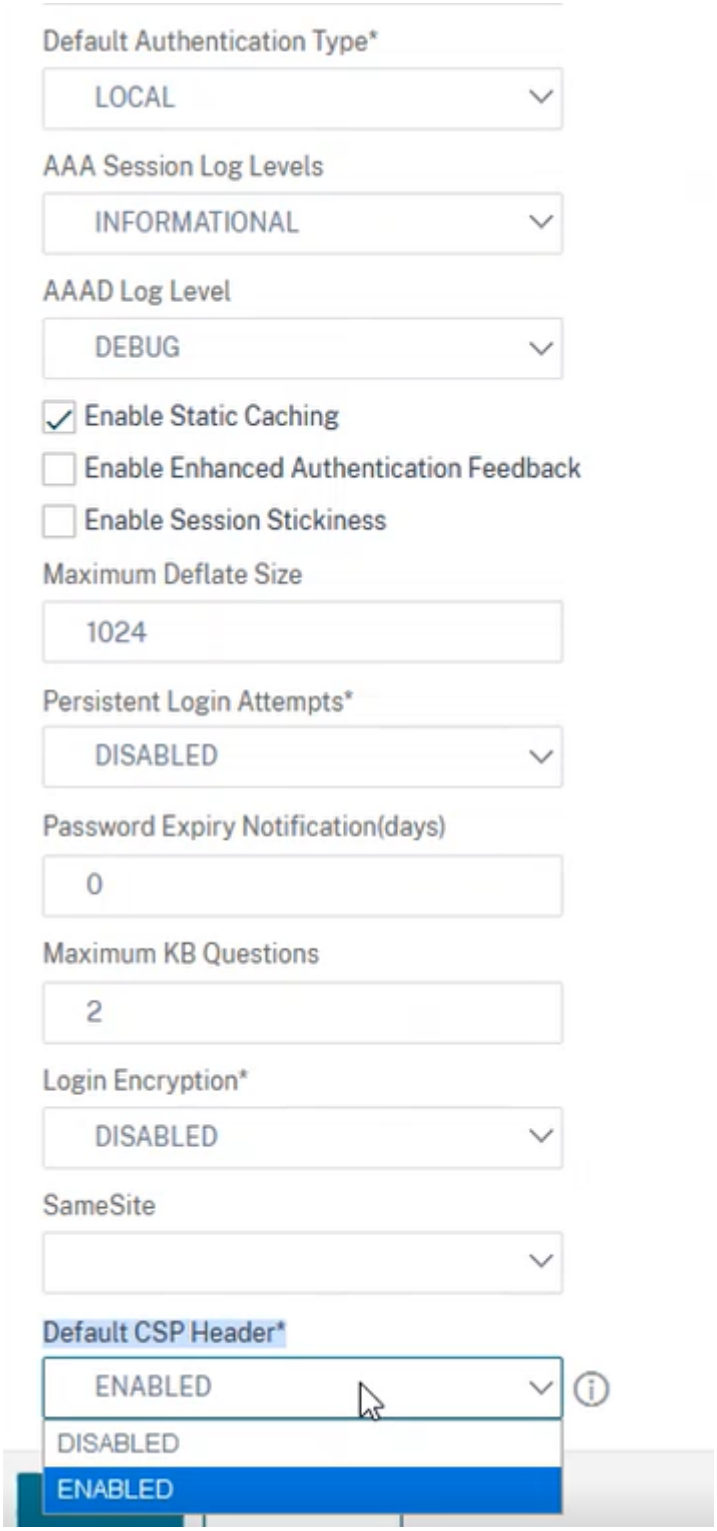
add vpn trafficaction tf_act3 [Additional config] -SSO ON
```

Configure SSO using GUI

1. Navigate to **Security > AAA – Application Traffic > Policies > Session**, Select **Session Profiles** tab, and click **Add**.



2. Enter a name for the session profile, click **Override Global** check box next to **Single Sign-on to Web Applications** field, and click **Create**.



3. Navigate to **Security > AAA – Application Traffic > Policies > Session**, Select **Session Policies** tab, and click **Add**.



4. Enter a name for the session policy, enter “True” in the **Expression** field and click **Create**.



5. Navigate to **Security > AAA – Application Traffic > Policies > Traffic**, Select **Traffic Profiles** tab, and click **Add**.



6. Enter a name for the traffic profile, select **ON** in the **Single Sign-on** drop-down menu, and click **Create**.



7. Navigate to **Security > AAA – Application Traffic > Policies> Traffic**, Select **Traffic Policies** tab, and click **Add**.



8. Enter a name for the traffic policy, enter “True” in the **Expression** field and click **Create**.



9. Navigate to **Citrix Gateway > Global Settings**, and click **Change Global Settings**.



10. on **Global Citrix Gateway Settings** page, select **Client Experience** tab, and check **Single Sign-on to Web Applications** field.



11. Navigate to **Citrix Gateway > Policies> Session**, Select **Session Profiles** tab, and click **Add**.



12. On **Create Citrix Gateway Session Profile** page, select **Client Experience** tab, and check **Single Sign-on to Web Applications** field.



13. Navigate to **Citrix Gateway > Policies > Traffic**, select **Traffic Profiles** tab, and click **Add**.



14. Enter a name for the traffic profile, select **ON** in the **Single Sign-on** drop-down menu, and click **Create**.



15. Navigate to **Citrix Gateway > Policies> Traffic**, Select **Traffic Policies** tab, and click **Add**.



16. On **Create Citrix Gateway Traffic Policy** page, enter name for the traffic policy, enter “True” in the **Expression** field and click **Create**.



Rewrite for Citrix Gateway and authentication server generated responses

September 14, 2021

Rewrite refers to the rewriting of some information in the requests or responses handled by the Citrix ADC appliance. Rewriting can help in providing access to the requested content without exposing unnecessary details about the website's actual configuration. For a detailed information on rewrite concept, see Rewrite

Starting from Citrix ADC release build 13.0-76.29, the support for rewrite policies has been extended to Citrix Gateway virtual server and authentication virtual server generated responses.

> **Note**
>
> A bind type **AAA_Response** is introduced to support rewrite policies for Citrix Gateway virtual server and authentication virtual server generated responses.

An example to use Rewrite

You can use Rewrite to share the resources available on on-premises Citrix ADC with Citrix Cloud deployment. This can be achieved securely by implementing CORS origin resource sharing. Rewrite can be used as follows to implement CORS header.

Sample configuration

```
add rewrite action cors_header_action insert_http_header access-control
    -allow-credentials \"true\"

add rewrite policy cors_header_pol true cors_header_action

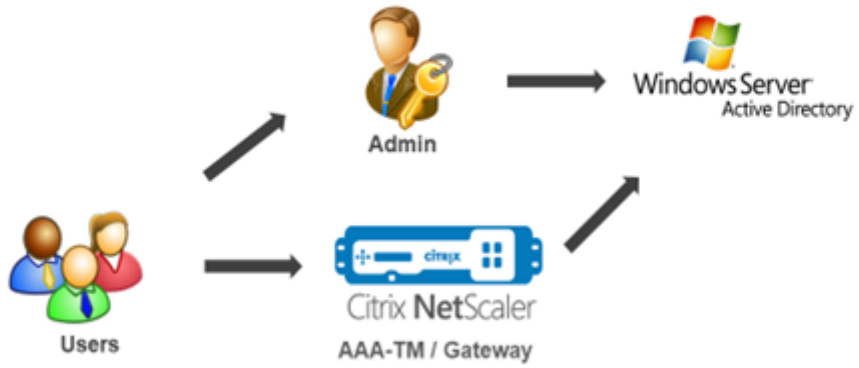
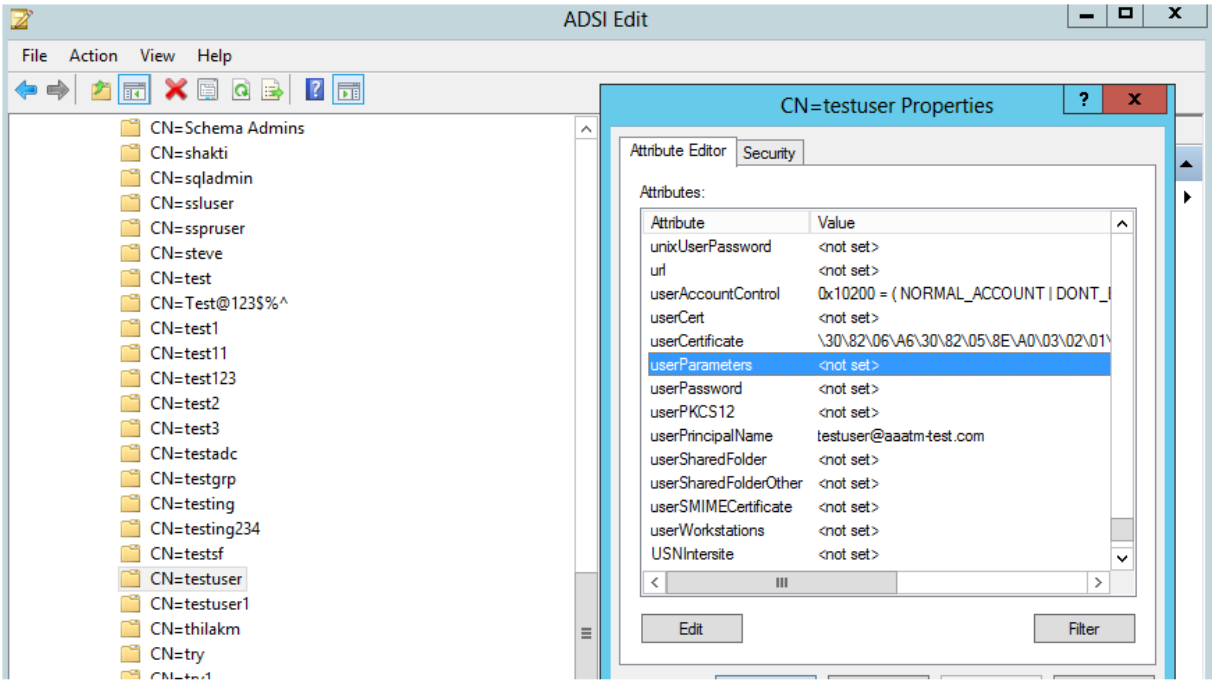
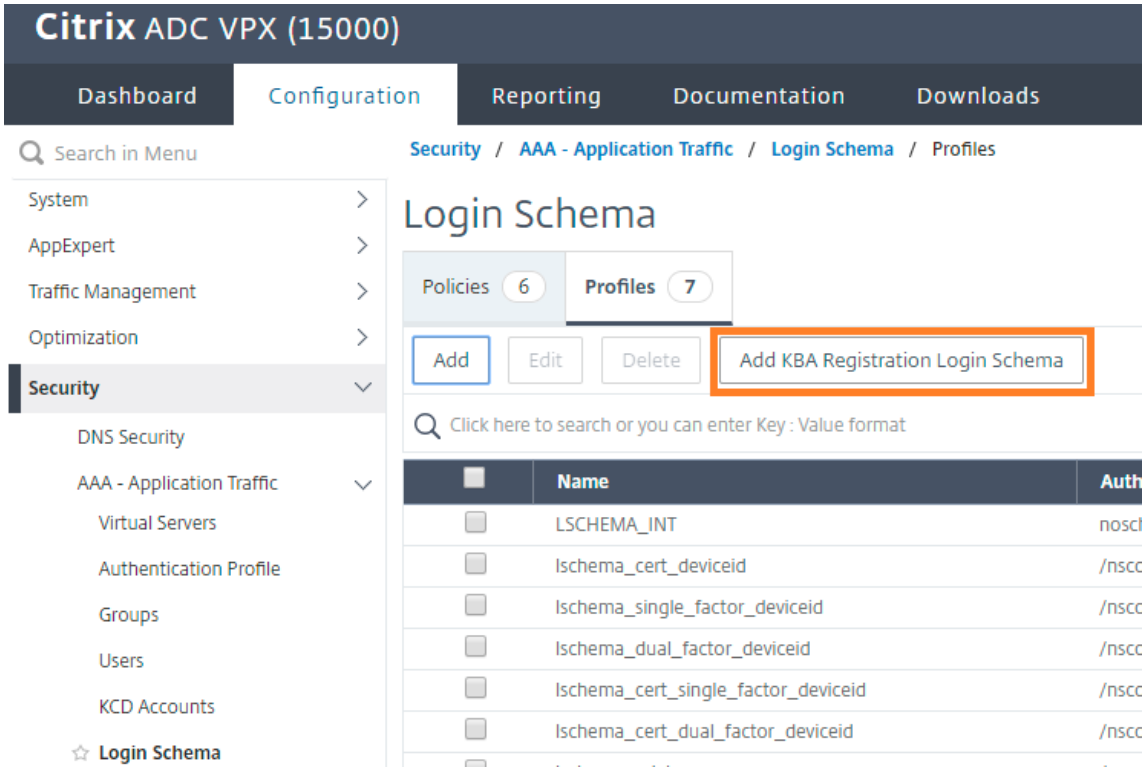
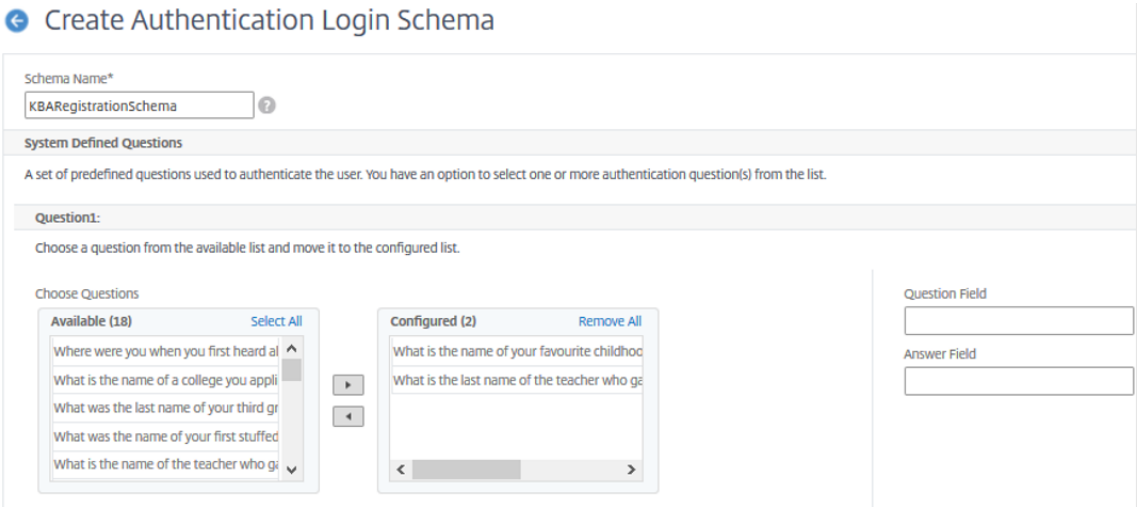
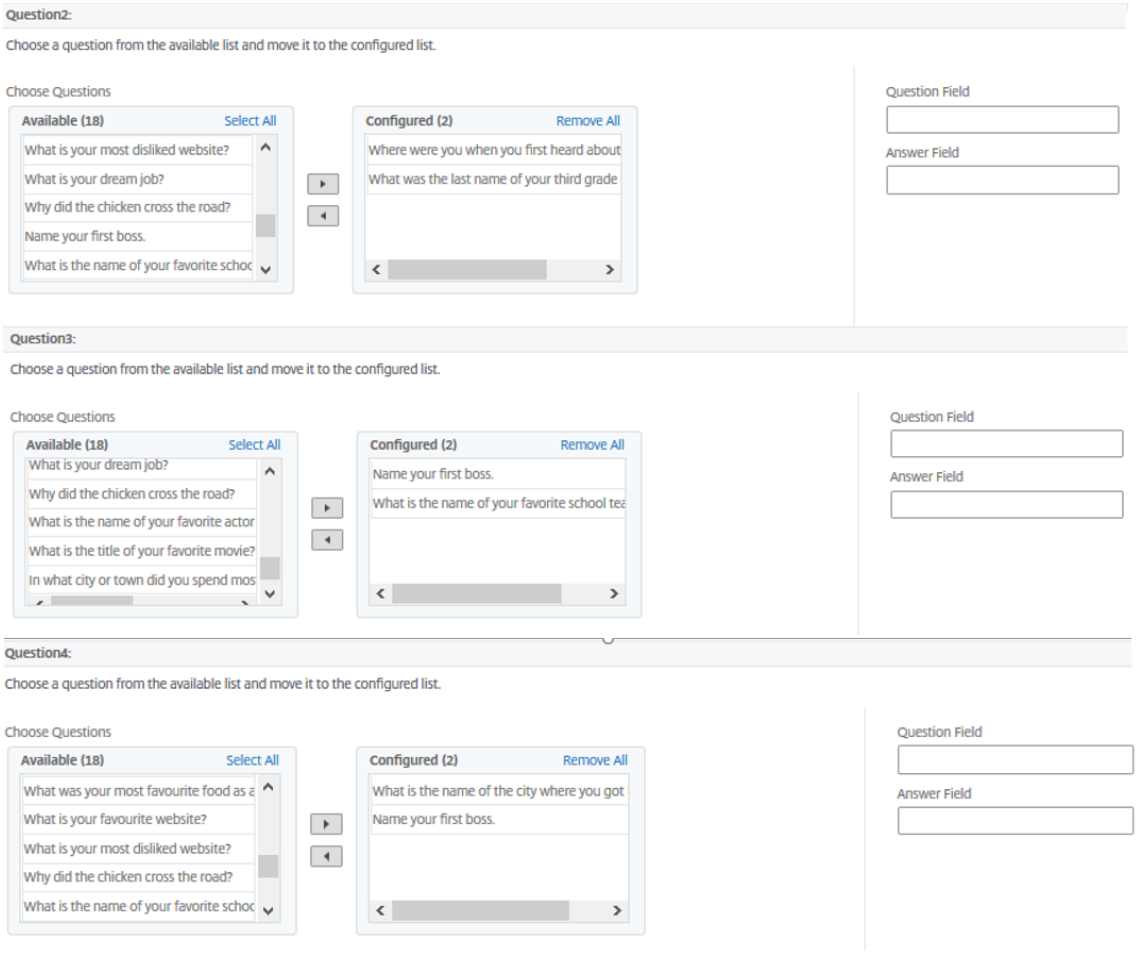
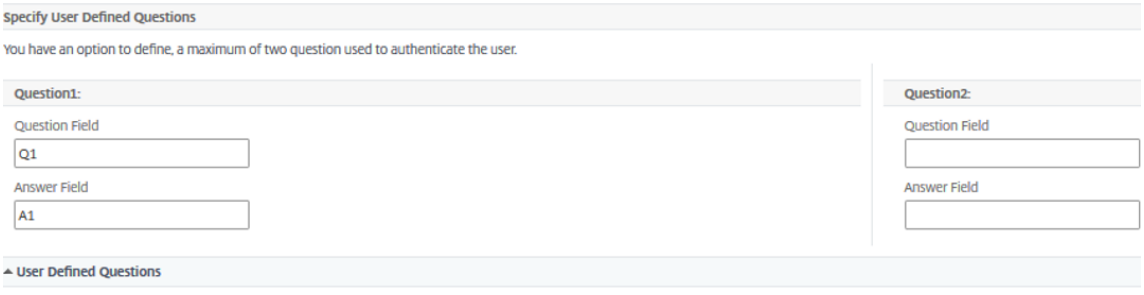
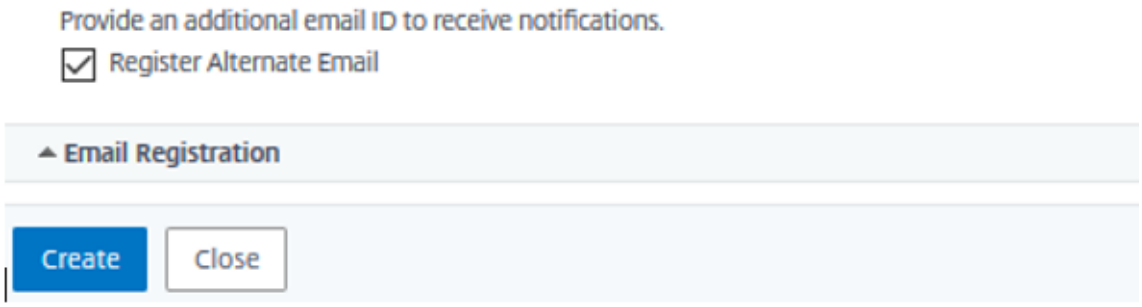
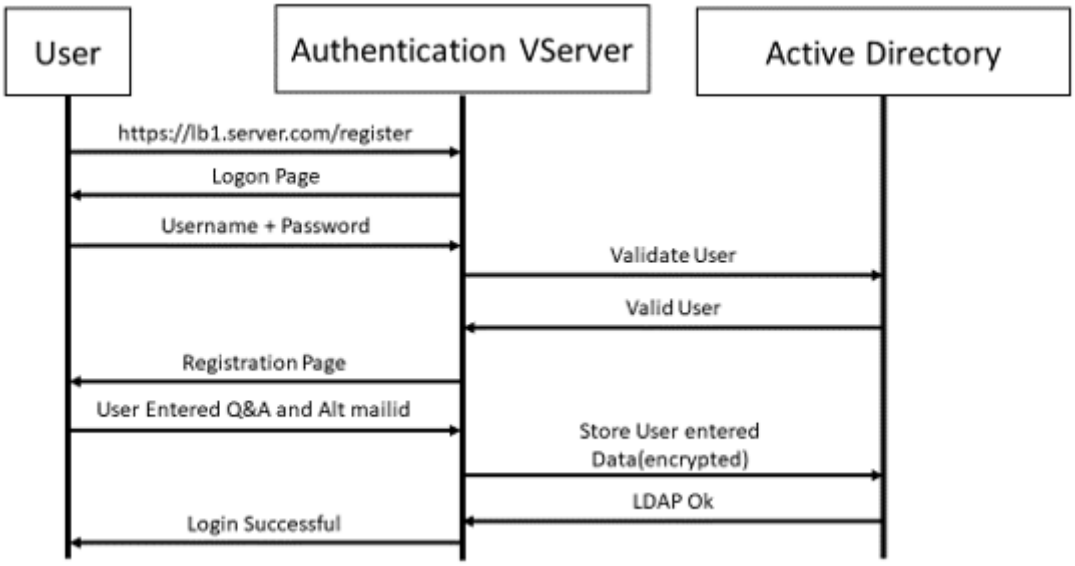
add rewrite action non_cors_header_action insert_http_header X-Frame-
    Options \'\"DENY\"\'

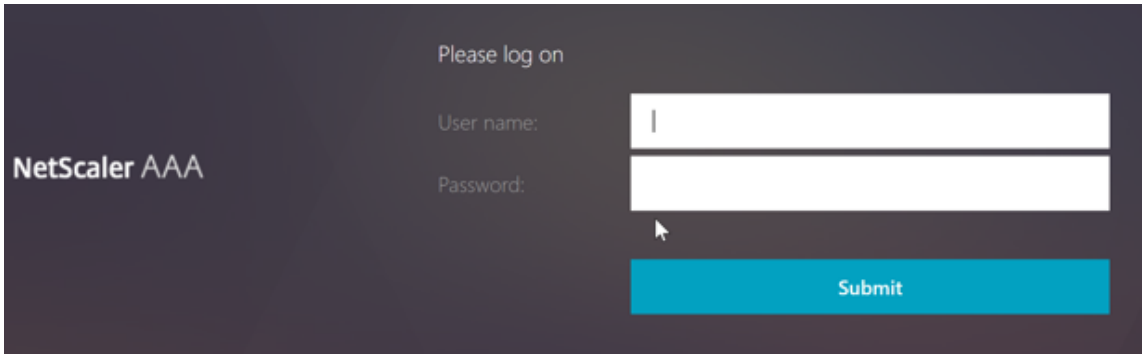
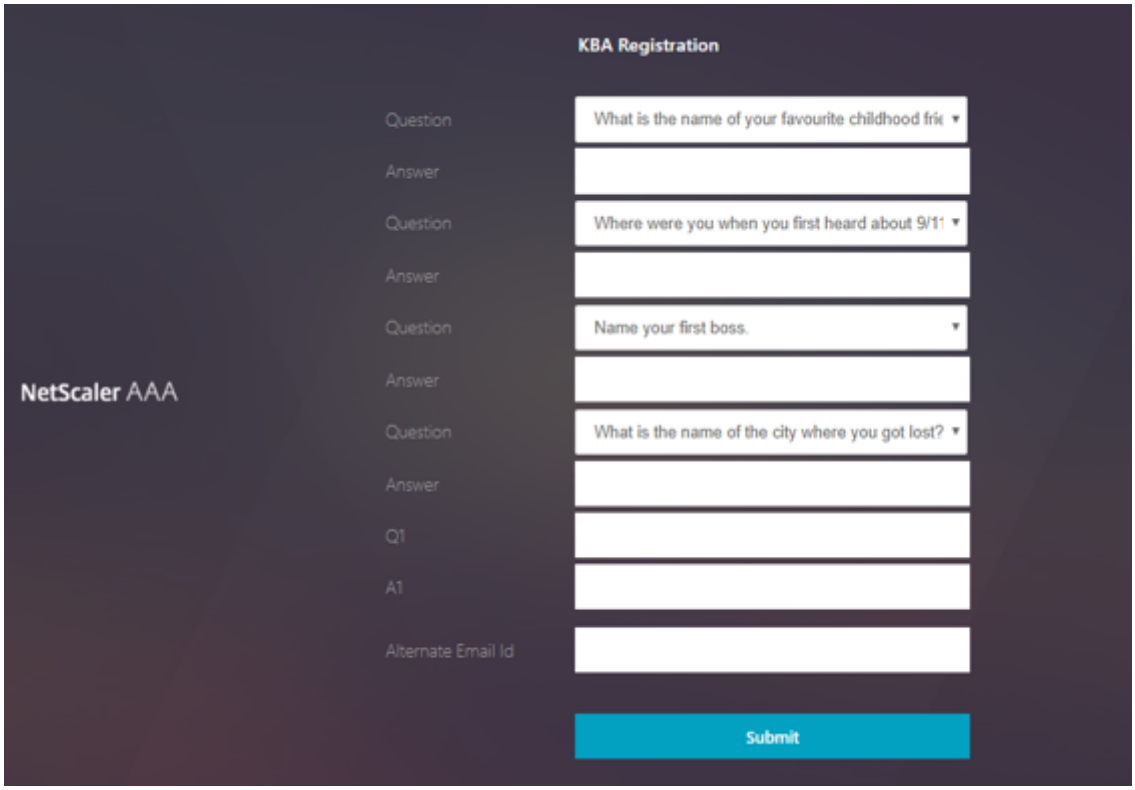
add rewrite policy non_cors_header_pol true non_cors_header_action

bind authentication vserver av_cors -policy cors_header_pol -priority
    100 -type AAA_RESPONSE

bind vpn vserver av_cors -policy cors_header_pol -priority 100 -type
    AAA_RESPONSE
```

Content Security Policy response header support for Citrix Gateway and authentication virtual server generated responses

September 14, 2021

Starting from Citrix ADC release build 13.0-76.29, the Content-Security-Policy (CSP) response header is supported for Citrix Gateway and authentication virtual server generated responses.

The Content-Security-Policy (CSP) response header is a combination of policies which browser uses to avoid Cross Site Scripting (CSS) attacks.
The HTTP CSP response header allows website administrators to control resources the user agent is allowed to load for a given page. With a few exceptions, policies mostly involve specifying server origins and script endpoints. This helps guard against cross-site scripting attacks.
The CSP header is designed to modify the way browsers render pages, and thus to protect from various cross-site injections, including CSS. It is important to set the header value correctly, in a way that does not prevent proper operation of the website. For example, if the header is set to prevent execution of inline JavaScript, the website must not use inline JavaScript in its pages.

The following are the advantages of CSP response header.

- The primary function of a CSP response header is to prevent CSS attacks.
- In addition to restricting the domains from which content can be loaded, the server can specify which protocols are allowed to be used; for example (and ideally, from a security standpoint), a server can specify that all contents must be loaded using HTTPS.

- CSP helps in securing Citrix ADC from cross-site scripting attacks by securing files like “tmindex.html” and “homepage.html. The file “tmindex.html” is related to authentication and file “homepage.html” is related to the published apps/links.

Configuring Content-Security-Policy header for Citrix Gateway and authentication virtual server generated responses

To enable CSP header, you need to configure your web server to return the CSP HTTP header.

> **Points to note**
>
> 1. By default, the CSP header is disabled.
> 2. While enabling/disabling default CSP policy, you are recommended to run the following command. `Flush cache contentgroup loginstaticobjects`
> 3. For modifying the CSP policy for tmindex.html, homepage.html and so on, you are recommended to modify `httpd.conf`. To modify `httpd.conf`, open `httpd.conf` in any xml editor, scroll down to the tag **DirectoryMatch** and locate the following directories, “/netscaler/ns_gui/vpns”, “/netscaler/ns_gui/epa”, and modify “Header set Content-Security-Policy”.

To configure CSP for authentication virtual server and Citrix Gateway generated responses using CLI, type the following command at the command prompt:

```
set aaa parameter -defaultCSPHeader <ENABLE/DISABLE>
```

To configure CSP for Citrix Gateway and authentication virtual server generated responses using GUI.

1. Navigate to **Citrix Gateway > Global Settings**, click **Change authentication AAA settings** under Authentication Settings.



2. On the **Configure AAA Parameters** page, select the **Enabled in Default CSP Header** field.



An example for Content-Security-Policy header customization

The following is an example for CSP header customization to include images and scripts only from the following two specified sources respectively, `https://company.fqdn.com, https://example.com`.

Sample configuration

```
add rewrite action modify_csp insert_http_header Content-Security-
    Policy "\"default-src 'self'; script-src 'self' https://company.fqdn
    .com 'unsafe-inline' 'unsafe-eval'; connect-src 'self'; img-src http
    ://localhost:* https://example.com 'self' data: http: https:; style-
    src 'self' 'unsafe-inline'; font-src 'self'; frame-src 'self'; child
    -src 'self' com.citrix.agmacepa://* citrixng://* com.citrix.
    nsgclient://*; form-action 'self'; object-src 'self'; report-uri /
    nscsp_violation/report_uri\""

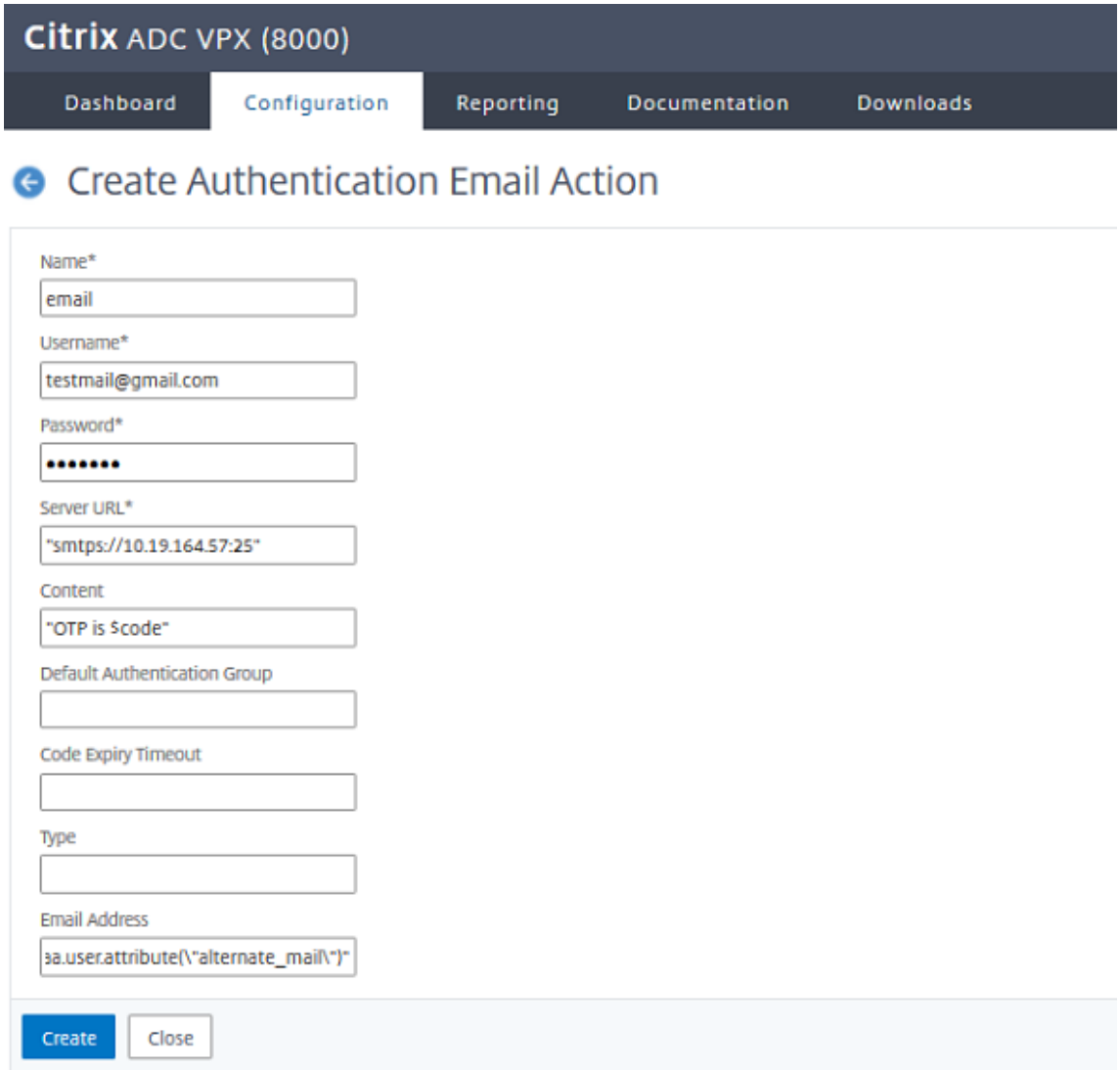
add rewrite policy add_csp true modify_csp

bind authentication vserver auth1 -policy add_csp -priority 1 -
    gotoPriorityExpression NEXT -type AAA_RESPONSE
```

Self-service password reset

September 14, 2021

Self-service password reset is a web-based password management solution. It is available in both authentication, authorization, and auditing feature of Citrix ADC appliance and Citrix Gateway. It eliminates the user's dependency on administrator's assistance for changing password.

The self-service password reset provides end user the ability to securely reset or create a password in the following scenarios:

- User has forgotten the password.
- User is unable to logon.

Until now, if an end user forgets an AD password, the end user had to contact the AD administrator to reset the password. With self-service password reset functionality, an end user can reset the password without an administrator's intervention.

The following are some of the benefits of using self-service password reset:

- Increased productivity through automatic password change mechanism, which eliminates the lead-time for users to wait for password resets.

- With automatic password change mechanism, admins can concentrate on other critical tasks.

The following figure illustrates the self-service password reset flow to reset the password.



To use the self-service password reset, a user must be registered either with the Citrix authentication, authorization, and auditing or with Citrix Gateway virtual server.

Self-service password reset provides the following capabilities:

- **New user self-registration.** You can self-register as a new user.
- **Configure knowledge-based questions.** As an administrator, you can configure a set of questions for users.
- **Alternate email ID registration.** You must provide an alternate email ID while registration. The OTP is sent to the alternate email ID because the user has forgotten the primary email ID password.

  > **Note:**
  >
  > Starting from version 12.1 build 51.xx, alternate email ID registration can be done as standalone. A new Loginschema, **AltEmailRegister.xml** is introduced to do only alternate email ID registration. Previously, alternate email ID registration could be done only while doing the KBA registration.

- **Reset forgotten password.** User can reset the password by answering the knowledge-based questions. As an administrator, you can configure and store the questions.

The self-service password reset provides the following two new authentication mechanisms:

- **Knowledge based question and answer.** You must register to Citrix authentication, authorization, and auditing or to a Citrix Gateway before selecting the knowledge-based question and

answer schema.

- **Email OTP authentication.** An OTP is sent to the alternate email ID, which user has registered during self-service password reset registration.

> **Note**
>
> These authentication mechanisms can be used for the self-service password reset use cases, and for any authentication purposes similar to any of the existing authentication mechanisms.

Prerequisites

Before you configure the self-service password reset, review the following prerequisites:

- Citrix ADC feature release 12.1, build 50.28.
- Supported version is 2016, 2012, and 2008 AD domain function level.
- The ldapBind username bound to the Citrix ADC needs to have write access to the users AD path.

> **Note**
>
> Self-service password reset is supported in nFactor authentication flow only. For more information, see nFactor Authentication through Citrix ADC.

Limitations

Following are some of the limitations of self-service password reset:

- Self-service password reset is available only if authentication back-end is LDAP.
- User cannot see the already registered alternate email ID.
- Knowledge-based question and answer, and email OTP authentication and registration cannot be the first factor in the authentication flow.
- For Native Plug-in and Receiver, registration is supported only through browser.
- The minimum certificate size used for self-service password reset is 1024 bytes, and must follow x.509 standard.

Active directory setting

The Citrix ADC knowledge-based question and answer, and email OTP uses AD attribute to store users data. You must configure an AD attribute to store the questions and answers along with the alternate email ID. The Citrix ADC appliance stores it in the configured KB attribute in the AD user object. When configuring an AD attribute, consider the following:

- The attribute length must be at least 128 characters.
- The AD attribute must support 32k value of maximum length.

- Attribute type must be a 'DirectoryString'.
- A single AD attribute can be used for knowledge-based question and answer and alternate email ID.
- A single AD Attribute cannot be used for Native OTP and knowledge-based question and answer or alternative email ID registration.
- Citrix ADC LDAP administrator must have write access to the selected AD attribute.

You can also use an existing AD attribute. However, make sure that the attribute you plan to use is not used for other cases. For example, userParameters is an existing attribute within the AD user that you could use. To verify this attribute, perform the following steps:

1. Navigate to **ADSI > select user**.
2. Right-click and scroll down to attribute list.
3. On the **CN=testuser Properties** window pane, you can see the **userParameters** attribute is not set.



Self-service password reset registration

To implement self-service password reset solution on a Citrix ADC appliance, you have to perform the following:

- Self-service password reset (knowledge-based question and answer/email ID) registration.
- User Logon Page (for password reset, which includes knowledge-based question and answer and email OTP validation and final password reset factor).

A set of predefined questions catalog is provided as a JSON file. As an administrator, you can select

the questions and create self-service password reset registration login schema through Citrix ADC GUI. You can choose any of the following options:

- Select a maximum of four system-defined questions.
- Provide an option for users to customize two questions and answers.

To view the default knowledge-based questions JSON file from CLI

```
root@ns# cd /nsconfig/loginschema/LoginSchema/
root@ns# cat KBQuestions.json
[
    {"question":"What is the last name of the teacher who gave you your first failing
grade?"},
    {"question":"What is the name of your favourite childhood friend?"},
    {"question":"Where were you when you first heard about 9/11?"},
    {"question":"What is the name of a college you applied to but didn't attend?"},
    {"question":"What was the last name of your third grade teacher?"},
    {"question":"What was the name of your first stuffed animal?"},
    {"question":"What is the name of the teacher who gave you your first A?"},
    {"question":"What is the name of the city where you got lost?"},
    {"question":"In what city or town did your mother and father meet?"},
    {"question":"What was your most hated food as a child?"},
    {"question":"What was your most favourite food as a child?"},
    {"question":"What is your favourite website?"},
    {"question":"What is your most disliked website?"},
    {"question":"What is your dream job?"},
    {"question":"Why did the chicken cross the road?"},
    {"question":"Name your first boss."},
    {"question":"What is the name of your favorite school teacher?"},
    {"question":"What is the name of your favorite actor or actress?"},
    {"question":"What is the title of your favorite movie?"},
    {"question":"In what city or town did you spend most of your youth?"}
]
```

Note

- Citrix Gateway includes the set of system-defined questions by default. Administrator can edit the “KBQuestions.json” file to include their choice of questions.
- System-defined questions are displayed only in English and language localisation support is not available for these questions.

To complete knowledge-based question and answer registration Login Schema using GUI

1. Navigate to **Security > AAA – Application Traffic > Login Schema**.



2. On the **Login Schema** page, click **Profiles**.
3. Click **Add KBA Registration Login Schema**.
4. On the **Create Authentication Login Schema** page, specify a name in **Schema Name** field.





5. Select the questions of your choice and move it to the **Configured** list.
6. In the **User Defined Questions** section, you can provide questions and answers in the Q1 and A1 fields.



7. In the **Email Registration** section, check the **Register Alternate Email** option. You can register the **Alternate Email ID** from user registration logon page to receive the OTP.



8. Click **Create**. The login schema once generated displays all the configured questions to the end user during registration process.

Create user registration and management workflow using CLI

The following are required before you begin the configuration:

- IP address assigned to the authentication virtual server
- FQDN corresponding to the assigned IP address
- Server certificate for authentication virtual server

To setup device registration and management page, you require an authentication virtual server. The following figure illustrates the user registration.



To create authentication virtual server

1. Configure an authentication virtual server. It must be of type SSL and make sure to bind authentication virtual server with portal theme.


```
> add authentication vserver <vServerName> SSL <ipaddress> <port>
> bind authentication vserver <vServerName> [-portaltheme<string>]
```

2. Bind SSL virtual server certificate-key pair.

```
> bind ssl vserver <vServerName> certkeyName <string>
```

Example:

```
> add authentication vserver authvs SSL 1.2.3.4 443
> bind authentication vserver authvs -portaltheme RFWebUI
> bind ssl vserver authvs -certkeyname c1
```

To create LDAP logon action

```
    > add authentication ldapAction <name> {
-serverIP <ipaddr|ipv6_addr|> [-serverPort <port>] [-ldapBase <BASE> ]
     [-ldapBindDn <AD USER>] [-ldapBindDnPassword <PASSWORD>] [-
ldapLoginName <USER FORMAT>]
```

> **Note**
>
> You can configure any authentication policy as the first factor.

Example:

```
    > add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4
      -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
    ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
    PASSWORD -ldapLoginName samAccountName -serverport 636 -sectype
    SSL -KBAttribute userParameters
```

To create authentication policy for LDAP logon

```
    > add authentication policy <name> <rule> [<reqAction]
```

Example:

```
    > add authentication policy ldap_logon -rule true -action
      ldap_logon_action
```


To create knowledge-based question and answer registration action

Two new parameters are introduced in ldapaction. “KBAttribute” for KBA Authentication (Registration and validation) and “alternateEmailAttr” for registration of user’s alternate email ID.

```
> add authentication ldapAction <name> {
-serverIP  <ipaddr|ipv6_addr|> [-serverPort <port>] [-ldapBase <BASE>
] [-ldapBindDn <AD USER>] [-ldapBindDnPassword <PASSWORD>] [-
ldapLoginName <USER FORMAT>] [-KBAttribute <LDAP ATTRIBUTE>] [-
alternateEmailAttr <LDAP ATTRIBUTE>]
```

Example:

```
> add authentication ldapAction ldap1 -serverIP 1.2.3.4 -sectype
   ssl -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
   ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
   PASSWORD -ldapLoginName samAccountName -KBAttribute
   userParameters -alternateEmailAttr userParameters
```

Display user registration and management screen

The “KBARegistrationSchema.xml” login schema is used to display the user registration page to the end user. Use the following CLI to display the login schema.

```
> add authentication loginSchema <name> -authenticationSchema <string>
```

Example:

```
> add authentication loginSchema kba_register -authenticationSchema /
   nsconfig/loginschema/LoginSchema/KBARegistrationSchema.xml
```

Citrix recommends two ways of displaying the user registration and management screen: URL or LDAP Attribute.

Using URL

If the URL path contains ‘/register’ (for example, https://lb1.server.com/register) then the user registration page is displayed using URL.

To create and bind registration policy

```
> add authentication policylabel user_registration -loginSchema
   kba_register
```



```
> add authentication policy ldap1 -rule true -action ldap1
> bind authentication policylabel user_registration -policy ldap1 -
    priority 1
```

To bind authentication policy to authentication, authorization, and auditing virtual server when the URL contains '/register'

```
> add authentication policy ldap_logon -rule "http.req.cookie.value(\"
    NSC_TASS\").contains(\"register\")" -action ldap_logon
> bind authentication vserver authvs -policy ldap_logon -nextfactor
    user_registration -priority 1
```

To bind certificate to VPN global

```
bind vpn global -userDataEncryptionKey c1
```

> **Note**
>
> You must bind the certificate to encrypt the user data (KB Q&A and registered alternate email ID) stored in AD attribute.

Using attribute

You can bind authentication policy to the authentication, authorization, and auditing virtual server to check if the user is already registered or not. In this flow, any of the preceding policies before knowledge-based question and answer registration factor needs to be LDAP with KBA attribute configured. This is to check if the AD user is registered or not using an AD attribute.

> **Important**
>
> The rule "AAA.USER.ATTRIBUTE("kba_registered").EQ("0")" enforces new users to register for knowledge-based questions and answer and alternate email.

To create authentication policy to check if the user is not already registered

```
> add authentication policy switch_to_kba_register -rule "AAA.USER.
    ATTRIBUTE(\"kba_registered\").EQ(\"0\")" -action NO_AUTHN
> add authentication policy first_time_login_forced_kba_registration -
    rule true -action ldap1
```


To create registration policy label and bind to the LDAP registration policy

```
> add authentication policylabel auth_or_switch_register -loginSchema
    LSCHEMA_INT
> add authentication policylabel kba_registration -loginSchema
    kba_register

> bind authentication policylabel auth_or_switch_register -policy
    switch_to_kba_register -priority 1 -nextFactor kba_registration
> bind authentication policylabel kba_registration -policy
    first_time_login_forced_kba_registration -priority 1
```

To bind authentication policy to authentication, authorization, and auditing virtual server

```
bind authentication vserver authvs -policy ldap_logon -nextfactor
    auth_or_switch_register -priority 2
```

User registration and management validation

Once you have configured all the steps mentioned in the previous sections, you should see the UI screenshots shown below.

1. Enter the lb vserver URL; for example, https://lb1.server.com. The logon screen is displayed.



2. Enter the user name and password. Click **Submit**. The **User Registration** screen is displayed.



3. Select the preferred question from the dropdown list and enter the **Answer**.
4. Click **Submit**. The user registration successful screen is displayed.

Configure user logon page

In this example, administrator assumes that the first factor is LDAP logon (for which the end user has forgotten the password). The user then follows the knowledge-based question and answer registration and email ID OTP validation, and finally resets the password using self-service password reset.

You can use any of the authentication mechanisms for self-service password reset. Citrix recommends having either a knowledge-based question and answer, and email OTP or both to achieve strong privacy, and to avoid any illegitimate user password resets.

The following are required before you start configuring the user logon page:

- IP for load balancer virtual server
- Corresponding FQDN for the load balancer virtual server
- Server certificate for the load balancer

Create load balancer virtual server by using CLI

To access the internal website, you have to create an LB virtual server to front the back-end service and delegate the authentication logic to authentication virtual server.


```
> add lb vserver lb1 SSL 1.2.3.162 443 -persistenceType NONE -
    cltTimeout 180 -AuthenticationHost otpauth.server.com -
    Authentication ON -authnVsName authvs

> bind ssl vserver lb1 -certkeyname c1
```

To represent the backend service in load balancing:

```
> add service iis_backendsso_server_com 1.2.3.4 HTTP 80

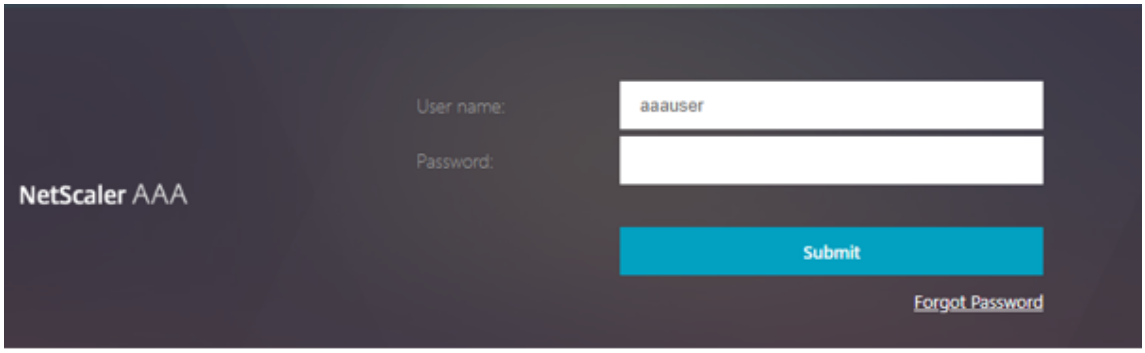
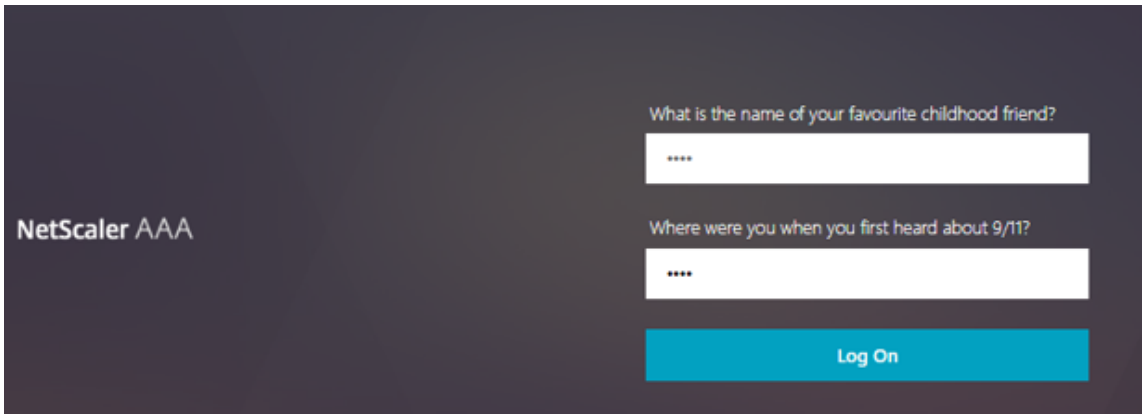
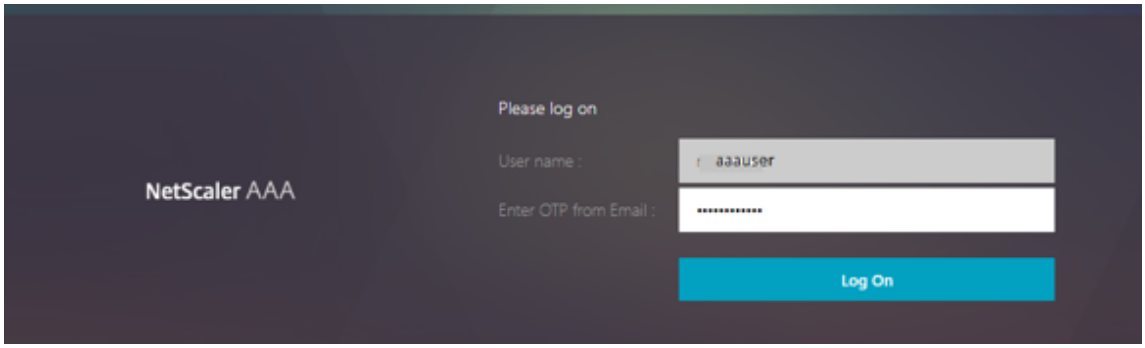
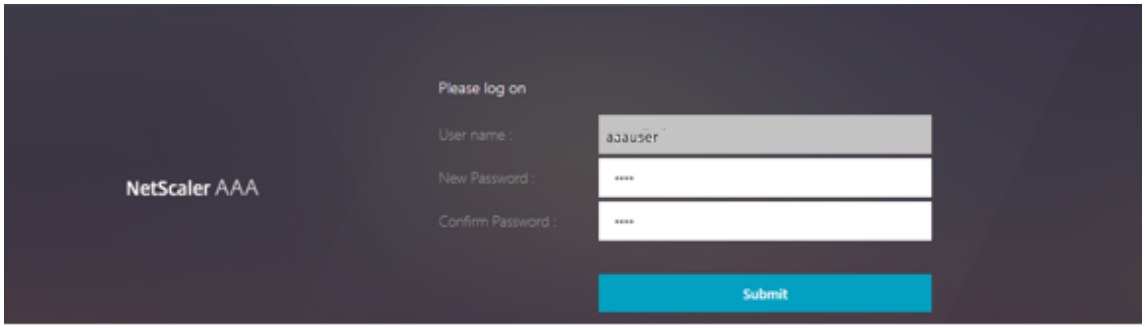
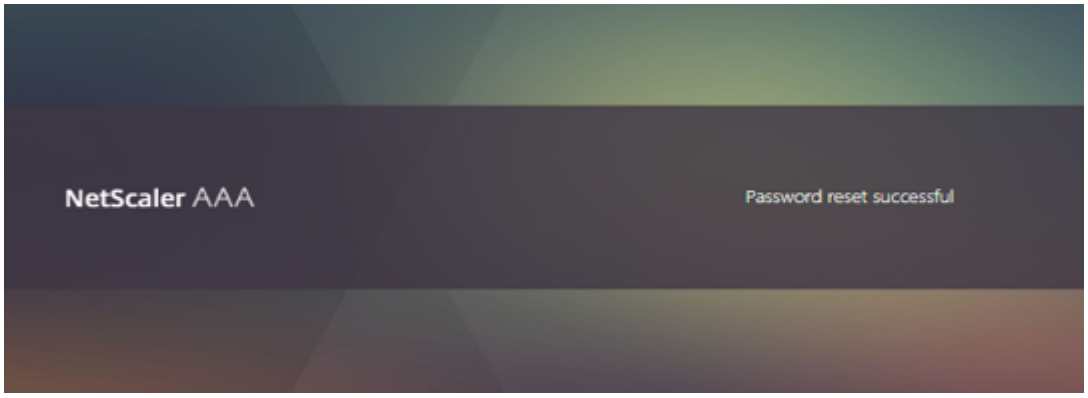
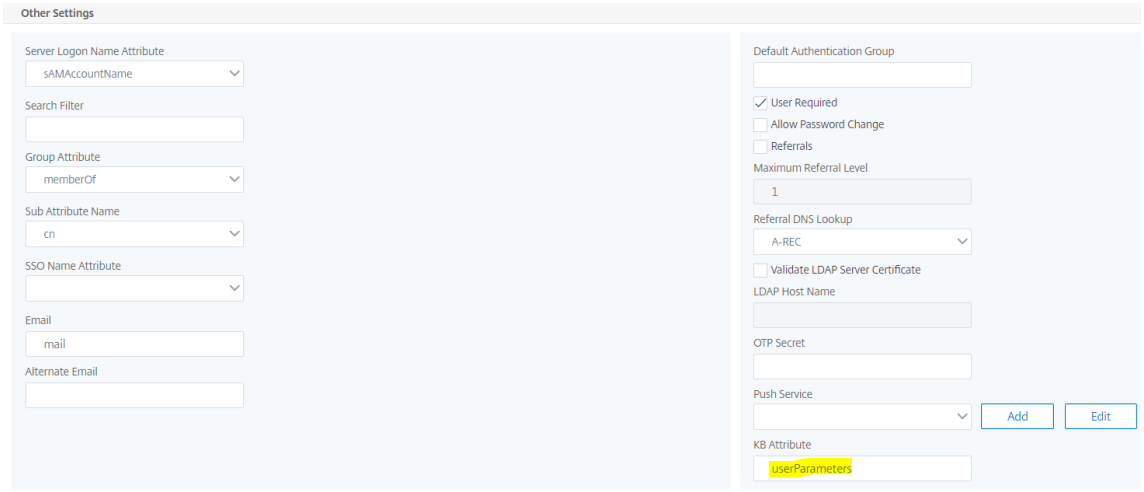
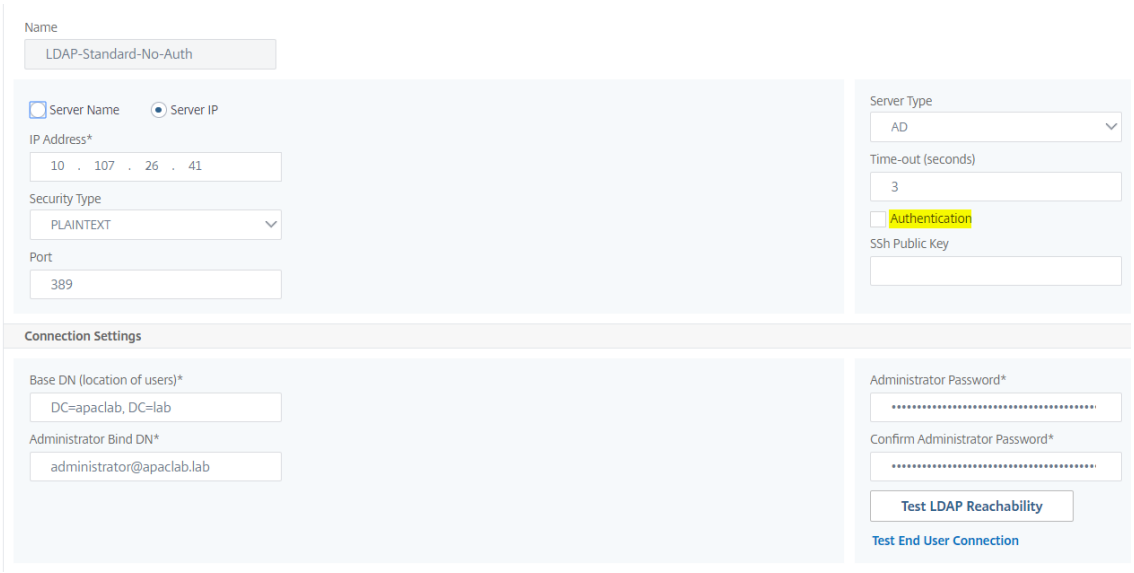
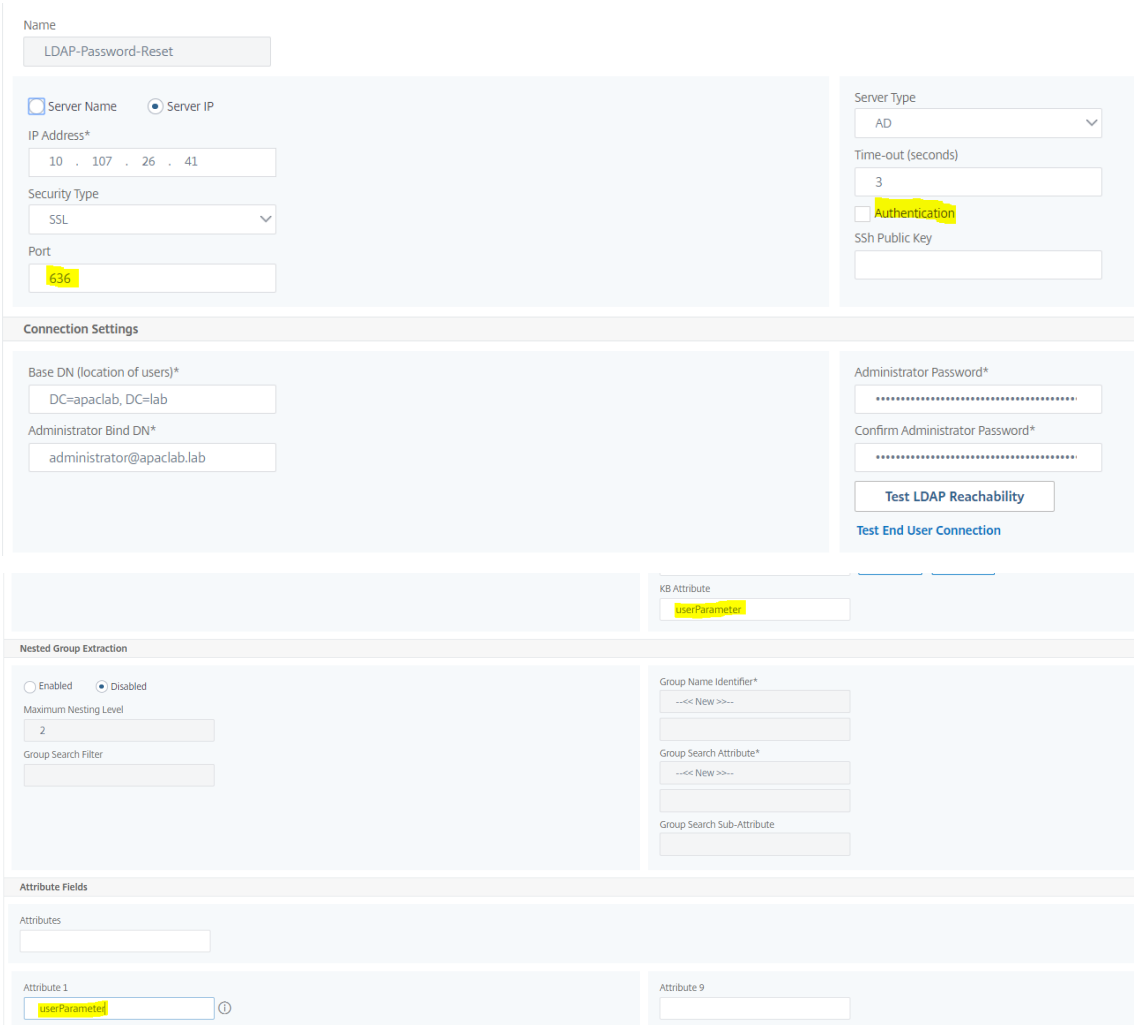
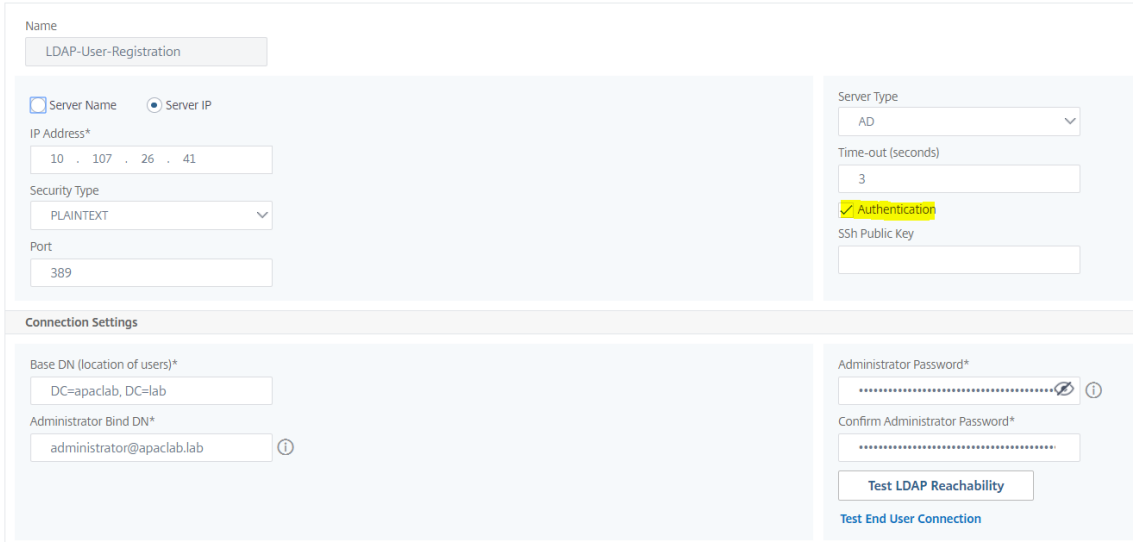
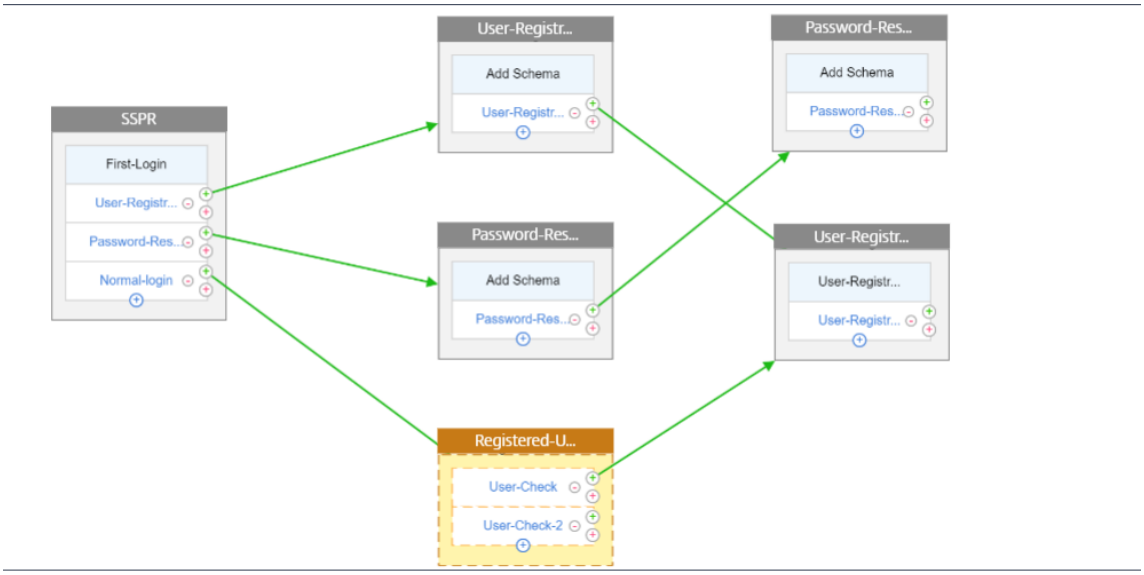
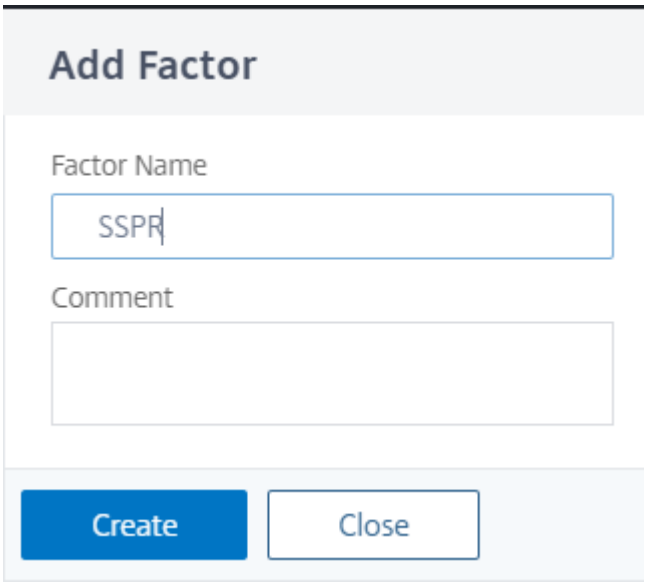
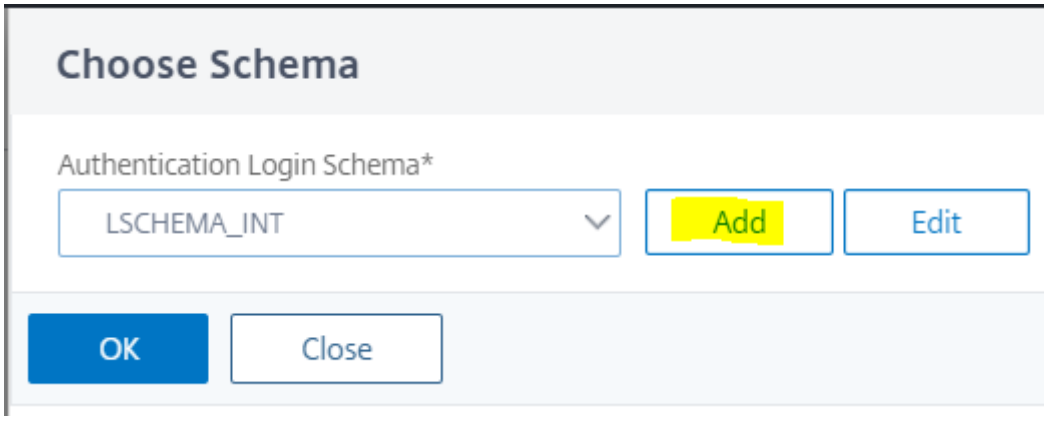
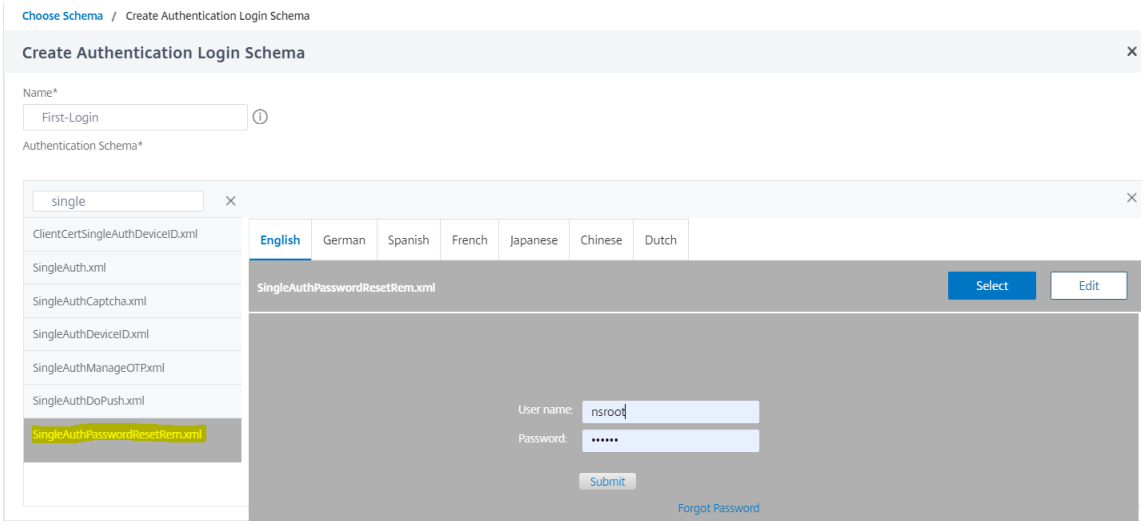
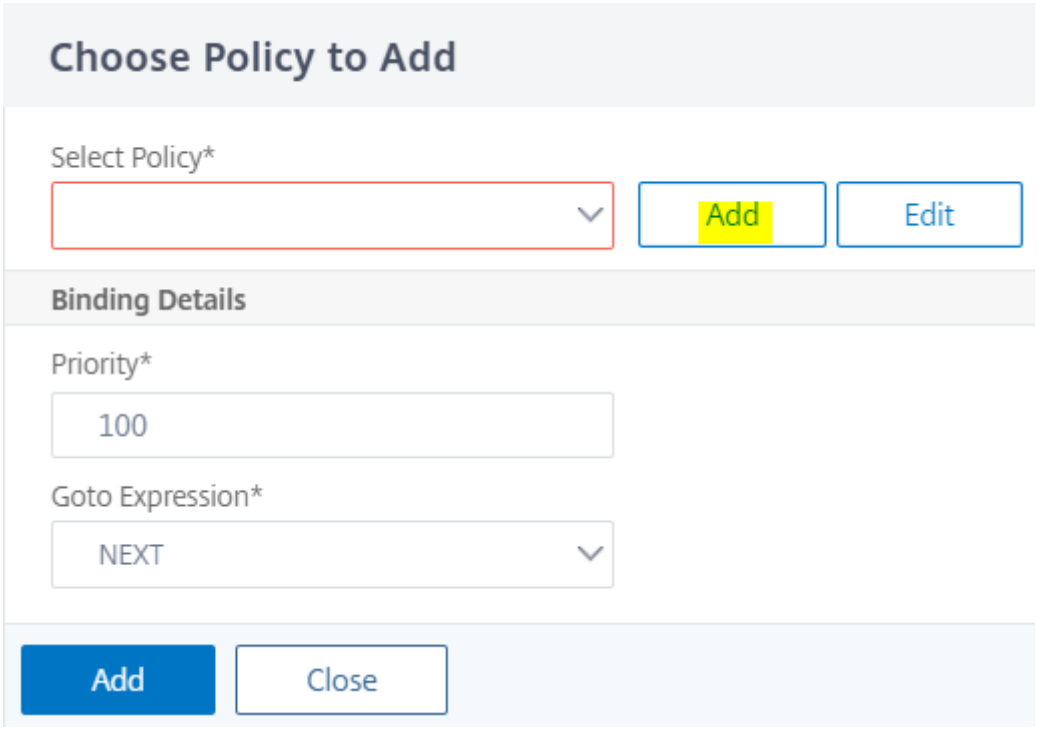
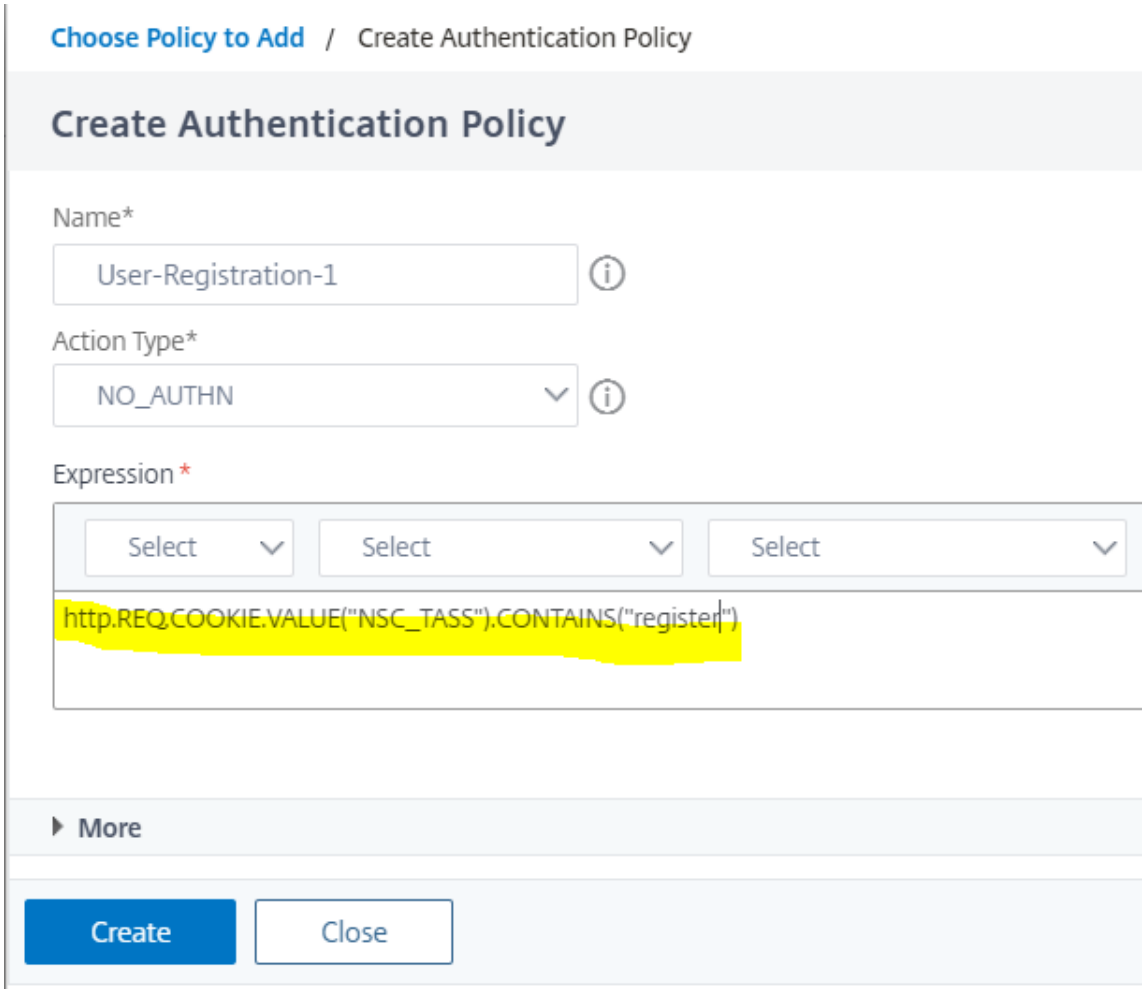
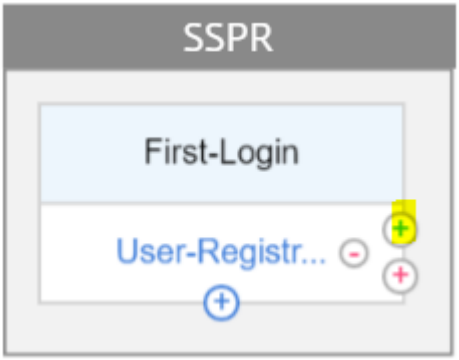
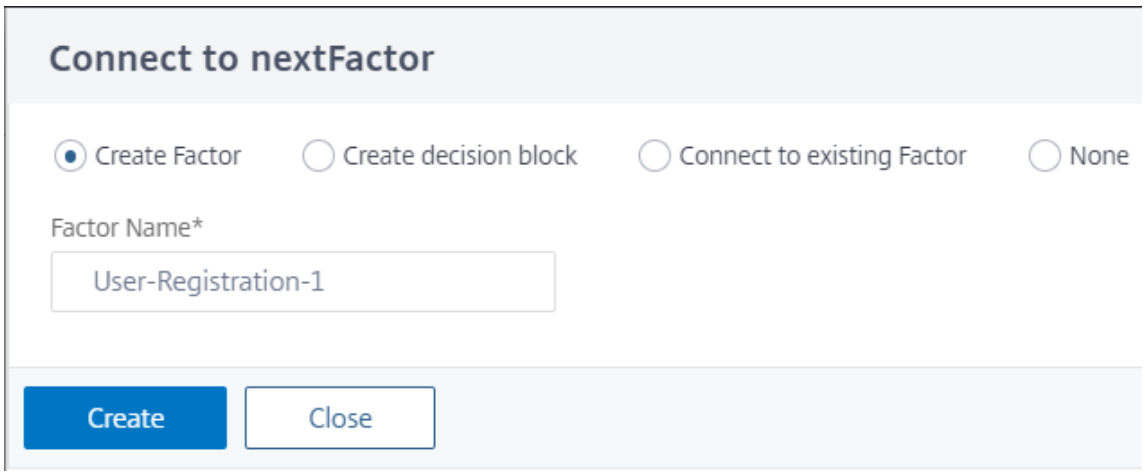
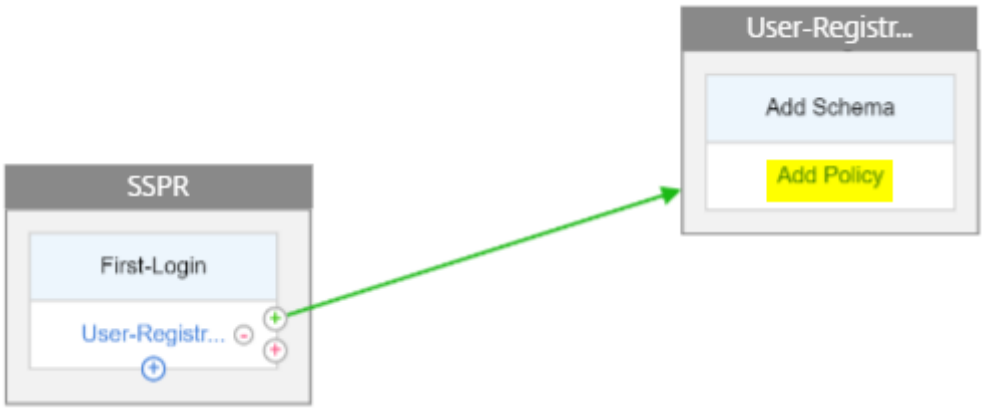
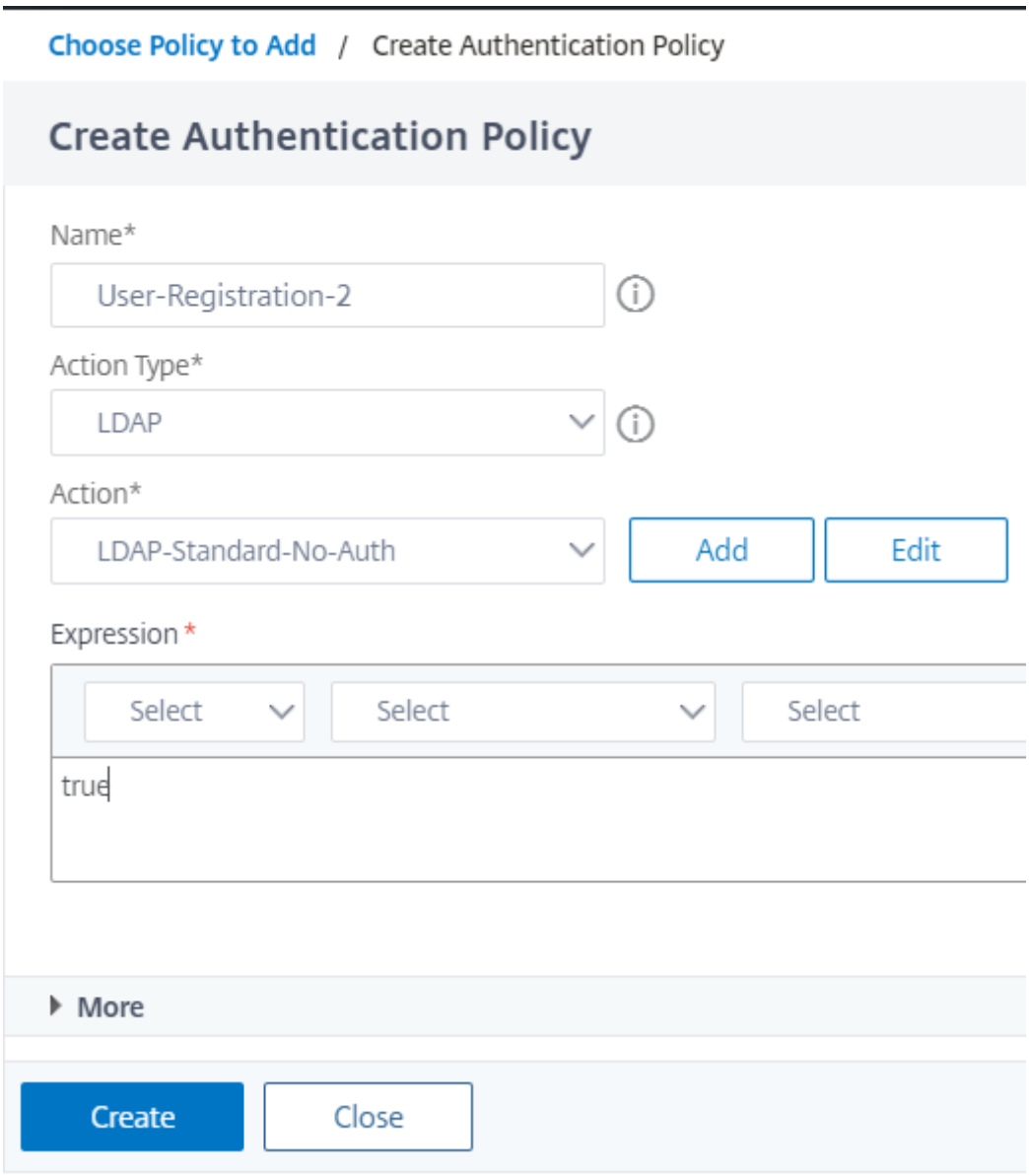
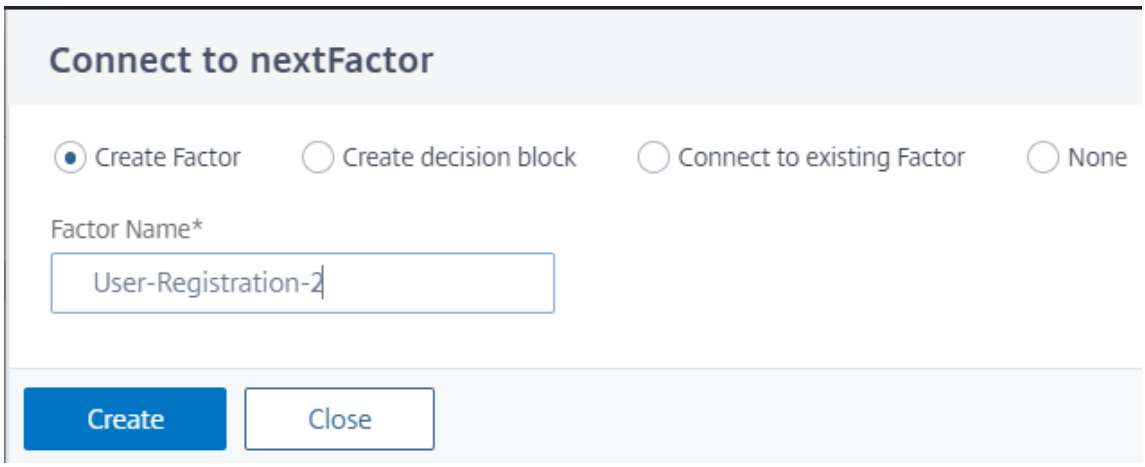
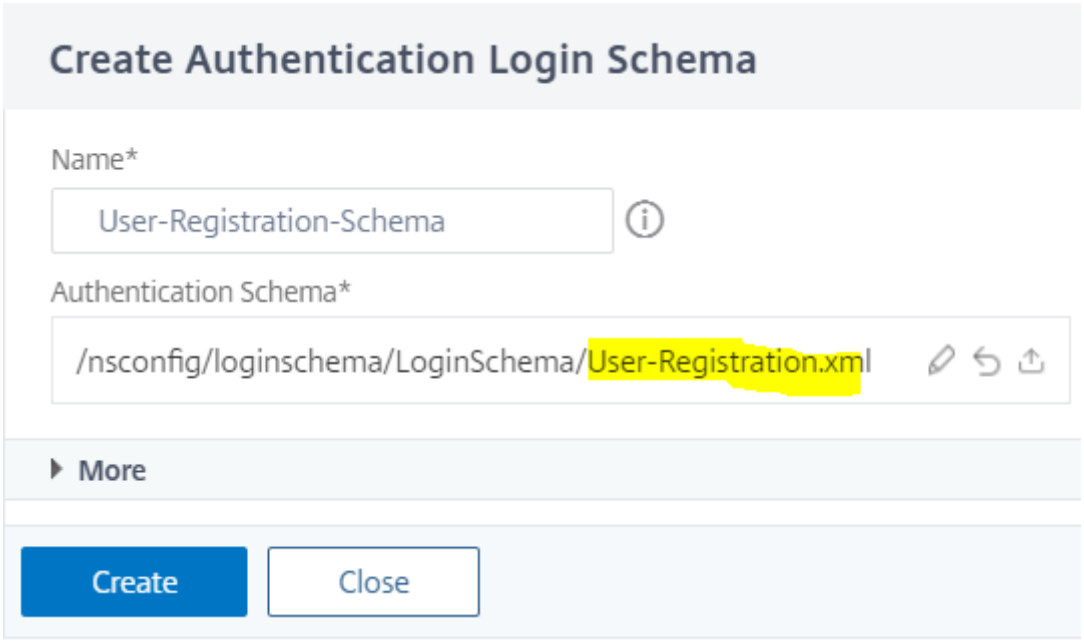
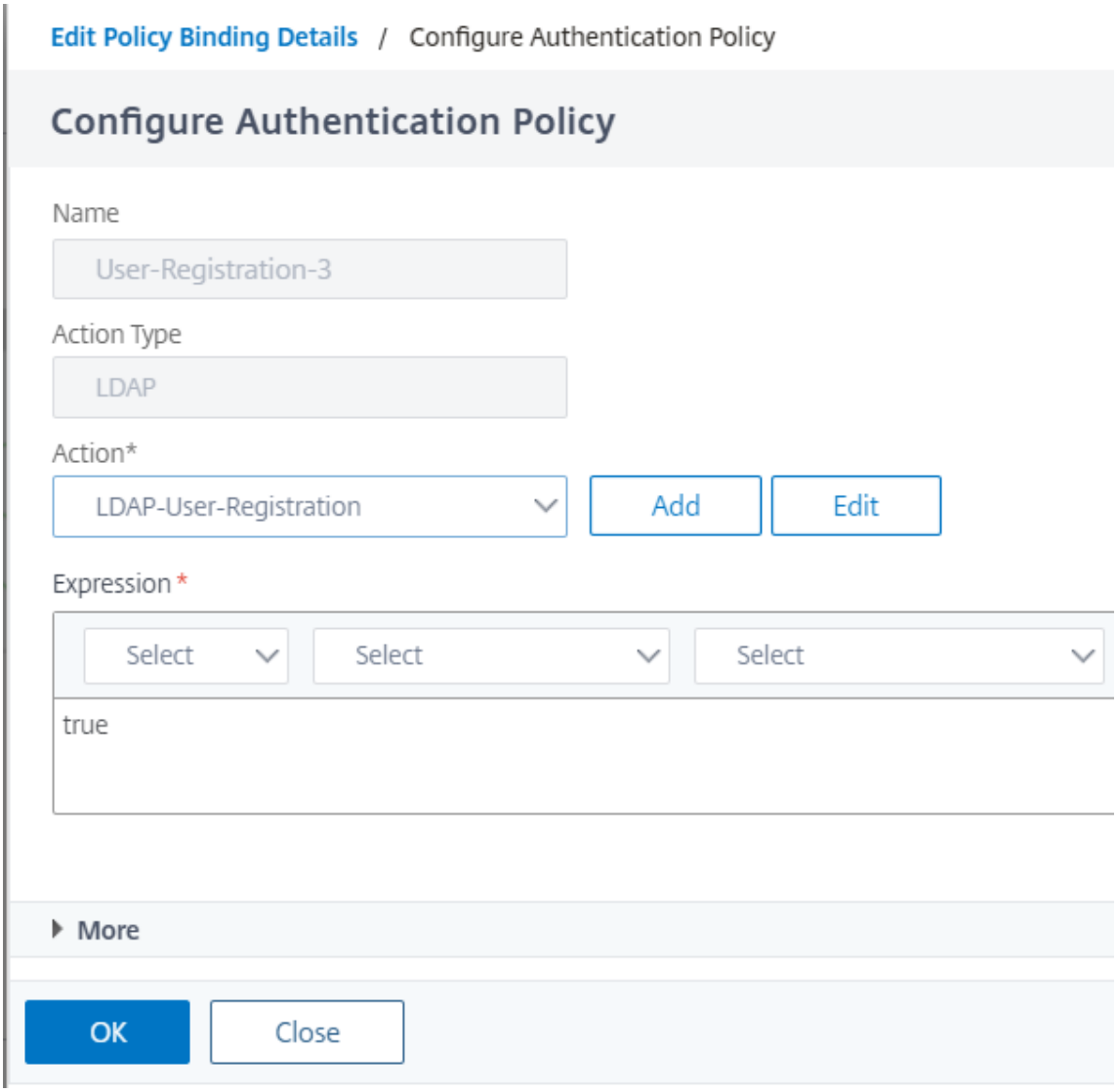
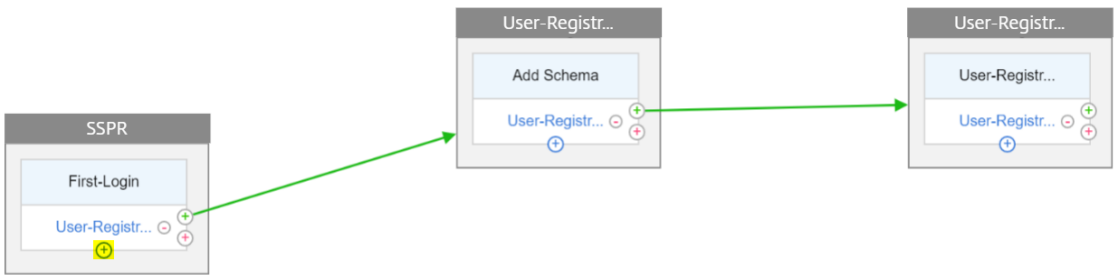
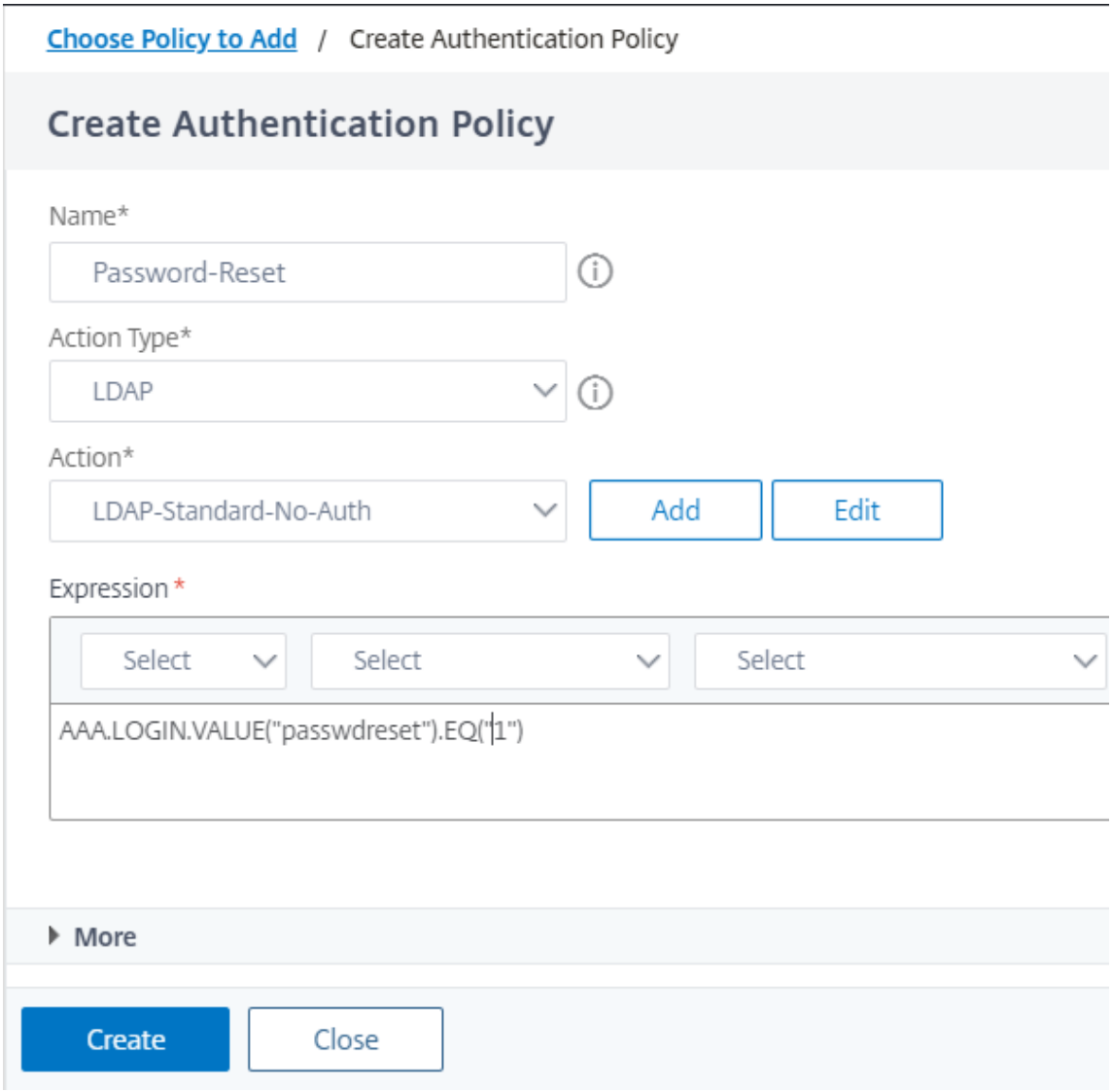
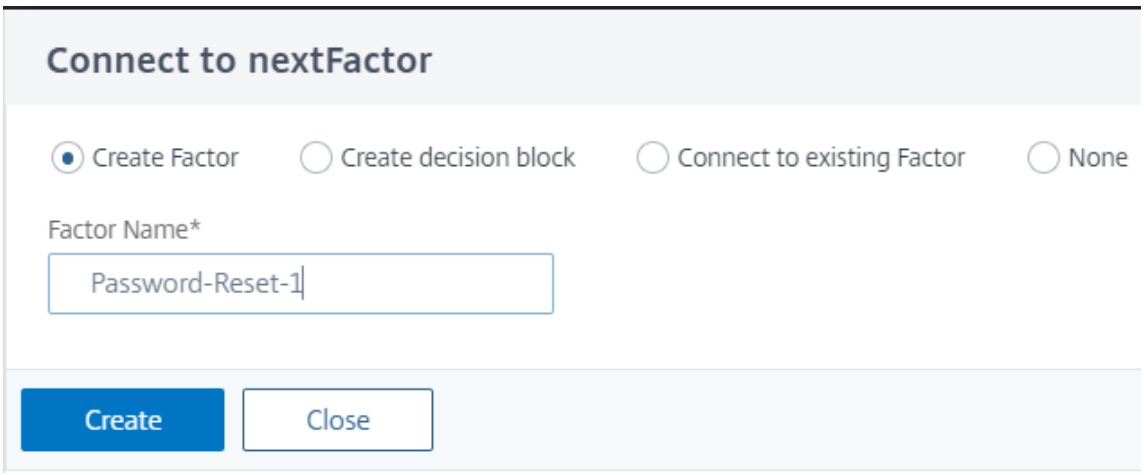
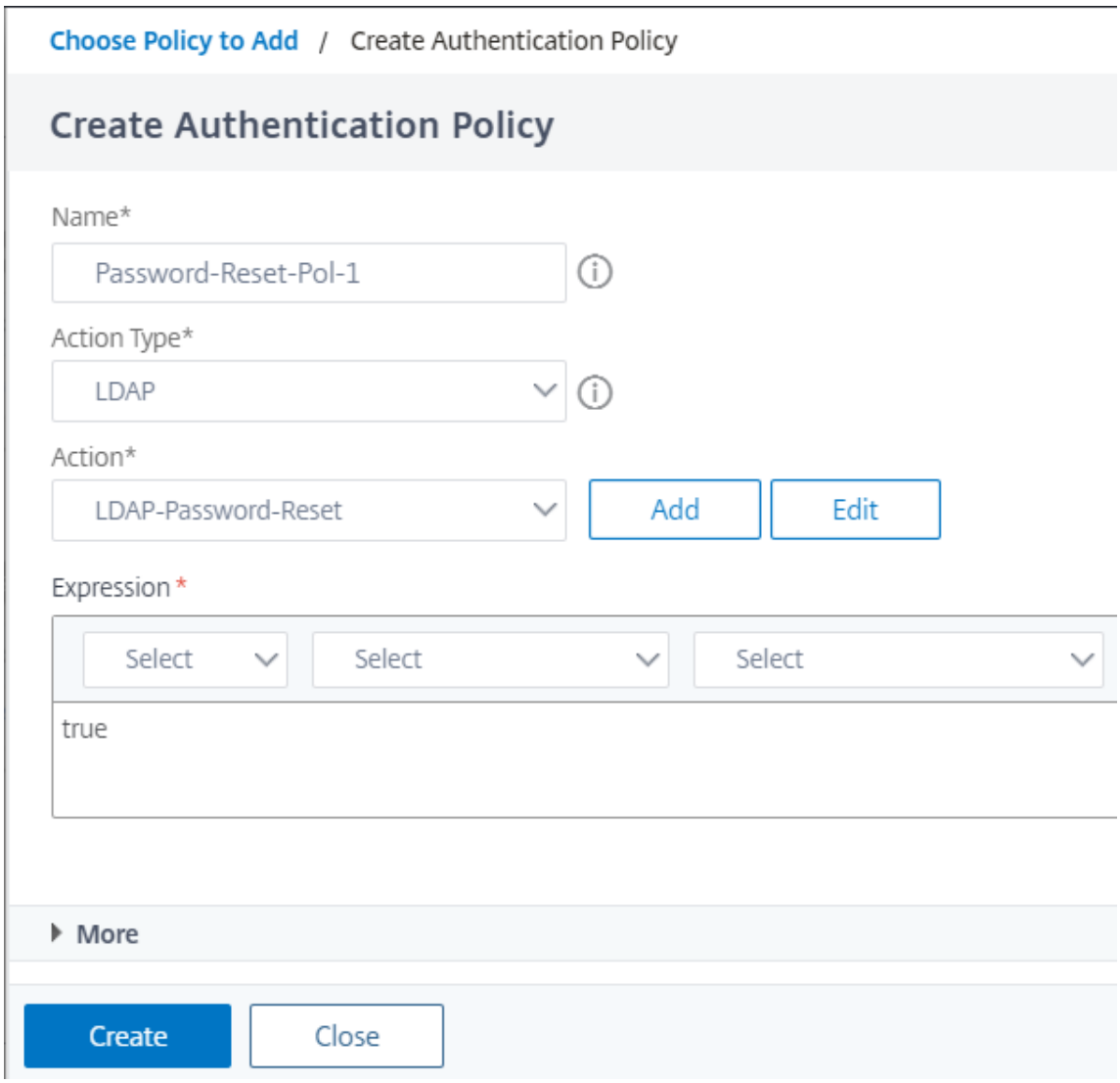
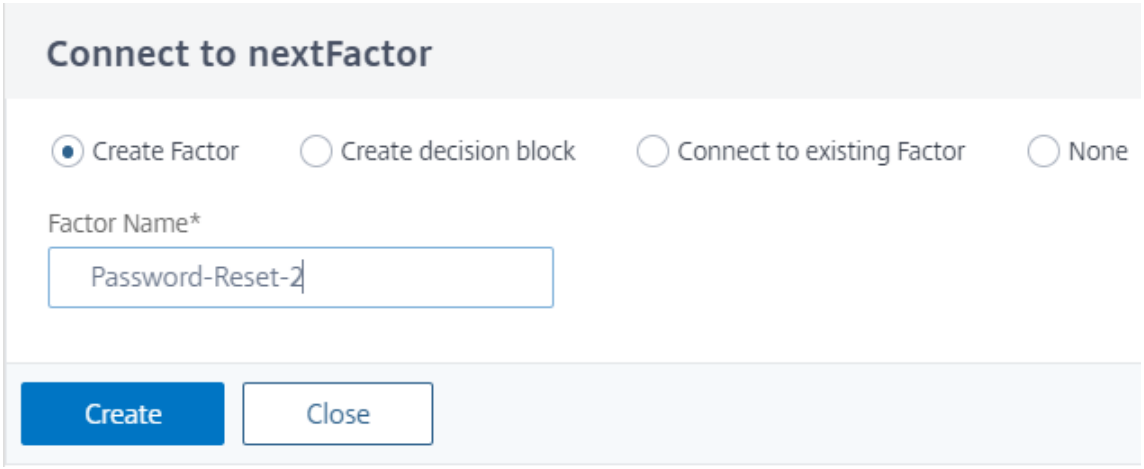
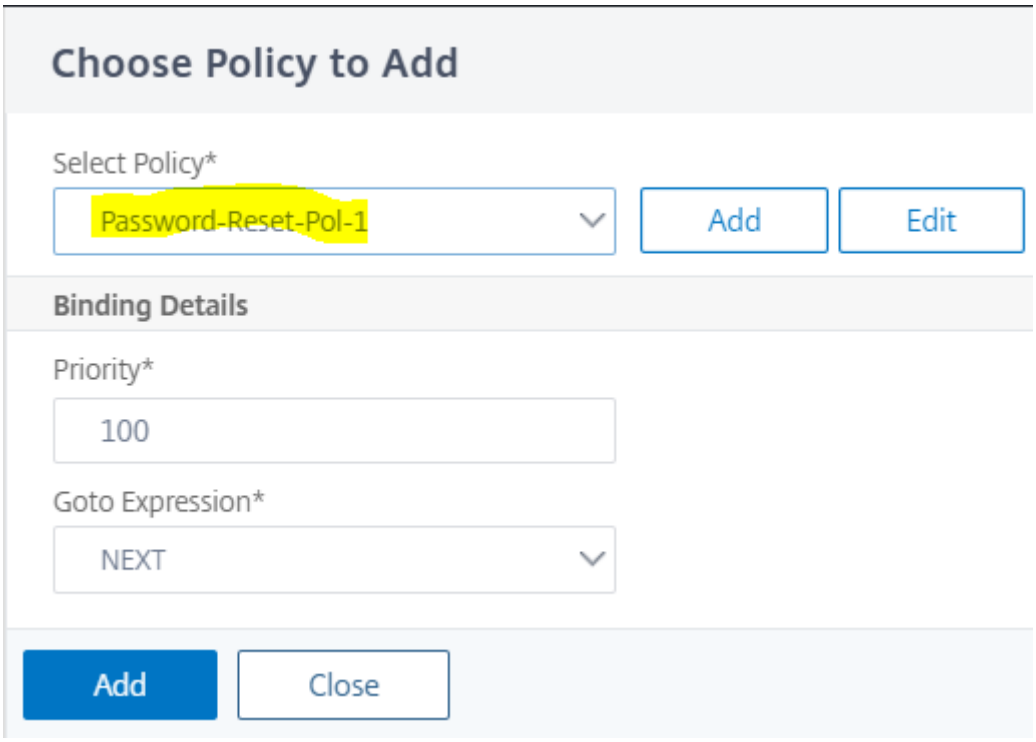
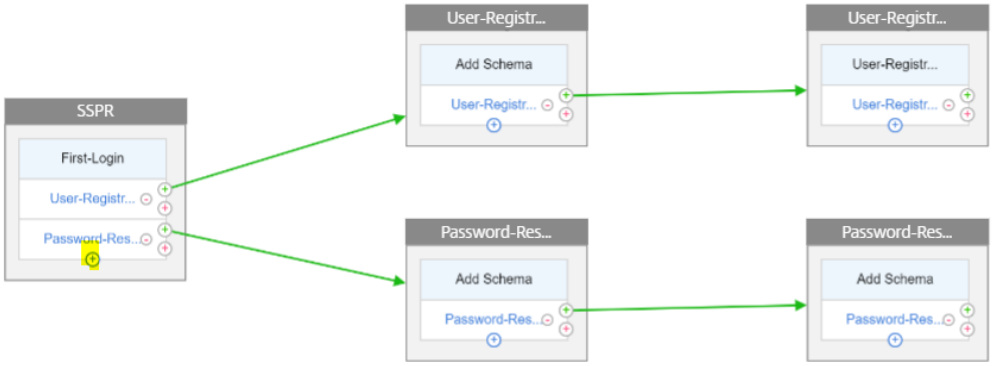
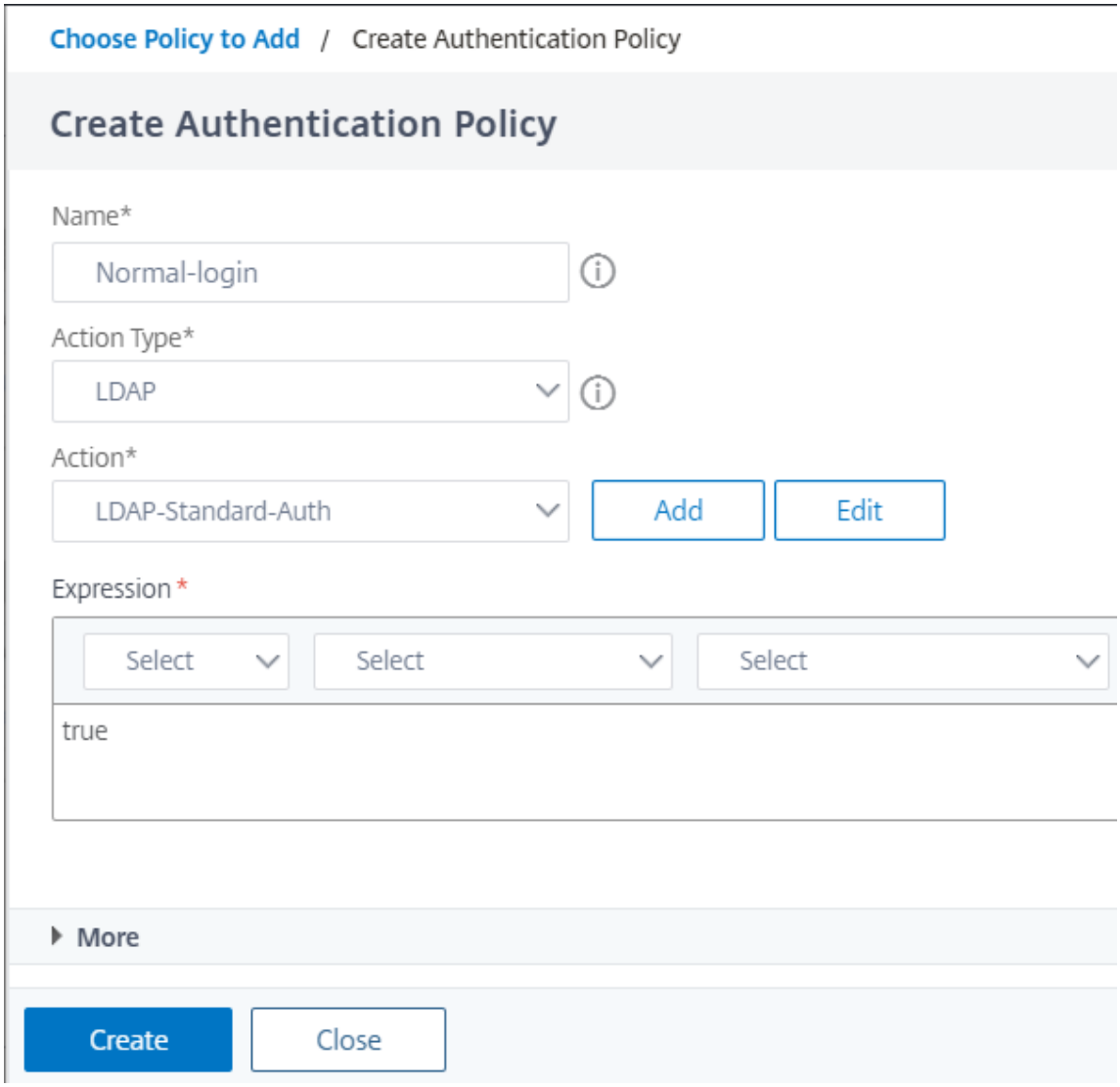
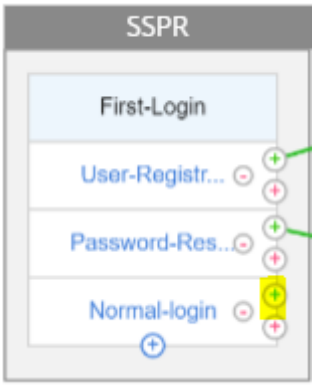
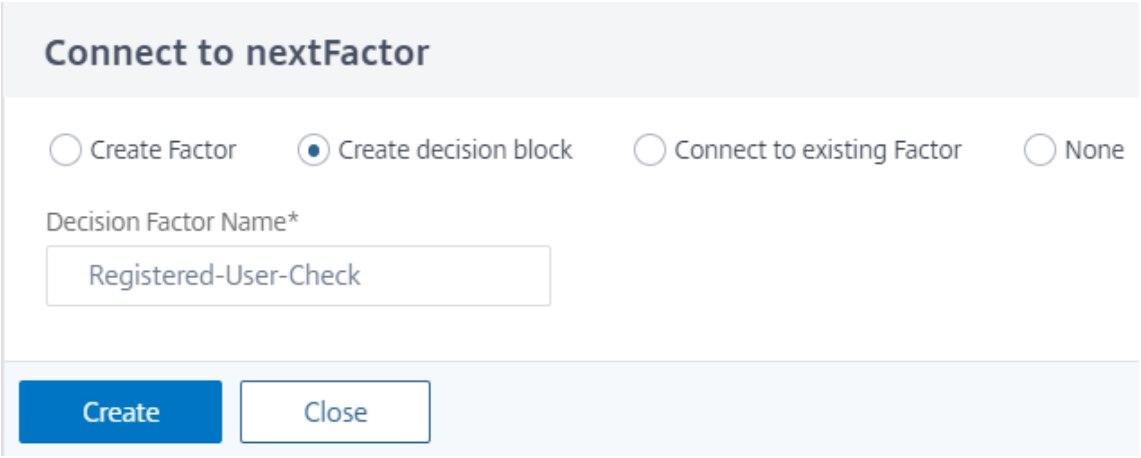
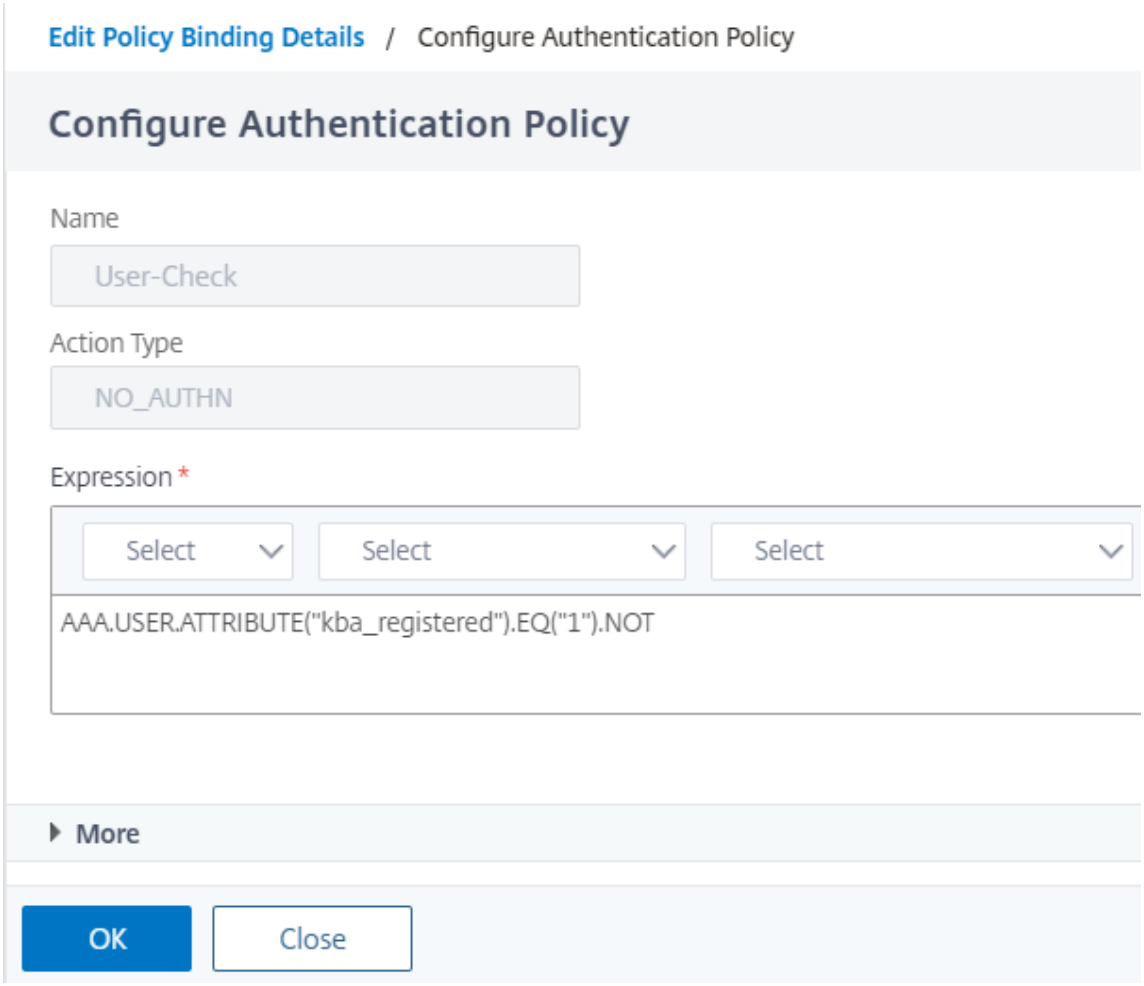
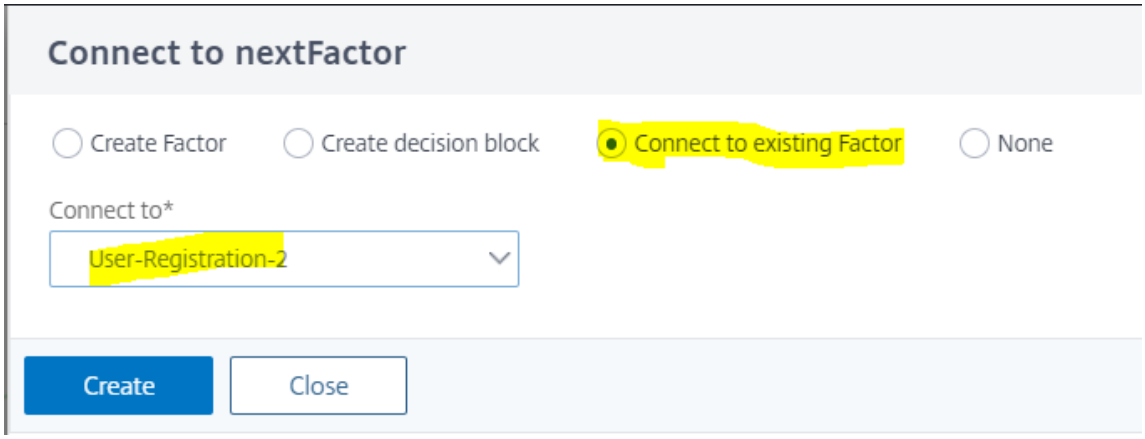
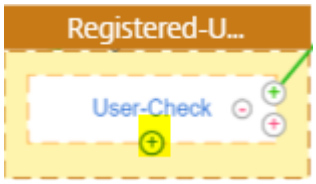
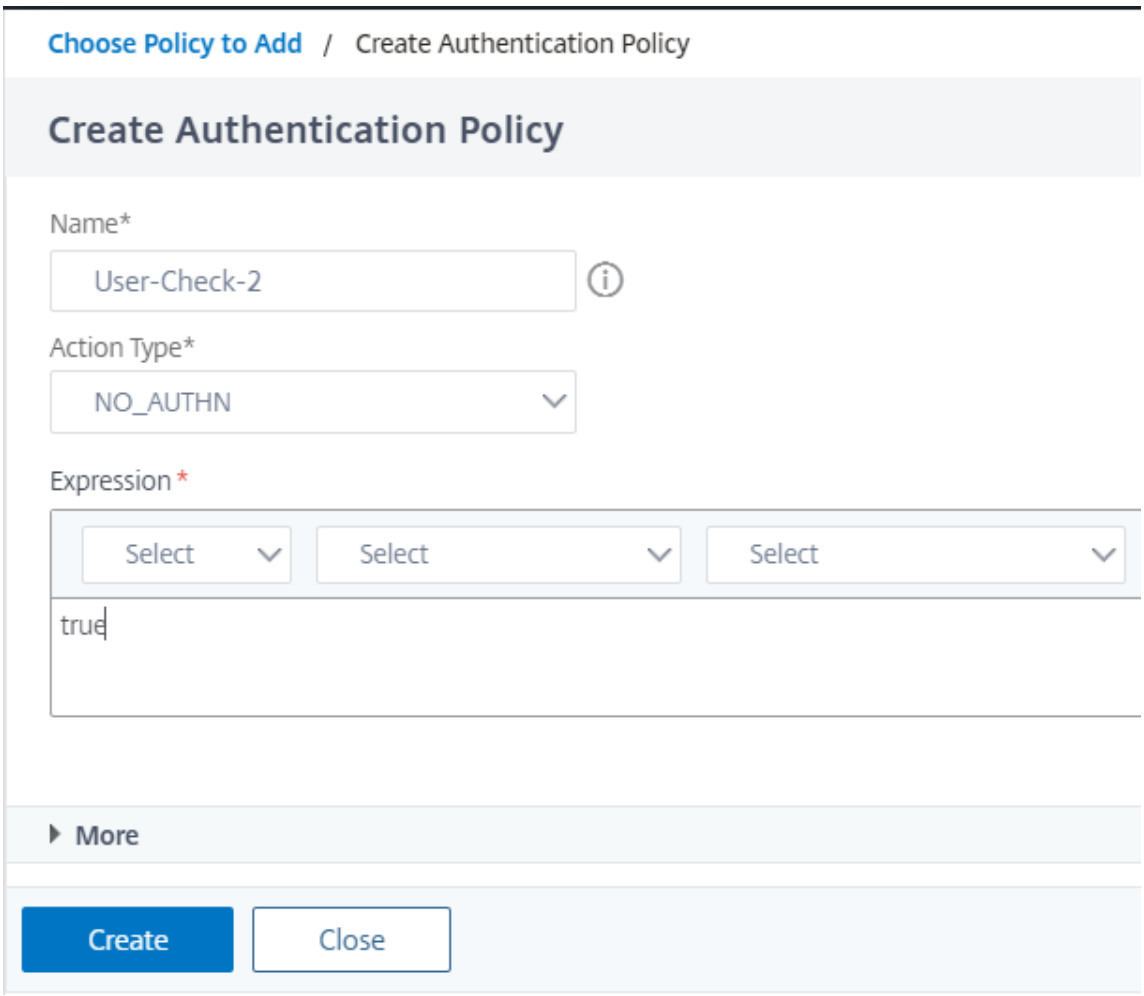
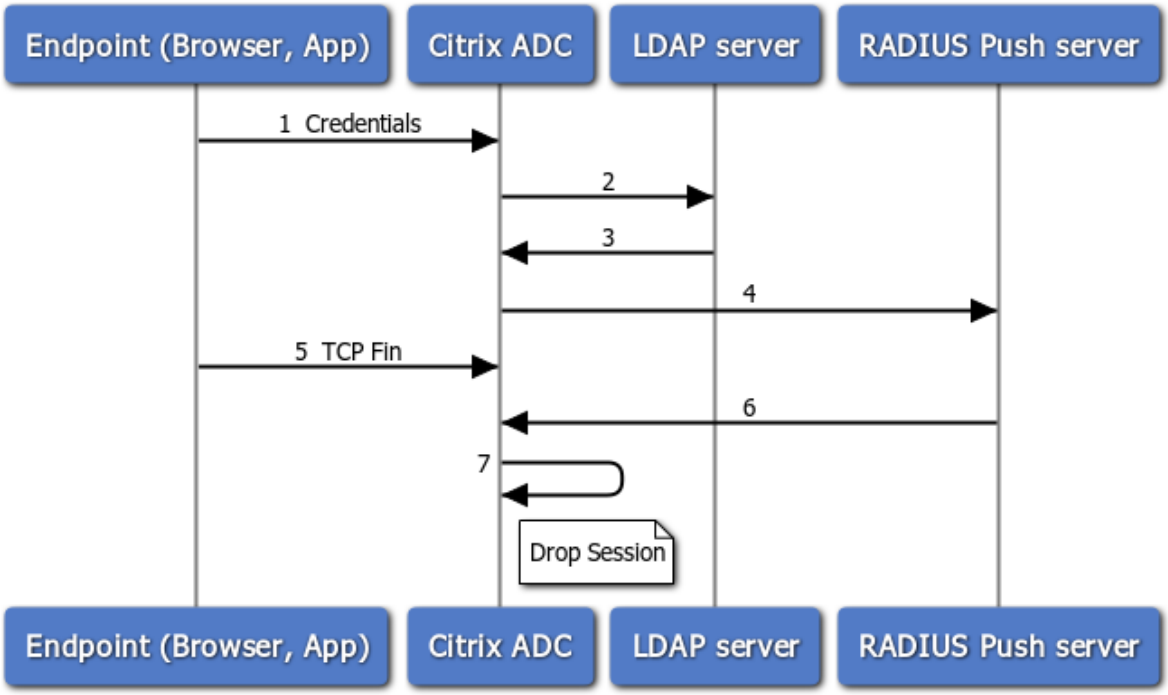
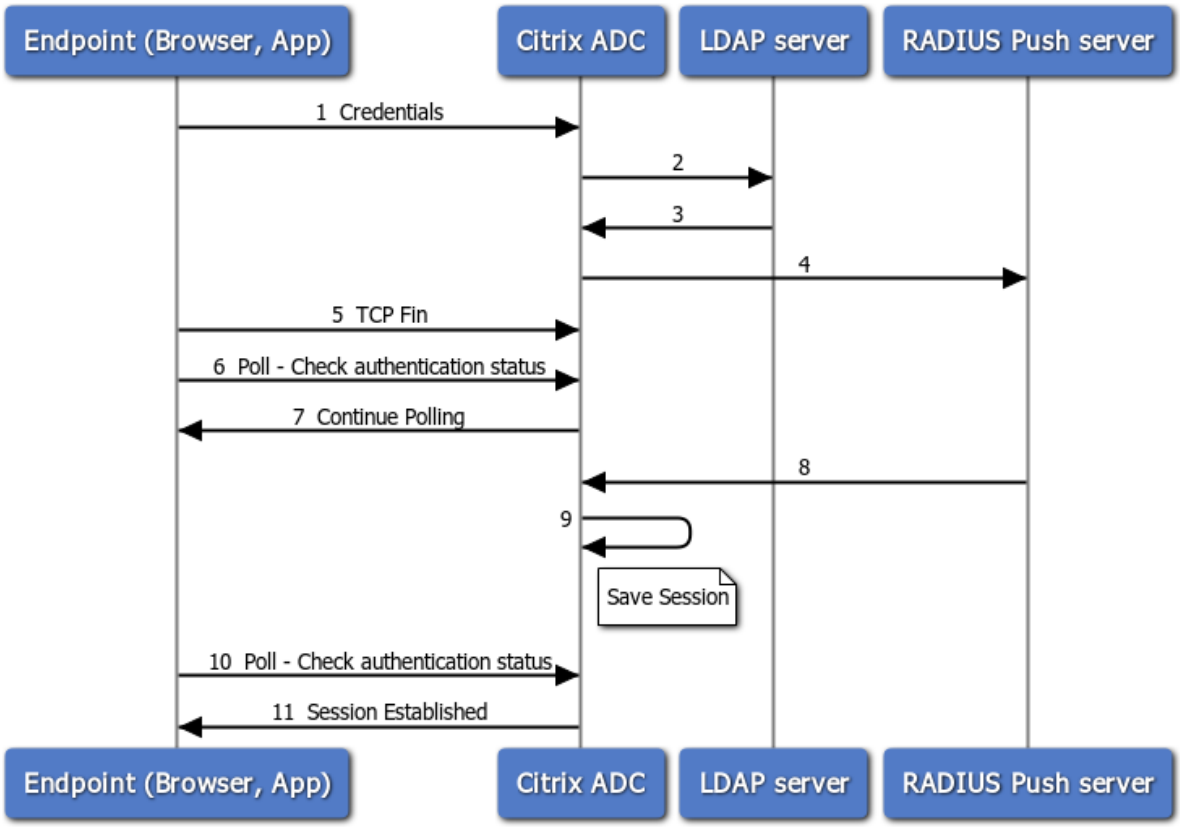
> bind lb vserver lb1 iis_backendsso_server_com
```

Create LDAP action with authentication disabled as first policy

```
> add authentication ldapAction ldap3 -serverIP 1.2.3.4 -serverPort 636
     -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName samAccountName -authentication disabled

> add authentication policy ldap3 -rule aaa.LOGIN.VALUE("passwdreset").
    EQ("1") -action ldap3
```

Create knowledge-based question and answer validation action

For knowledge-based question and answer validation in self-service password reset flow, you need to configure LDAP server with authentication disabled.

```
> add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP
    > -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
    KBAttribute  <LDAP ATTRIBUTE> - alternateEmailAttr <LDAP ATTRIBUTE>
    -authentication DISABLED
```

Example:

```
> add authentication ldapAction ldap2 -serverIP 1.2.3.4 -serverPort 636
     -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName samAccountName -KBAttribute userParameters -
    alternateEmailAttr userParameters -authentication disabled
```


To create authentication policy for knowledge-based question and answer validation using CLI

```
add authentication policy kba_validation -rule true -action ldap2
```

Create email validation action

LDAP must be a prior factor to email validation factor because you need the user's email ID or alternate email ID as part of the self-service password reset registration.

> **Note:**
>
> For Email OTP solution to work, ensure that the login based authentication is enabled on the SMTP server.

To ensure that the login based authentication is enabled, type the following command on the SMTP server. If the login based authentication is enabled, you will notice that the text **AUTH LOGIN** appears in bold in the output.

```
root@ns# telnet <IP address of the SMTP server><Port number of the
   server>
ehlo
```

Example:

```
root@ns# telnet 10.106.3.66 25
Trying 10.106.3.66...
Connected to 10.106.3.66.
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22
   Nov 2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.41.151]
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
```


For information on how to enable login based authentication, see https://support.microfocus.com/kb/doc.php?id=7020367.

To configure email action using CLI

```
add authentication emailAction emailact -userName sender@example.com -password <Password> -serverURL "smtps://smtp.example.com:25" -content "OTP is $code"
```

Example:

```
add authentication emailAction email -userName testmail@gmail.com -password 298a34b1a1b7626cd5902bbb416d04076e5ac4f357532e949db94c0534832670 -encrypted -encryptmethod ENCMTHD_3 -serverURL "smtps://10.19.164.57:25" -content "OTP is $code" -emailAddress "aaa.user.attribute(\"alternate_mail\")"
```

> **Note**
>
> The "emailAddress" parameter in the configuration is a PI expression. Hence, this is configured to take either the default user email ID from the session or the already registered alternative email ID.

To configure email ID using GUI

1. Navigate to **Security > AAA – Application Traffic > policies > Authentication > Advanced Policies > Actions > Authentication Email Action**. Click **Add**.
2. On the **Create Authentication Email Action** page, fill the details, and click **Create**.



To create authentication policy for email validation by using CLI

```
1 add authentication policy email_validation -rule true -action email
```

To create authentication policy for password reset factor

```
1 add authentication policy ldap_pwd -rule true -action ldap_logon_action
```

Presenting UI through Login Schema

There are three LoginSchema's for self-service password reset to reset the password. Use the following CLI commands to view the three Login Schema:


```
root@ns# cd  /nsconfig/loginschema/LoginSchema/
root@ns# ls -ltr | grep -i password
-r--r--r--  1 nobody  wheel  2088 Nov 13 08:38
   SingleAuthPasswordResetRem.xml
-r--r--r--  1 nobody  wheel  1541 Nov 13 08:38
   OnlyUsernamePasswordReset.xml
-r--r--r--  1 nobody  wheel  1391 Nov 13 08:38 OnlyPassword.xml
```

To create single authentication password reset by using CLI

```
> add authentication loginSchema lschema_password_reset -
   authenticationSchema "/nsconfig/loginschema/LoginSchema/
   SingleAuthPasswordResetRem.xml"

> add authentication loginSchemaPolicy lpol_password_reset -rule true -
   action lschema_password_reset
```

Create knowledge-based question and answer and email OTP validation factor through policy label

If the first factor is LDAP logon, you can create a knowledge-based question and answer and email OTP policy labels for the next factor using the following commands.

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel kba_validation -loginSchema
   lschema_noschema

> add authentication policylabel email_validation -loginSchema
   lschema_noschema
```

Create password reset factor through policy label

You can create password reset factor through policy label by using the following commands.

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel password_reset -loginSchema
   lschema_noschema
```



```
4
5 > bind authentication policylabel password_reset -policyName ldap_pwd -
      priority 10 -gotoPriorityExpression NEXT
```

Bind the knowledge-based question and answer and email policy to the previous created policies using the following commands.

```
1 > bind authentication policylabel email_validation -policyName
      email_validation -nextfactor password_reset -priority 10 -
      gotoPriorityExpression NEXT
2
3 > bind authentication policylabel kba_validation -policyName
      kba_validation -nextfactor email_validation -priority 10 -
      gotoPriorityExpression NEXT
```

Bind the flow

You must have the LDAP logon flow created under authentication policy for LDAP Logon. In this flow, user clicks on forgot password link presented on first LDAP logon page, then KBA validation followed by OTP validation and finally password reset page.

```
1 bind authentication vserver authvs -policy ldap3 -nextfactor
      kba_validation -priority 10 -gotoPriorityExpression NEXT
```

To bind all the UI flow

```
1 bind authentication vserver authvs -policy lpol_password_reset -
      priority 20 -gotoPriorityExpression END
```

User logon workflow to reset password

Following is a user logon workflow if the user needs to reset password:

1. Enter the lb vserver URL; for example, https://lb1.server.com. The logon screen is displayed.



2. Click **Forgot Password**. A validation screen displays two questions out of maximum of six questions and answers registered against an AD user.



3. Answer the questions, and click **Log on**. An email OTP Validation screen where you must enter the OTP received on the registered alternate email ID, is displayed.



4. Enter the email OTP. Once the email OTP validation is successful, the password reset page is displayed.



5. Enter a new password and confirm the new password. Click **Submit**. After the password reset is successful, the password reset successful screen is displayed.



You can now logon using the reset password.

Troubleshooting

Citrix provides an option to troubleshoot some of the basic issues that you might face while using self-service password reset. The following section helps you troubleshoot some of the issues that might occur in specific areas.

NS Log

Before analyzing the log, it is recommended to set the log level to debug using the following command:

```
1 > set syslogparams -loglevel DEBUG
```

Registration

The following message indicates a successful user registration.

```
1 "ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1"
```



```
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  "
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw=="
```

Knowledge-based question and answer validation

The following message indicates successful knowledge-based question and answer validation.

```
"NFactor: Successfully completed KBA Validation, nextfactor is email"
```

Email ID validation

The following message indicates successful password reset.

```
"NFactor: Successfully completed email auth, nextfactor is pwd_reset"
```

Configure SSPR using nFactor visualizer

Before we begin the SSPR configuration, we need to add the following LDAP servers:

1. Standard LDAP server with authentication enabled for user authentication and AD attribute specified.



2. LDAP server for user parameter extraction with no auth.



3. LDAP server for password reset on SSL with no auth. Also, the AD attribute to be used for storing the user details needs to be defined in this server.



4. LDAP server for user registration, with auth enabled, and AD attribute specified



5. The complete flow is shown below:



6. Bind the certificate globally using the following CLI command:

```
1 bind vpn global -userDataEncryptionKey Wildcard
```

Now that the LDAP servers are added, proceed with the nFactor configuration using visualizer

1. Navigate to, **Security > AAA > Application Traffic > nFactor Visualizer > nFactor Flows**, click **Add** and click on the plus icon inside the box.

2. Give the flow a name.



3. Click **Add Schema**, which will serve as the default schema. Click **Add** on the login schema page.



4. After giving the schema a name, select the schema as shown below. Click **Select** on the top right corner for the schema to be selected.



5. Click **Create** and click **OK**.

Once the default Schema is added, then we have to configure the following three flows:

- **User registration**: For explicit user registration
- **Password reset**: For password reset
- **Normal login + Registered user check**: In case the user is registered and enters correct password, the user will be logged in. In case the user is not registered it will take the user to registration page.

User Registration

Let us continue from where we left after adding the schema.

1. Click **Add Policy**, this will check if the user is trying to explicitly register.





2. Click **Create** and then click **Add**.
3. Click on the highlighted green '+' icon, to add the next authentication factor to the user-registration flow.





4. Click **Create**.
5. Click **Add Policy** for User-Registration-1 factor.



6. Create the authentication policy. This policy extracts the user information and validates it before redirecting it to the registration page.



7. Click **Create** and then click **Add**.
8. Now click on the green '+' icon to create another factor for the user registration and click **Create**. Click **Add Schema**.



9. Create the following schema.



10. Click **Add Policy** and create the following authentication policy.



11. Click **Create** and click **Add**.

Password reset

1. Click the Blue '+' icon to add another policy (Password reset flow) for the parent SSPR factor.



2. Click **Add** and create the below authentication policy. This policy will be triggered if the user

clicks “Forgot password” on the login page.



3. Click **Create** and click **Add**.
4. Click the green ‘+’ icon for the password reset authentication policy to add another factor.



5. Click **Create**.
6. Click **Add policy** to create an authentication policy for the above created factor. This factor will be for validating the user.



7. Click **Create** and click **Add**.
8. Click on the Green '+' icon to add another factor for the password factor flow, this will validate the answers provided for resetting the password. Click **Create**.



9. Click **Add Policy** to add authentication policy for the factor.
10. Select the same authentication policy from the drop-down menu that we created earlier and click **Add**.



Normal login + Registered user check

1. Click the blue '+' icon to add another authentication policy (Normal login flow) to the parent SSPR factor.



2. Click **Add**, to create the below Authentication policy for normal user login.



3. Click on **Create** and click on **Add**.
4. Click the green '+' icon for the above created policy to add another factor i.e decision block.

Click **Create**.



5. Click **Create**.



6. Click **Add Policy** to create an authentication policy for this decision factor.



7. Click **Create** then click **Add**. This will check if the user is registered or not.
8. Click on the green '+' icon to point the user to the registration policy.

9. Select the registration factor from the drop-down menu and click **Create**.



10. Now click the blue '+' icon to add another policy to the decision block, this policy will be for the registered user to end the auth.



11. Click **Add Policy** to create the below authentication policy.



12. Click **Create** and click **Add**.

Polling during authentication

September 14, 2021

Starting from Citrix ADC release build 13.0.79.64, a Citrix ADC appliance can be configured for Polling mechanism during multifactor authentication.

If Polling is configured on a Citrix ADC appliance, endpoints (like a web browser or an app) can poll (probe) the appliance during authentication at the configured intervals to get the status of the submitted authentication request.

Polling can be configured to handle authentications when an endpoint drops a TCP connection while authenticating with a Citrix ADC appliance.

Points to note

- The Polling configuration is supported for LDAP, RADIUS, and TACACS authentication methods.
- Client can probe authentication requests from second factor onwards.

Why configure Polling?

Sometimes while authenticating, switching between the apps (for example a login app and an authenticator app) causes endpoints to lose connection with the Citrix ADC appliance leading to a break in the authentication flow. With Polling configured, this break in authentication can be avoided.

Understanding the Polling mechanism

The following is an example for the flow of events during authentication without Polling configured.

The Polling mechanism enables a Citrix ADC appliance to resume an ongoing authentication with the endpoint without having to restart the authentication process in a rare case of a TCP connection reset at the endpoint.



1. An endpoint (App or Web browser) authenticates with credentials.
2. The user name and password is verified against an existing first factor directory (LDAP/Active Directory).
3. If the correct credentials are supplied, the authentication moves to the next factor.
4. At this point, the Citrix ADC appliance sends request to the RADIUS Push server.

5. While the Citrix ADC appliance waits for a response from the RADIUS server, the endpoint drops TCP connection.
6. The Citrix ADC receives a response from the RADIUS Push server.
7. As no client TCP connection is found, the Citrix ADC appliance drops session and the login fails.

The following is an example for the flow of events during authentication with Polling configured.



1. An endpoint (App or Web browser) authenticates with credentials.
2. The user name and password is verified against an existing first factor directory (LDAP/Active Directory).
3. If the correct credentials are supplied, the authentication moves to the next factor.
4. At this point, the Citrix ADC appliance sends request to the RADIUS Push server.
5. While the Citrix ADC appliance waits for a response from the RADIUS server, the endpoint drops TCP connection.
6. Endpoint sends a poll (probe) to the Citrix ADC appliance to check for the authentication status.
7. As the Citrix ADC appliance does not hear back from the RADIUS server, it requests the endpoint to continue polling.
8. The Citrix ADC appliance receives response from the RADIUS Push server.
9. As no client TCP connection is found, ADC saves the session state.
10. Endpoint again polls to check for the authentication status.
11. Citrix ADC appliance establishes the session and the login succeeds.

Configure Polling using CLI

The following is a sample CLI configuration.

Configure First factor

```
add authentication ldapAction ldap-new -serverIP 10.106.40.65 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword 2
    f63d3659103464a4fad0ade65e2ccfd4e8440e36ddff941d29796af03e01139 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -
    groupAttrName memberof -subAttributeName CN -secType SSL -
    alternateEmailAttr userParameters

add authentication Policy ldap-new -rule true -action ldap-new

bind authentication vserver avs -policy ldap-new -priority 1 -
    nextFactor rad_factor
<!--NeedCopy-->
```

Configure Second factor

```
add authentication radiusAction rad1 -serverIP 10.102.229.120 -radKey 1
    b1613760143ce2371961e9a9eb5392c86a4954a62397f29a01b5d12b42ce232 -
    encrypted -encryptmethod ENCMTHD_3

add authentication Policy rad -rule true -action rad1
<!--NeedCopy-->
```

Configure Poll.xml login schema

```
add authentication loginSchema polling_schema -authenticationSchema
    LoginSchema/Poll.xml

add authentication policylabel rad_factor -loginSchema polling_schema

bind authentication policylabel rad_factor -policyName rad -priority 1
    -gotoPriorityExpression NEXT
<!--NeedCopy-->
```


Configure Polling using GUI

For detailed steps on configuring multifactor authentication using GUI see, Configuring nFactor authentication

Following are the sample high level steps required for configuring Citrix ADC for Polling from second factor onwards.

1. Create a first factor for authentication, for example LDAP.
2. Create a second factor for authentication, for example RADIUS.
3. Add **Poll.xml** present in Citrix ADC (/nsconfig/loginschema/LoginSchema/) as login schema for the second factor.

Session and traffic management

September 14, 2021

Session settings

After you configure your authentication, authorization, and auditing profiles, you configure session settings to customize your user sessions. The session settings are:

- **The session timeout.**

  Controls the period after which the user is automatically disconnected and must authenticate again to access your intranet.

- **The default authorization setting.**

  Determines whether the Citrix ADC appliance will by default allow or deny access to content for which there is no specific authorization policy.

- **The single sign-on setting.**

  Determines whether the Citrix ADC appliance will log users on to all web applications automatically after they authenticate, or will pass users to the web application logon page to authenticate for each application.

- **The credential index setting.**

  Determines whether the Citrix ADC appliance uses the primary or the secondary authentication credentials for single sign-on.

To configure the session settings, you can take one of two approaches. If you want different settings for different user accounts or groups, you create a profile for each user account or group for which you want to configure custom sessions settings. You also create policies to select the connections to which

to apply particular profiles, and you bind the policies to users or groups. You can also bind a policy to the authentication virtual server that handles the traffic to which you want to apply the profile.

If you want the same settings for all sessions, or if you want to customize the default settings for sessions that do not have specific profiles and policies configured, you can simply configure the global session settings.

Session profiles

To customize your user sessions, you first create a session profile. The session profile allows you to override global settings for any of the session parameters.

> **Note**
>
> The terms “session profile” and “session action” mean the same thing.

To create a session profile by using the command line interface

At the command prompt, type the following commands to create a session profile and verify the configuration:

```
add tm sessionAction <name> [-sessTimeout <mins>] [-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
     )] [-persistentCookieValidity <minutes>]

show tm sessionAction <name>
<!--NeedCopy-->
```

Example

```
> add tm sessionAction session-profile -sessTimeout 30 -
    defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```


To modify a session profile by using the command line interface

At the command prompt, type the following commands to modify a session profile and verify the configuration:

```
set tm sessionAction <name> [-sessTimeout <mins>] [-
   defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
   ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
   httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]

show tm sessionAction
<!--NeedCopy-->
```

Example

```
> set tm sessionAction session-profile -sessTimeout 30 -
   defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

To remove a session profile by using the command line interface

At the command prompt, type the following command to remove a session profile:

```
rm tm sessionAction <name>
<!--NeedCopy-->
```

To configure session profiles by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Session.**
2. Navigate to **Security > AAA - Application Traffic > Policies > Session.**
3. In the details pane, click the **Profiles** tab.
4. On the **Profiles** tab, do one of the following:
   - To create a new session profile, click **Add.**
   - To modify an existing session profile, select the profile, and then click **Edit.**
5. In the Create TM Session Profile or Configure TM Session Profile dialog, type or select values for the parameters.

- Name*—actionname (Cannot be changed for a previously configured session action.)
- Session Time-out—sesstimeout
- Single sign-on to Web Applications—sso
- Default Authorization Action—defaultAuthorizationAction
- Credential Index—ssocredential
- Single Sign-on Domain—ssoDomain
- HTTPOnly Cookie—httpOnlyCookie
- Enable Persistent Cookie—persistentCookie
- Persistent Cookie Validity—persistentCookieValidity

6. Click **Create** or **OK**. The session profile that you created appears in the Session Policies and Profiles pane.

Session policies

After you create one or more session profiles, you create session policies and then bind the policies globally or to an authentication virtual server to put them into effect.

To create a session policy by using the command line interface

At the command prompt, type the following commands to create a session policy and verify the configuration:

```
-  add tm sessionPolicy <name> <rule> <action>
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

Example

```
> add tm sessionPolicy session-pol "URL == /*.png" session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

To modify a session policy by using the command line interface

At the command prompt, type the following commands to modify a session policy and verify the configuration:


```
-  set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

Example

```
> set tm sessionPolicy session-pol "URL == /*.png" session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

To globally bind a session policy by using the command line interface

At the command prompt, type the following commands to globally bind a session policy and verify the configuration:

```
bind tm global -policyName <policyname> [-priority <priority>]
<!--NeedCopy-->
```

Example

```
> bind tm global -policyName session-pol
 Done

> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.png'
        Action: session-profile
        Policy is bound to following entities
        1) TM GLOBAL    PRIORITY : 0
 Done

<!--NeedCopy-->
```

To bind a session policy to an authentication virtual server by using the command line interface

At the command prompt, type the following command to bind a session policy to an authentication virtual and verify the configuration:


```
bind authentication vserver <name> -policy <policyname> [-priority <
    priority>]
<!--NeedCopy-->
```

Example

```
bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -
    priority 1000
Done
<!--NeedCopy-->
```

To unbind a session policy from an authentication virtual server by using the command line interface

At the command prompt, type the following commands to unbind a session policy from an authentication virtual server and verify the configuration:

```
unbind authentication vserver <name> -policy <policyname>
<!--NeedCopy-->
```

Example

```
unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
Done
<!--NeedCopy-->
```

To unbind a globally bound session policy by using the command line interface

At the command prompt, type the following commands to unbind a globally bound session policy:

```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```

Example

```
unbind tm global -policyName Session-Pol-1
Done
<!--NeedCopy-->
```


To remove a session policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type the following commands to remove a session policy and verify the configuration:

```
rm tm sessionPolicy <name>
<!--NeedCopy-->
```

Example

```
rm tm sessionPolicy Session-Pol-1
Done

<!--NeedCopy-->
```

To configure and bind session policies by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Session**.
2. Navigate to **Security > AAA - Application Traffic > Policies > Session**.
3. In the details pane, on the **Policies** tab, do one of the following:
   - To create a new session policy, click **Add**.
   - To modify an existing session policy, select the policy, and then click **Edit**.
4. In the **Create Session Policy** or **Configure Session Policy** dialog, type or select the values for the parameters.
   - Name*—policyname (Cannot be changed for a previously configured session policy.)
   - Request Profile*—actionname
   - Expression*—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking **Add** to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click **Create** or **OK**. The policy that you created appears in the details pane of the **Session Policies** and **Profiles** page.
6. To globally bind a session policy, in the details pane, select **Global Bindings** from the **Action** drop-down list, and fill in the dialog.
   - Select the name of the session policy you want to globally bind.
   - Click **OK**.
7. To bind a session policy to an authentication virtual server, in the navigation pane, click **Virtual Servers**, and add that policy to the policies list.
   - In the details pane, select the virtual server, and then click **Edit**.
   - In the **Advanced Selections** to the right of the detail area, click **Policies**.
   - Select a policy, or click the **plus** icon to add a policy.

- In the **Priority** column to the left, modify the default priority to ensure that the policy is evaluated in the proper order.
- Click **OK**.
  A message appears in the status bar, stating that the policy has been configured successfully.

Global session settings

In addition to or instead of creating session profiles and policies, you can configure global session settings. These settings control the session configuration when there is no explicit policy overriding them.

To configure the session settings by using the command line interface

At the command prompt, type the following commands to configure the global session settings and verify the configuration:

```
set tm sessionParameter [-sessTimeout <mins>][-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )][-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )][-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]
<!--NeedCopy-->
```

Example

```
> set tm sessionParameter -sessTimeout 30
  Done
> set tm sessionParameter -defaultAuthorizationAction DENY
  Done
> set tm sessionParameter -SSO ON
  Done
> set tm sessionParameter -ssoCredential PRIMARY
  Done
<!--NeedCopy-->
```

To configure the session settings by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic**
2. In the details pane, under **Settings**, click Change global settings.
3. In the **Global Session Settings** dialog, type or select values for the parameters.
   - Session Time-out—sessTimeout

- Default Authorization Action—defaultAuthorizationAction
- Single Sign-on to Web Applications—sso
- Credential Index—ssoCredential
- Single Sign-on Domain—ssoDomain
- HTTPOnly Cookie—httpOnlyCookie
- Enable Persistent Cookie—persistentCookie
- Persistent Cookie Validity (minutes)—persistentCookieValidity
- Home Page—home page

4. Click **OK**.

Traffic settings

If you use forms-based or SAML single sign-on (SSO) for your protected applications, you configure that feature in the Traffic settings. SSO enables your users to log on once to access all protected applications, rather than requiring them to log on separately to access each one.

Forms-based SSO allows you to use a web form of your own design as the sign-on method instead of a generic pop-up window. You can therefore put your company logo and other information you might want your users to see on the logon form. SAML SSO allows you to configure one Citrix ADC appliance or virtual appliance instance to authenticate to another Citrix ADC appliance on behalf of users who have authenticated with the first appliance.

To configure either type of SSO, you first create a forms or SAML SSO profile. Next, you create a traffic profile and link it to the SSO profile you created. Next, you create a policy, link it to the traffic profile. Finally, you bind the policy globally or to an authentication virtual server to put your configuration into effect.

Traffic profiles

After creating at least one forms or SAML sso profile, you must next create a traffic profile.

> **Note:**
>
> In this feature, the terms “profile” and “action” mean the same thing.

To create a traffic profile by using the command line interface

At the command prompt, type:

```
1 add tm trafficAction <name> [-appTimeout <mins>][-SSO ( ON | OFF ) [-
     formSSOAction <string>]][-persistentCookie ( ENABLED | DISABLED )][-
     InitiateLogout ( ON | OFF )]
2 <!--NeedCopy-->
```


Example

```
add tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

To modify a session profile by using the command line interface

At the command prompt, type:

```
set tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF ) [-
    formSSOAction <string>]] [-persistentCookie ( ENABLED | DISABLED )]
    [-InitiateLogout ( ON | OFF )]
<!--NeedCopy-->
```

Example

```
set tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

To remove a session profile by using the command line interface

At the command prompt, type:

```
rm tm trafficAction <name>
<!--NeedCopy-->
```

Example

```
rm tm trafficAction Traffic-Prof-1
<!--NeedCopy-->
```

To configure traffic profiles by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Traffic.**
2. Navigate to **Security > AAA - Application Traffic > Policies > Traffic.**
3. In the details pane, click the Profiles tab.
4. On the Profiles tab, do one of the following:
   - To create a new traffic profile, click **Add.**
   - To modify an existing traffic profile, select the profile, and then click **Edit.**

5. In the **Create Traffic Profile** or **Configure Traffic Profile** dialog box, specify values for the parameters.
   - Name*—name (Cannot be changed for a previously configured session action.)
   - AppTimeout—appTimeout
   - Single Sign-On—SSO
   - Form SSO Action—formSSOAction
   - SAML SSO Action—samlSSOAction
   - Enable Persistent Cookie—persistentCookie
   - Initiate Logout—InitiateLogout
6. Click **Create** or **OK**. The traffic profile that you created appears in the Traffic Policies, Profiles, and either the Form SSO Profiles or SAML SSO Profiles pane, as appropriate.

Support for AAA.USER and AAA.LOGIN expressions

The AAA.USER expression is now implemented to replace the existing HTTP.REQ.USER expressions. The AAA.USER expression is applicable to handle non-HTTP traffic, such as the Secure Web Gateway (SWG) and role-based access (RBA) mechanism. The AAA.USER expressions are equivalent to HTTP.REQ.USER expressions.

You can use the expression at various actions or profiles configuration.

At the command prompt, type:

```
add tm trafficAction <name> [SSO (ON|OFF)] [-userExpression <string>]

add tm trafficAction <name> [SSO (ON|OFF)] [-passwdExpression <string>]

<!--NeedCopy-->
```

Example

```
add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.NAME"

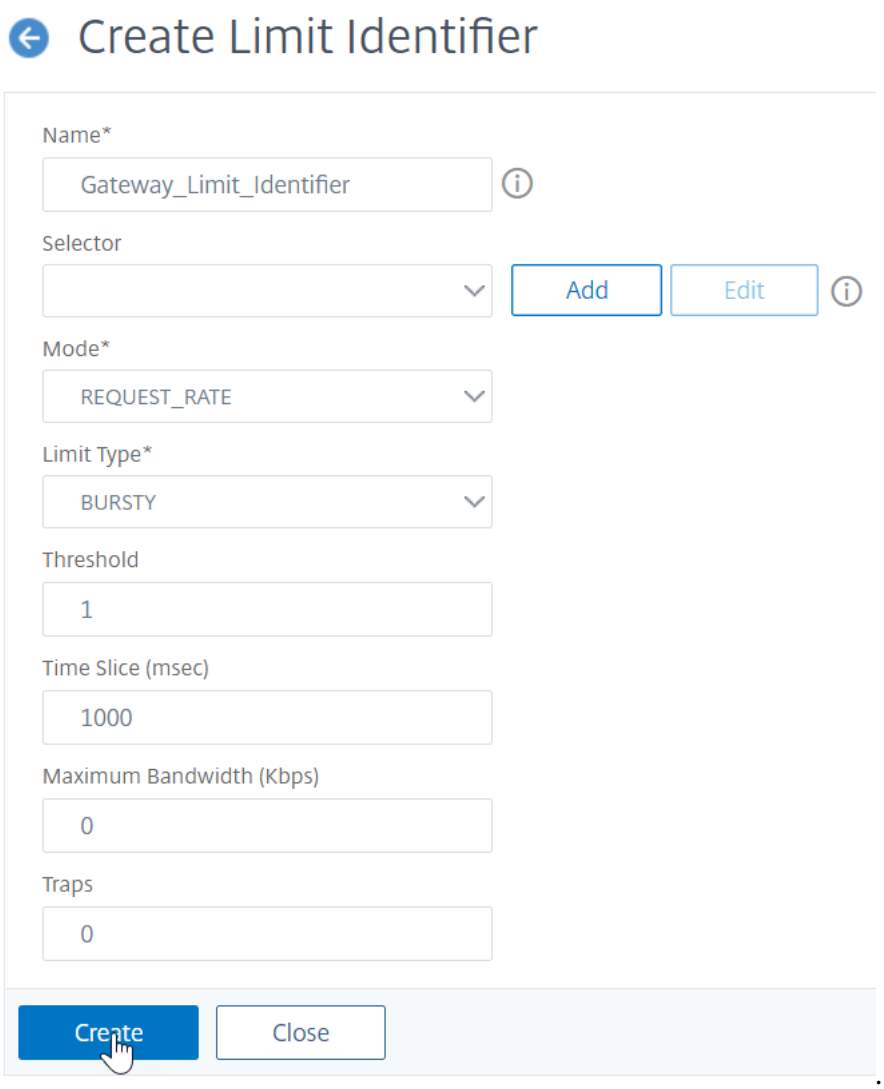
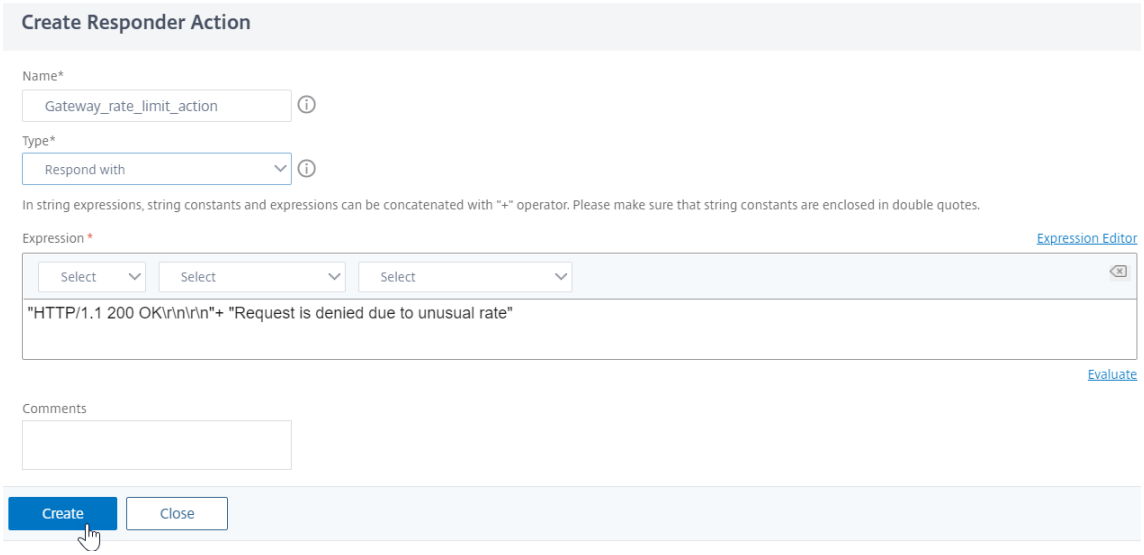
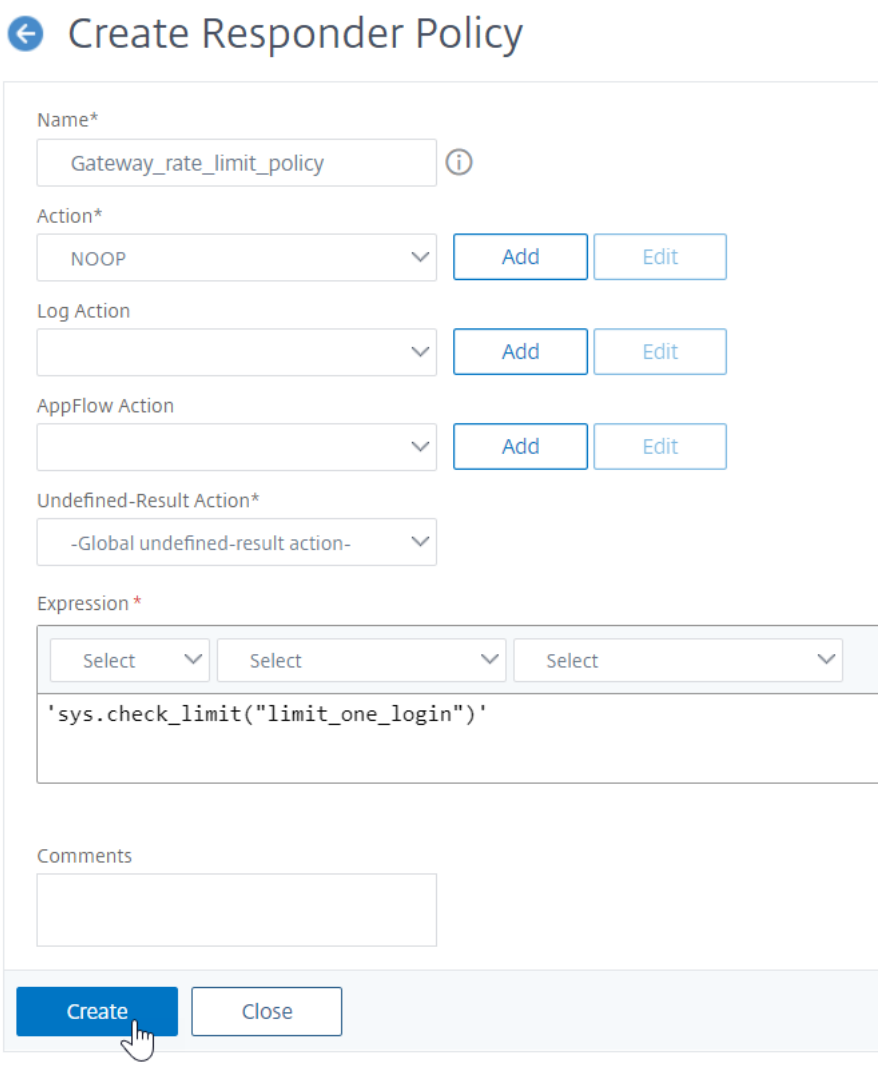
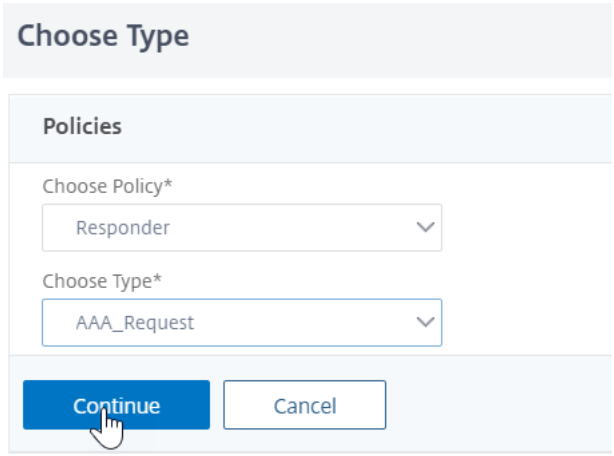
add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.PASSWD"

add tm trafficPolicy tm_pol true tm_act

bind lb vserver lb1 -policyName tm_pol -priority 2
<!--NeedCopy-->
```

> **Note:**

> If you use HTTP.REQ.USER expression, a warning message “HTTP.REQ.USER has been deprecated. Use AAA.USER instead” appears on the command prompt.

- **AAA.LOGIN Expression.** The LOGIN expression represents pre-login, also known as the login request. The login request can be from Citrix Gateway, SAML IdP, or from OAuth authentication. The Citrix ADC will abstract the required attributes from the policy configuration. The AAA.LOGIN expression contains the attributes, which can be fetched based on the following:
  - **AAA.LOGIN.USERNAME.** The user name (if found) is fetched from the current login request. The same expression applied to a non-login request (determined by an authentication, authorization, and auditing) results in an empty string.
  - **AAA.LOGIN.PASSWORD.** The user password (if found) is fetched from the current login request. The expression results in an empty string if the password is not found.
  - **AAA.LOGIN.PASSWORD2.** The second password (if found) is fetched from the login request.
  - **AAA.LOGIN.DOMAIN.** The domain information is fetched from the login request.
- **AAA.USER.ATTRIBUTE(“#”).** The expression is used to store user attribute. Here # can either be an integer value (between 1 and 16) or a string value. You can use these index values by using the expression AAA.USER.ATTRIBUTE(“#”). The authentication, authorization, and auditing module looks up the user sessions attribute and `AAA.USER.ATTRIBUTE("##")` would query the hash table for that particular attribute. For example, if `Attributes("samaccountname")` is set, `AAA.USER.ATTRIBUTE("samaccountname")` would query the hash map and would fetch the value corresponding to `samaccountname`.

Traffic policies

After you create one or more form SSO and traffic profiles, you create traffic policies and then bind the policies, either globally or to a traffic management virtual server, to put them into effect.

To create a traffic policy by using the command line interface

At the command prompt, type:

```
1 add tm trafficPolicy <name> <rule> <action>
2 <!--NeedCopy-->
```

Example

```
1 add tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS(
     "login=true")" Traffic-Prof-1
2 <!--NeedCopy-->
```


To modify a traffic policy by using the command line interface

At the command prompt, type:

```
set tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```

Example

```
set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS(
   "login=true")" Traffic-Prof-1
<!--NeedCopy-->
```

To globally bind a traffic policy by using the command line interface

At the command prompt, type:

```
bind tm global -policyName <string> [-priority <priority>]
<!--NeedCopy-->
```

Example

```
bind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

To bind a traffic policy to a load balancing or content switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

```
bind lb vserver <name> -policy <policyName> [-priority <priority>]

bind cs vserver <name> -policy <policyName> [-priority <priority>]
<!--NeedCopy-->
```

Example

```
bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -
   priority 1000
<!--NeedCopy-->
```


To unbind a globally bound traffic policy by using the command line interface

At the command prompt, type:

```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```

Example

```
unbind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

To unbind a traffic policy from a load balancing or content switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

```
unbind lb vserver <name> -policy <policyname>

unbind cs vserver <name> -policy <policyname>
<!--NeedCopy-->
```

Example

```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1
<!--NeedCopy-->
```

To remove a traffic policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type:

```
rm tm trafficPolicy <name>
<!--NeedCopy-->
```

Example

```
rm tm trafficPolicy Traffic-Pol-1
<!--NeedCopy-->
```

To configure and bind traffic policies by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Traffic.**
2. Navigate to **Security > AAA - Application Traffic > Policies > Traffic.**

3. In the details pane, do one of the following:
   - To create a new session policy, click **Add.**
   - To modify an existing session policy, select the policy, and then click **Edit.**
4. In the **Create Traffic Policy** or **Configure Traffic Policy** dialog, specify values for the parameters.
   - Name*—policyName (Cannot be changed for a previously configured session policy.)
   - Profile*—actionName
   - Expression—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click **Create** or **OK**. The policy that you created appears in the details pane of the **Session Policies** and **Profiles** page.

Form SSO profiles

To enable and configure forms-based SSO, you first create an SSO profile.

> **Note**
>
> - Forms-based single sign-on does not work if the form is customized to include Javascript.
> - In this feature, the terms “profile” and “action” mean the same thing.

To create a form SSO profile by using the command line interface

At the command prompt, type:

```
add tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]

show tm formSSOAction [<name>]
<!--NeedCopy-->
```

Example

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
```



```
<!--NeedCopy-->
```

To modify a form SSO by using the command line interface

At the command prompt, type:

```
set tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]
<!--NeedCopy-->
```

Example

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

To remove a form SSO profile by using the command line interface

At the command prompt, type:

```
rm tm formSSOAction <name>
<!--NeedCopy-->
```

Example

```
rm tm sessionAction SSO-Prof-1
<!--NeedCopy-->
```

To configure form SSO profiles by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Policies > Traffic**.
2. In the details pane, click the **Form SSO Profiles** tab.
3. On the Form SSO Profiles tab, do one of the following:
   - To create a new form SSO profile, click **Add**.
   - To modify an existing form SSO profile, select the profile, and then click Edit.

4. In the **Create Form SSO Profile** or **Configure Form SSO Profile** dialog, specify the values for the parameters:
   - Name*—name (Cannot be changed for a previously configured session action.)
   - Action URL*—actionURL
   - User Name Field*—userField
   - Password Field*—passField
   - Expression*—ssoSuccessRule
   - Name Value Pair—nameValuePair
   - Response Size—responsesize
   - Extraction—nvtype
   - Submit Method—submitMethod
5. Click **Create** or **OK**, and then click **Close**. The form SSO profile that you created appears in the **Traffic Policies**, **Profiles**, and **Form SSO Profiles** pane.

SAML SSO profiles

To enable and configure SAML-based SSO, you first create a SAML SSO profile.

To create a SAML SSO profile by using the command line interface

At the command prompt, type:

```
add tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
<!--NeedCopy-->
```

Example

```
add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
<!--NeedCopy-->
```

To modify a SAML SSO by using the command line interface

At the command prompt, type:

```
set tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
```



```
<!--NeedCopy-->
```

Example

```
set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
<!--NeedCopy-->
```

To remove a SAML SSO profile by using the command line interface

At the command prompt, type:

```
rm tm samlSSOProfile <name>
<!--NeedCopy-->
```

Example

```
rm tm sessionAction saml-SSO-Prof-1
<!--NeedCopy-->
```

To configure a SAML SSO profile by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Policies > Traffic.**
2. In the details pane, click the **SAML SSO Profiles** tab.
3. On the **SAML SSO Profiles** tab, do one of the following:
   - To create a new SAML SSO profile, click **Add.**
   - To modify an existing SAML SSO profile, select the profile, and then click **OpenEdit**.
4. In the **Create SAML SSO Profiles** or the **Configure SAML SSO Profiles** dialog box, set the following parameters:
   - Name*
   - Signing Certificate Name*
   - ACS URL*
   - Relay State Rule*
   - Send Password
   - Issuer Name
5. Click **Create** or **OK**, and then click **Close.** The SAML SSO profile that you created appears in the Traffic Policies, Profiles, and SAML SSO Profiles pane.

Session timeout for OWA 2010

You can now force OWA 2010 connections to time out after a specified period of inactivity. OWA sends repeated keepalive requests to the server to prevent timeouts. Keeping the connections open can interfere with single sign-on.

To force OWA 2010 to time out after a specified period by using the command line interface

At the command prompt, type the following commands:

```
add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -
    forcedTimeoutVal <mins>]
<!--NeedCopy-->
```

For <actname>, substitute a name for your traffic policy. For <mins>, substitute the number of minutes after which to initiate a forced timeout. For <forcedTimeout>, substitute one of the following values:

-**START** — Starts the timer for forced timeout if a timer has not already been started. If a running timer exists, has no effect.
-**STOP** — Stops a running timer. If no running timer is found, has no effect.
-**RESET** — Restarts a running timer. If no running timer is found, starts a timer as if the START option had been used.

```
add tm trafficPolicy <polname> <rule> <actname>
<!--NeedCopy-->
```

For <polname>, substitute a name for your traffic policy. For <rule>, substitute a rule in Citrix ADC default syntax.

```
bind lb vserver <vservername>  - policyName <name> -priority <number>
<!--NeedCopy-->
```

For <vservername>, substitute the name of the authentication, authorization, and auditing traffic management virtual server. For <priority>, substitute an integer that designates the policy's priority.

Example

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -
    forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
<!--NeedCopy-->
```


Rate Limiting for Citrix Gateway

September 14, 2021

The rate limiting feature for Citrix Gateway enables you to define the maximum load for a given network entity or virtual entity on the Citrix Gateway appliance. Since the Citrix Gateway appliance consumes all the unauthenticated traffic, the appliance is often exposed to process requests at a high rate. The rate limiting feature enables you to configure the Citrix Gateway appliance to monitor the rate of traffic associated with an entity and take preventive action, in real time, based on the traffic. For more information about how rate limiting works in a Citrix ADC appliance, see Rate limiting.

Citrix ADC has the rate limiting feature that provides protection to back-end servers for unforeseen rate. Since the feature for Citrix ADC did not serve the unauthenticated traffic that Citrix Gateway handles, Citrix Gateway needed its own rate limiting functionality. This is needed to check an unforeseen rate of requests from various sources the Citrix Gateway appliance is exposed to. For example, unauthenticated/login/control requests and certain APIs exposed for end user or device validations.

Common use-cases for Rate limiting

- Limit the number of requests per second from a URL.
- Drop a connection based on cookies received in request from from a particular host if the request exceeds the rate limit.
- Limit the number of HTTP requests that arrive from the same host (with a particular subnet mask) and that have the same destination IP address.

Configure Rate Limiting for Citrix Gateway

Prerequisites

A configured authentication virtual server.

Points to note

- In the configuration steps, a sample limit identifier is configured. The same can be configured with all the supported parameters like stream selector, mode. For an exhaustive description of the rate limiting capabilities, see Rate limiting.
- The policy can also be bound to a VPN virtual server as follows. You need a configured VPN virtual server to bind the policies using the following command.

```
1  bind vpn vserver  -policy denylogin  -pri 1  -type aaa_request
```



```
<!--NeedCopy-->
```

- AAA_REQUEST is a newly introduced bindpoint for responder policies. The policies configured at this bind point are applied to all the incoming request at the specified virtual server. The policies are processed for the unauthenticated/control traffic first before any other processing.
- Binding the policy to the Citrix Gateway virtual server enables rate limiting at the AAA_REQUEST bindpoint for all the traffic consumed by Citrix Gateway including unauthenticated requests.
- Binding the policy to an authentication virtual server rate limits the unauthenticated/control requests hitting the authentication virtual server.

To configure rate limiting by using the command line interface, at the command prompt, type the following commands:

```
add limitIdentifier <limitIdentifier name> -threshold <positive_integer
    > -timeslice <positive_integer> -mode <mode type>
<!--NeedCopy-->
```

```
Example: add limitIdentifier limit_one_login -threshold 10 -timeslice
    4294967290 -mode REQUEST_RATE
<!--NeedCopy-->
```

```
add responderaction denylogin respondwith  ' "HTTP/1.1 200 OK\r\n\r\n"
     +  "Request is denied due to unusual rate" '
<!--NeedCopy-->
```

```
add responder policy denylogin 'sys.check_limit("limit_one_login")'
    denylogin
<!--NeedCopy-->
```

```
bind authentication vserver <vserver name> -policy denylogin  -pri 1  -
    type aaa_request
<!--NeedCopy-->
```

```
Example: bind authentication vserver authvserver -policy denylogin  -
    pri 1  -type aaa_request
<!--NeedCopy-->
```

Parameter description

- **limitIdentifier -** Name for a rate limit identifier. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Reserved

words must not be used. This is a mandatory argument. Maximum Length: 31

- **threshold -** A maximum number of requests that are allowed in the given timeslice when requests (mode is set as REQUEST_RATE) are tracked per timeslice. When connections (mode is set as CONNECTION) are tracked, it is the total number of connections that would be let through. Default value: 1 Minimum value: 1 Maximum Value: 4294967295
- **timeSlice -** Time interval, in milliseconds, specified in multiples of 10, during which requests are tracked to check if they cross the threshold. The argument is needed only when the mode is set to REQUEST_RATE. Default value: 1000 Minimum value: 10 Maximum Value: 4294967295
- **mode -** Defines the type of traffic to be tracked.
  - REQUEST_RATE - Tracks requests/timeslice.
  - CONNECTION - Tracks active transactions.

To configure Rate Limiting using the Citrix ADC GUI:

1. Navigate to **AppExpert > Rate Limiting > Limit Identifiers**, click **Add** and specify the relevant details as specified in the CLI section.



2. Navigate to **AppExpert>Responder>Policies**. On the **Responder Policies** page, click **Add**.
3. On the **Create Responder Policy** page, create a responder policy with a responder action which has the limit identifier.
4. To create responder action, Click **Add** next to **Action** and enter a name for responder action.
5. Select type as **Respond with** from the drop-down menu, specify the following expression, “HTTP/1.1 200 OK\r\n\r\n”+ “Request is denied due to unusual rate”, and click **Create**.



6. To create a responder policy, on **Create Responder Policy** page, enter a name for the responder policy, specify the following expression, 'sys.check_limit("limit_one_login")', and click **Create**.



7. Bind the responder policy to the authentication virtual server.
   - a. Go to **Security>AAA-Application Traffic>Virtual server**.
   - b. Select the virtual server.
   - c. Add a policy.
   - d. Choose the responder policy that you want to bind to the server, set the priority.
   - e. Choose the type as **AAA-REQUEST** and click **Continue**.



> **Note:** You can also enable rate limiting at the AAA_REQUEST bind point for the VPN virtual server.

Configuration for the common use cases for applying rate limiting to Citrix Gateway

The following are the examples of commands to configure common use cases.

- Limit the number of requests per second from a URL.

```
add stream selector ipStreamSelector http.req.url "client.ip.src"

add ns limitIdentifier ipLimitIdentifier -threshold 4 -timeslice 1000 -mode request_rate -limitType smooth -selectorName ip StreamSelector

add responder policy ipLimitResponderPolicy "http.req.url.contains(\" myasp.asp\" ) && sys.check_limit(\" ipLimitIdentifier\" )" myWebSiteRedirectAction

bind authentication virtual server authvserver -policy denylogin -pri 1 -type aaa_request
<!--NeedCopy-->
```

- Drop a connection based on cookies received in request from www.yourcompany.com if the request exceeds the rate limit.

```
add stream selector cacheStreamSelector "http.req.cookie.value(\" mycookie\" )" "client.ip.src.subnet(24)"

add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000 -selectorName reqCookieStreamSelector

```



```
add responder action sendRedirectURL redirect `"http://www.
    mycompany.com\"` + http.req.url' -bypassSafetyCheck Yes

add responder policy rateLimitCookiePolicy

 “http.req.url.contains(\www.yourcompany.com\) && sys.check_limit
    (\” myLimitIdentifier\” )” sendRedirectUrl

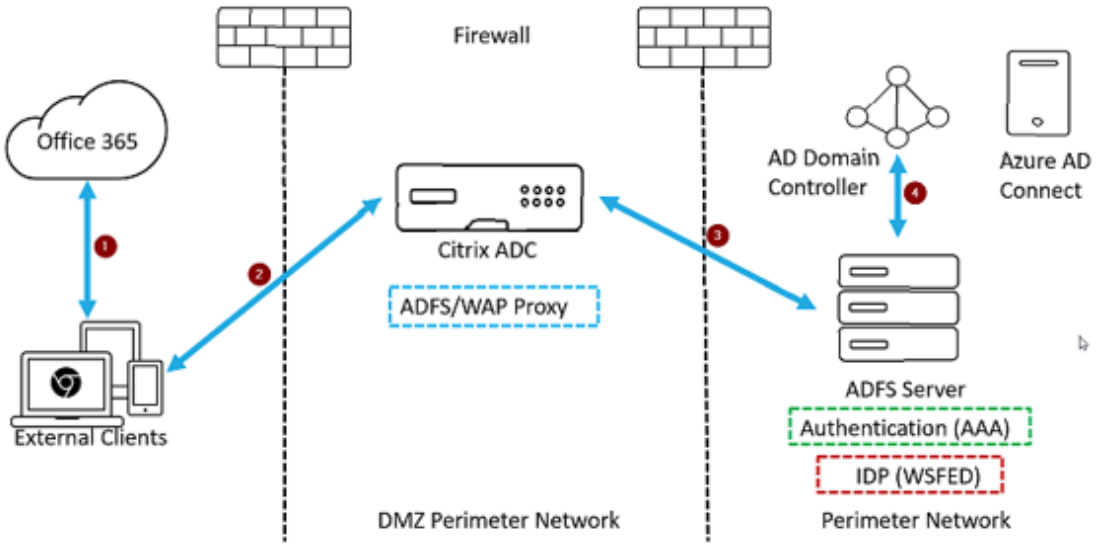
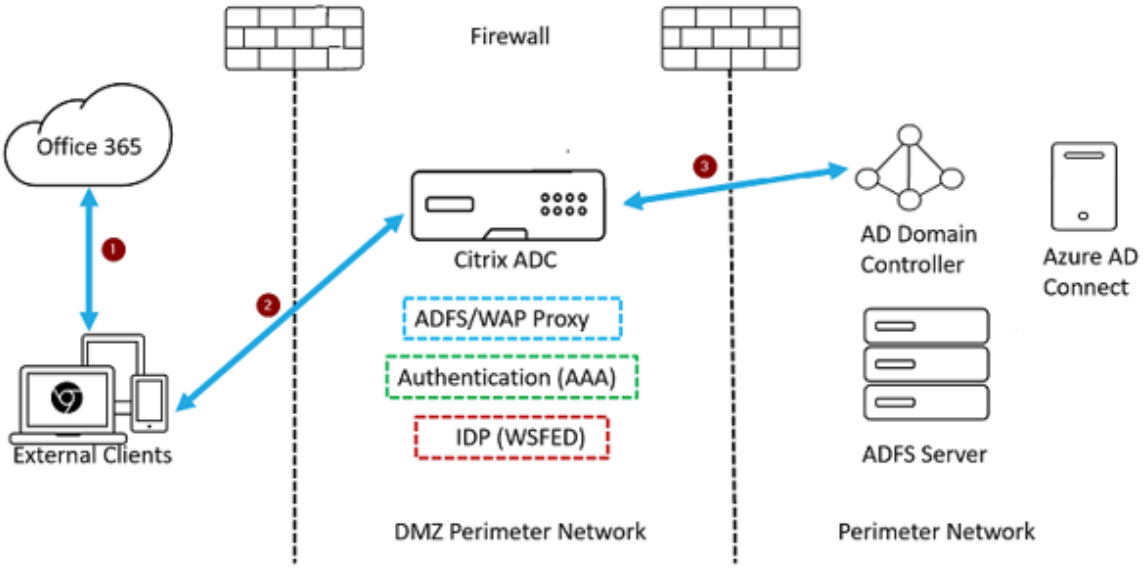
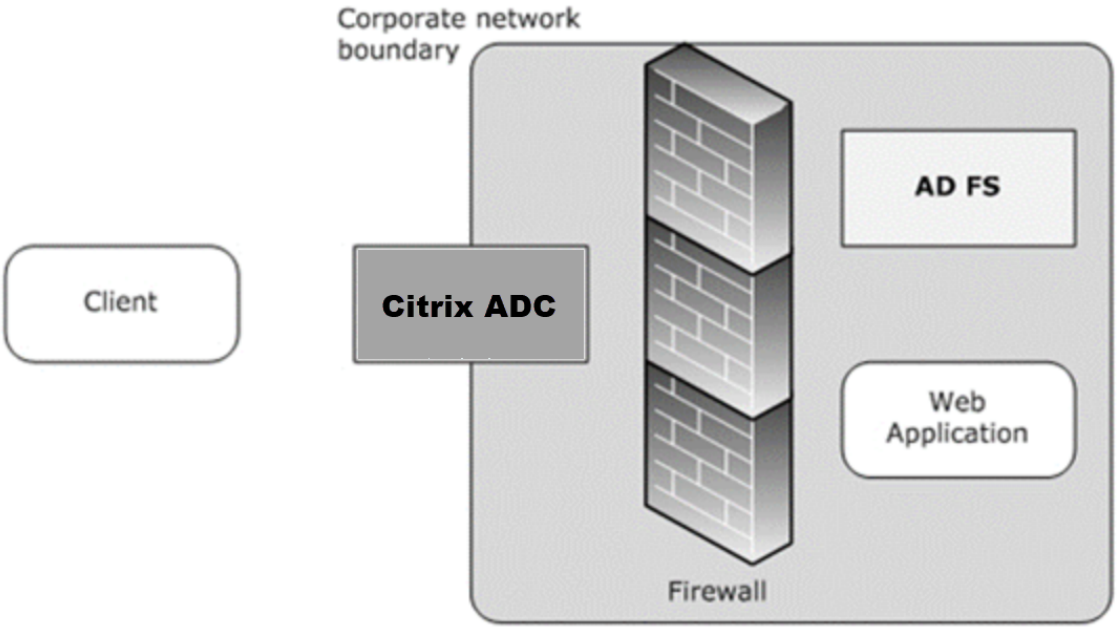
<!--NeedCopy-->
```

- Limit the number of HTTP requests that arrive from the same host (with a subnet mask of 32) and that have the same destination IP address.

```
add stream selector ipv6_sel “CLIENT.IPv6.src.subnet(32)” CLIENT
    .IPv6.dst Q.URL

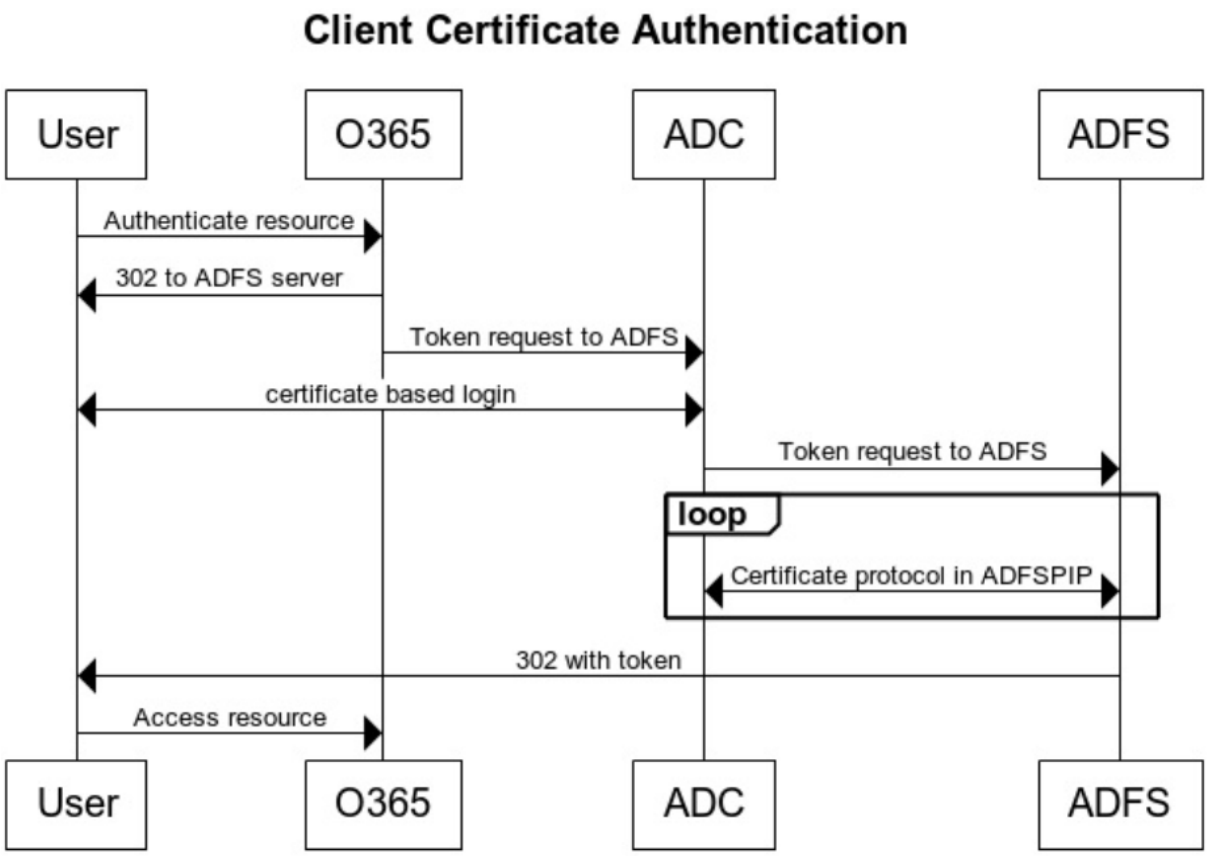
add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName
    ipv6_sel

add lb vserver ipv6_vip HTTP 3ffe:: 209 80 -persistenceType NONE
     -cltTime

add responder action redirect_page redirect “\” `http://
    redirectpage.com/\” "`

add responder policy ipv6_resp_pol “SYS.CHECK_LIMIT(\” ipv6_id\
    ” )” redirect_page

bind responder global ipv6_resp_pol 5 END -type DEFAULT
<!--NeedCopy-->
```

Authorizing user access to application resources

September 14, 2021

You can control the resources that an authenticated user can access within an application.

To do this, associate an authorization policy to each of the users, either individually or by associating the policy to a group of users. The authorization policy must specify the following:

- **Rule.** The resource to which access must be authorized. This can be specified by using basic or advanced expressions.
- **Action.** Whether access to the resource must be allowed or denied.

By default, access to all resources within an application is **DENIED** to all users. However, you can change this default authorization action to **ALLOW** access to all users (by setting the session parameters in session profile or by setting the global session parameters).

> **Warning**
>
> For optimum security, Citrix recommends that you do not to change the default authorization action from DENY to ALLOW. Instead, it is advised to create specific authorization policies for users who need access to specific resources.

To configure authorization by using the CLI

1. Configure the authorization policy.

   ns-cli-prompt> **add authorization policy <name> <rule> <action>**

2. Associate the policy with the appropriate user or group.

   - Bind the policy to a specific user.

     ns-cli-prompt> **bind aaa user <username> -policy** <policyname>

   - Bind the policy to a specific group.

     ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname>

To configure authorization by using the GUI (Configuration tab)

1. Create the authorization policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authorization**, click **Add** and then define the policy as required.

2. Associate the policy with the appropriate user or group.

   Navigate to **Security > AAA - Application Traffic > Users** or **Groups**, and edit the relevant user or group to associate it with the authorization policy.

Sample authorization configurations

Here are some example configurations to authorize user access to some application resources. Note that these are CLI commands. You can do similar configurations using the GUI, although you must not enclose the expression within quotes (“).

- `add authorization policy authzpol1 "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" ALLOW<!--NeedCopy-->`
- `bind aaa user user1 -policy authzpol1<!--NeedCopy-->`

- `add authorization policy authzpol2 "HTTP.REQ.URL.SUFFIX.EQ(\"png\")" DENY<!--NeedCopy-->`
- `bind aaa group group1 -policy authzpol2<!--NeedCopy-->`

Audit authenticated sessions

September 14, 2021

You can configure the Citrix ADC appliance to keep a log of all the events that are triggered in an authenticated session. Using this information, you can audit state and status information, to see the history for users in chronological order.

To do this, define an audit policy that specifies the following:

- **Log type.** The logs can be stored remotely (syslog) or locally on the Citrix ADC appliance (nslog).
- **Rule.** The conditions on which the logs are stored.
- **Action.** Details of the log server and other details for creating the log entries.

This audit policy can be configured at different levels: user-level, group-level, authentication, authorization, and auditing virtual server, and global system level. The policies configured at the user-level have the highest priority.

> **Note**
>
> This topic details steps for using syslog. Make necessary changes to use nslog.

To configure syslog auditing by using the CLI

1. Configure the audit server with the relevant log settings.

   ns-cli-prompt> **add audit syslogAction <name> <serverIP> ...**

2. Configure the audit policy by associating the audit server.

   ns-cli-prompt> **add audit syslogPolicy <name> <rule> <action>**

3. Associate the audit policy with one of the following entities:

   - Bind the policy to a specific user.

     ns-cli-prompt> **bind aaa user <userName>-policy** <policyname> ...

   - Bind the policy to a specific group.

     ns-cli-prompt> **bind aaa group <groupName>-policy** <policyname> ...

- Bind the policy to a authentication, authorization, and auditing virtual server.

  ns-cli-prompt> **bind authentication vserver <name> -policy** <policyname> …

- Bind the policy globally to the Citrix ADC appliance.

  ns-cli-prompt> **bind tm global -policyName** <policyname> …

To configure syslog auditing by using the GUI (Configuration tab)

1. Configure the audit server and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Auditing > Syslog**, and configure the server and the policy in the relevant tabs.

2. Associate the policy with one of the following:

   - Bind the policy to a specific user.

     Navigate to **Security > AAA - Application Traffic > Users**, and associate the authorization policy with the relevant user.

   - Bind the policy to a specific group.

     Navigate to **Security > AAA - Application Traffic > Groups**, and associate the authorization policy with the relevant group.

   - Bind the policy to a authentication, authorization, and auditing virtual server.

     Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the authorization policy with the relevant virtual server.

   - Bind the policy globally to the Citrix ADC appliance.

     Navigate to **Security > AAA - Application Traffic > Policies > Auditing > Syslog** or **Nslog**, select the authorization policy, and click **Action > Global Bindings** to bind the policy globally.

Citrix ADC as an Active Directory Federation Services proxy

September 14, 2021

Active Directory Federation Services (ADFS) is a Microsoft service that enables single sign-on (SSO) experience for Active Directory-authenticated clients to resources outside the enterprise data center. An ADFS server farm allows internal users to access external cloud-hosted services. But the moment external users are brought into the mix, the external users must be given a way to connect remotely and access cloud-based services through federated identity. Most enterprises do not prefer keeping

the ADFS server exposed in the DMZ. Therefore, ADFS proxy plays a critical role in remote user connectivity and application access.

For more than a decade, Citrix ADC appliance is playing similar roles of remote user connectivity, and application access. Citrix ADC appliance becomes the preferred solution to be used as ADFS proxy for supporting a new ADFS implementation to enable the following services:

- Secure connectivity.
- Authentication and handling of federated identity.

For more information about Citrix ADC as a SAML IdP, see Citrix ADC as a SAML IdP.

Advantages of ADFS proxy

- Reduces the footprint in DMZ to cater the need for most of the enterprises.
- Provides an SSO experience for end users.
- Supports rich methods for pre-authentication and enables multifactor authentication.
- Supports both active and passive clients.

Prerequisites for using Citrix ADC as ADFS proxy

Before you configure the Citrix ADC appliance as ADFS proxy, make sure the following prerequisites are met.

- A Citrix ADC appliance with 12.1 build or later.
- Domain ADFS server.
- Domain SSL certificate.
- Virtual IP for Content Switching virtual server.
- Enable Load Balancing, SSL Offload, Content Switching, Rewrite, and authentication, authorization, and auditing traffic management features on Citrix ADC appliance.

Configure Citrix ADC appliance as ADFS proxy

To achieve this use case, you configure Citrix ADC as ADFS proxy in DMZ zone. The ADFS server is configured along with the AD domain controller in the back-end.



1. A client request to access Microsoft Office365 gets redirected to Citrix ADC deployed as ADFS proxy.
2. User's credentials are passed to ADFS server.
3. ADFS server authenticates the credentials with on-premises AD of the domain.
4. ADFS server upon successful validation of credentials with AD, generates a token which is passed to Microsoft Office365 for session establishment.

The following are the high-level steps involved in configuring Citrix ADC appliance before you configure as ADFS proxy.

At the Citrix ADC command prompt, type the following commands:

1. Create an SSL profile for back end and enable SNI in the SSL profile. Disable SSLv3/TLS1.

   ```
   add ssl profile <new SSL profile> -sniEnable ENABLED -ssl3 DISABLED -
   tls1 DISABLED -commonName <FQDN of ADFS>
   ```

2. Disable SSLv3/TLS1 for the service.

   ```
   set ssl service <adfs service name> -sslProfile <SSL profile created in
    the above step>
   ```

3. Enable SNI extension for back-end server handshakes.

   - `set vpn parameter -backendServerSni ENABLED`
   - `set ssl parameter -denySSLReneg NONSECURE`

Configure Citrix ADC appliance as ADFS proxy using the CLI

The following sections are categorized based on the requirement to complete the configuration steps.

To configure ADFS service

1. Configure ADFS service on Citrix ADC for ADFS server.

   ```
   add service <Domain_ADFS_Service> <ADFS Server IP> SSL 443 -gslb NONE -
   maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF
   -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
   ```

   Example

   add service CTXTEST_ADFS_Service 1.1.1.1 SSL 443 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

2. Configure FQDN for content switching virtual server and enable SNI.

   ```
   set ssl service <Domain_ADFS_Service> -SNIEnable ENABLED -commonName <
   sts.domain.com>
   ```

   Example

   set ssl service CTXTEST_ADFS_Service -SNIEnable ENABLED -commonName sts.ctxtest.com

To configure ADFS load balancing virtual server

> **Important**
>
> Domain SSL certificate (SSL_CERT) is required for secure traffic.

1. Configure ADFS load balancing virtual server.

   ```
   add lb vserver <Domain_ADFS_LBVS> SSL <IP_address> -persistenceType
   NONE -cltTimeout 180
   ```

   Example

   ```
   add lb vserver CTXTEST_ADFS_LBVS SSL 192.168.1.0 -persistenceType NONE
   -cltTimeout 180
   ```

2. Bind ADFS load balancing virtual server to ADFS service.

   ```
   bind lb vserver <Domain_ADFS_LBVS> <Domain_ADFS_Service>
   ```

   Example

   ```
   bind lb vserver CTXTEST_ADFS_LBVS CTXTEST_ADFS_Service
   ```

3. Bind an SSL virtual server certificate-key pair.

   ```
   bind ssl vserver <Domain_ADFS_LBVS> -certkeyName <SSL_CERT>
   ```

   Example

   ```
   bind ssl vserver CTXTEST_ADFS_LBVS -certkeyName ctxtest_newcert_2019
   ```


To configure content switching virtual server for domain

> **Note**
>
> One free virtual IP (for example, 2.2.2.2), which is Natted to public IP is required for content switching virtual server. It must be reachable for both external and internal traffic.

1. Create a content switching virtual server with free VIP.

   ```
   add cs vserver <Domain_CSVS> SSL <FREE VIP> 443 -cltTimeout 180 -
   persistenceType NONE
   ```

   Example

   ```
   add cs vserver CTXTEST_CSVS SSL 2.2.2.2 443 -cltTimeout 180 -persistenceType
    NONE
   ```

2. Bind content switching virtual server to load balancing virtual server.

   ```
   bind cs vserver <Domain_CSVS> -lbvserver <Domain_ADFS_LBVS>
   ```

   Example

   - `bind cs vserver CTXTEST_CSVS -lbvserver CTXTEST_ADFS_LBVS`
   - `set ssl vserver CTXTEST_CSVS -sessReuse DISABLED`

3. Bind an SSL virtual server certificate-key pair.

   ```
   bind ssl vserver <Domain_CSVS> -certkeyName <SSL_CERT>
   ```

   Example

   ```
   bind ssl vserver CTXTEST_CSVS -certkeyName ctxtest_newcert_2019
   ```

Supported protocols

The Microsoft provided protocols plays a vital role in integrating with Citrix ADC appliance. Citrix ADC as ADFS proxy supports the following protocols:

- **WS-Federation**. For details, see Web Services Federation protocol.
- **ADFSPIP**. For details, see Active Directory Federation Service Proxy Integration Protocol compliance.

> **Note**
>
> Citrix ADC appliance does not support device certificate authentication when deployed as an ADFS proxy.

Web Services Federation protocol

September 14, 2021

Web Services Federation (WS-Federation) is an identity protocol that allows a Security Token Service (STS) in one trust domain to provide authentication information to an STS in another trust domain when there is a trust relationship between the two domains.

Advantages of WS-Federation

WS-Federation supports both active and passive clients whereas SAML IdP supports only passive clients.

- Active clients are Microsoft native clients such as, Outlook, and Office clients (Word, PowerPoint, Excel, and OneNote).
- Passive clients are browser based clients such as, Google Chrome, Mozilla Firefox, and Internet Explorer.

Prerequisites for using Citrix ADC as WS-Federation

Before you configure the Citrix ADC appliance as ADFS proxy, review the following:

- Active Directory.
- Domain SSL certificate.
- Citrix ADC SSL certificate and ADFS token signing certificate on ADFS server must be same.

> **Important**
>
> SAML IdP is now capable of handling WS-Federation protocol. Therefore, to configure WS-Federation IdP, you must actually configure the SAML IdP. You do not see any user interface explicitly mentioning WS-Federation.

Features supported by Citrix ADC when configured as ADFS proxy and WS-Federation IdP

The following table lists feature supported by Citrix ADC appliance when configured as ADFS proxy and WS-Federation IdP.

Features	Configure Citrix ADC appliance as ADFS proxy	Citrix ADC as WS-Federation IdP	Citrix ADC as ADFSPIP
Load Balancing	Yes	Yes	Yes
SSL Termination	Yes	Yes	Yes
Rate Limiting	Yes	Yes	Yes
Consolidation (reduces DMZ server footprint and saves public IP)	Yes	Yes	Yes
Web Application Firewall (WAF)	Yes	Yes	Yes
Authentication Offload to Citrix ADC appliance	No	Yes (Active and Passive clients)	Yes
Single sign-on (SSO)	No	Yes (Active and Passive clients)	Yes
Multi-Factor (nFactor) authentication	No	Yes (Active and Passive clients)	Yes
Azure Multi-Factor authentication	No	Yes (Active and Passive clients)	Yes
ADFS server farm can be avoided	No	Yes	Yes

Configure Citrix ADC appliance as WS-Federation IdP

Configure Citrix ADC as WS-Federation IdP (SAML IdP) in a DMZ zone. The ADFS server is configured along with the AD domain controller in the back-end.



1. The client request to Microsoft Office365 gets redirected to Citrix ADC appliance.
2. The user enters the credentials for multifactor authentication.
3. Citrix ADC validates the credentials with AD and generates a token natively on Citrix ADC appliance. The credentials are passed to Office365 for access.

> **Note**
>
> WS-Federation IdP support is done natively through Citrix ADC appliance when compared to the F5 Networks load balancer.

Configure Citrix ADC appliance as WS-Federation IdP (SAML IdP) using the CLI

The following sections are categorized based on the requirement to complete the configuration steps.

To configure LDAP authentication and add policy

> **Important**
>
> For domain users, to log on to the Citrix ADC appliance by using their corporate email addresses, you must configure the following:

- Configure LDAP authentication server and policy on the Citrix ADC appliance.
- Bind it to your authentication, authorization, and auditing virtual IP address (use of an existing LDAP configuration is also supported).
- `add authentication ldapAction <Domain_LDAP_Action> -serverIP <Active Directory IP> -serverPort 636 -ldapBase "cn=Users,dc=domain,dc=com"-ldapBindDn "cn=administrator,cn=Users,dc=domain,dc=com"-ldapBindDnPassword`


```
<administrator password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
 sAMAccountName -groupAttrName memberOf -subAttributeName cn -secType
SSL -ssoNameAttribute UserPrincipalName -followReferrals ON -Attribute1
 mail -Attribute2 objectGUID
```

- `add authentication Policy <Domain_LDAP_Policy> -rule true -action <Domain_LDAP_Action>`

Example

- `add authentication ldapAction CTXTEST_LDAP_Action -serverIP 3.3.3.3 -serverPort 636 -ldapBase "cn=Users,dc=ctxtest,dc=com"-ldapBindDn "cn=administrator,cn=Users,dc=ctxtest,dc=com"-ldapBindDnPassword xxxxxxxxxx -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberOf -subAttributeName cn -secType SSL -ssoNameAttribute UserPrincipalName -followReferrals ON -Attribute1 mail -Attribute2 objectGUID`
- `add authentication Policy CTXTEST_LDAP_Policy -rule true -action CTXTEST_LDAP_Action`

To configure Citrix ADC as WS-Federation IdP or SAML IdP

Create WS-Federation IdP (SAML IdP) action and policy for token generation. Bind it to the authentication, authorization, and auditing virtual server in later stage.

- `add authentication samlIdPProfile <Domain_SAMLIDP_Profile> -samlIdPCertName <SSL_CERT> -assertionConsumerServiceURL "https://login.microsoftonline.com/login.srf"-samlIssuerName <Issuer Name for Office 365 in ADFS Server> -rejectUnsignedRequests OFF -audience urn:federation:MicrosoftOnline -NameIDFormat persistent -NameIDExpr "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE"-Attribute1 IDPEmail -Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"`
- `add authentication samlIdPPolicy <Domain_SAMLIDP_Policy> -rule "HTTP.REQ.HEADER(\"referer\").CONTAINS(\"microsoft\")|| true"-action <Domain_SAMLIDP_Profile>`

Example

- `add authentication samlIdPProfile CTXTEST_SAMLIDP_Profile -samlIdPCertName ctxtest_newcert_2019 -assertionConsumerServiceURL "https://login.microsoftonline.com/login.srf"-samlIssuerName "http://ctxtest.com/adfs/services/trust/"-rejectUnsignedRequests OFF -audience urn:federation:MicrosoftOnline -NameIDFormat persistent -NameIDExpr "HTTP.REQ.USER.`


```
ATTRIBUTE(2).B64ENCODE"-Attribute1 IDPEmail -Attribute1Expr "HTTP.REQ.
USER.ATTRIBUTE(1)"
```

- `add authentication samlIdPPolicy CTXTEST_SAMLIDP_Policy -rule "HTTP.REQ .HEADER(\"referer\").CONTAINS(\"microsoft\")|| true"-action CTXTEST_SAMLIDP_Profi`

To configure authentication, authorization, and auditing virtual server to authenticate the employees who log on to Office365 using corporate credentials

`add authentication vserver <Domain_AAA_VS> SSL <IP_address>`

Example

- `add authentication vserver CTXTEST_AAA_VS SSL 192.168.1.0`
- `bind authentication vserver CTXTEST_AAA_VS -portaltheme RfWebUI`

To bind authentication virtual server and policy

- `bind authentication vserver <Domain_AAA_VS> -policy <Domain_SAMLIDP_Policy > -priority 100 -gotoPriorityExpression NEXT`
- `bind authentication vserver <Domain_AAA_VS> -policy <Domain_LDAP_Policy > -priority 100 -gotoPriorityExpression NEXT`

Example

- `bind authentication vserver CTXTEST_AAA_VS -policy CTXTEST_SAMLIDP_Policy -priority 100 -gotoPriorityExpression NEXT`
- `bind authentication vserver CTXTEST_AAA_VS -policy CTXTEST_LDAP_Policy -priority 100 -gotoPriorityExpression NEXT`
- `bind ssl vserver CTXTEST_AAA_VS -certkeyName ctxtest_newcert_2019`

To configure content switching

- `add cs action <Domain_CS_Action> -targetVserver <Domain_AAA_VS>`
- `add cs policy <Domain_CS_Policy> -rule "is_vpn_url || http.req.url. contains(\"/adfs/ls\")|| http.req.url.contains(\"/adfs/services/trust \")|| -action <Domain_CS_Action>`

Example

- `add cs action CTXTEST_CS_Action -targetVserver CTXTEST_AAA_VS`
- `add cs policy CTXTEST_CS_Policy -rule "is_vpn_url || http.req.url. contains(\"/adfs/ls\")|| http.req.url.contains(\"/adfs/services/trust \")|| -action CTXTEST_CS_Action`

To bind content switching virtual server to policy

```
bind cs vserver CTXTEST_CSVS -policyName CTXTEST_CS_Policy -priority 100
```

Active Directory Federation Service Proxy Integration Protocol compliance

September 14, 2021

If third party proxies are to be used in place of the Web Application Proxy, they must support the MS-ADFSPIP protocol which specifies the ADFS and WAP integration rules. ADFSPIP integrates Active Directory Federation Services with an authentication and application proxy to enable access to services located inside the boundaries of the corporate network for clients that are located outside of that boundary.



Prerequisites

To successfully establish Trust between the proxy server and the ADFS farm, review the following configuration in the Citrix ADC appliance:

- Create a SSL profile for back end and enable SNI in the SSL profile. Disable SSLv3/TLS1. At the command prompt, type the following command:


```
add ssl profile <new SSL profile> -sniEnable ENABLED -ssl3
   DISABLED -tls1 DISABLED -commonName <FQDN of ADFS>
<!--NeedCopy-->
```

- Disable SSLv3/TLS1 for the service. At the command prompt, type the following command:

```
set ssl service <adfs service name> -sslProfile
   ns_default_ssl_profile_backend
<!--NeedCopy-->
```

- Enable SNI extension for back-end server handshakes. At the command prompt, type the following command:

```
set vpn parameter  - backendServerSni ENABLED

set ssl parameter -denySSLReneg NONSECURE
<!--NeedCopy-->
```

> **Important**
>
> For Home Realm Discovery (HRD) scenarios where authentication must be offloaded to the ADFS server, Citrix recommends you disable both authentication and SSO on the Citrix ADC appliance.

Authentication mechanism

The following are the high-level flow of events for the authentication.

1. **Establish Trust with ADFS server** – Citrix ADC server establishes Trust with the ADFS server by registering a client certificate. Once the Trust is established, the Citrix ADC appliance re-establishes the trust after reboot without user intervention.

   Upon certificate expiry, you must reestablish the trust by removing and adding ADFS proxy profile again.

2. **Published endpoints** - The Citrix ADC appliance automatically fetches the list of published endpoints on the ADFS server post trust establishment. These published endpoints filter the requests forwarded to the ADFS server.

3. **Insert headers to client requests** – When the Citrix ADC appliance tunnels client requests, the HTTP headers related to ADFSPIP are added in the packet while sending them to ADFS server. You can implement access control at the ADFS server based on these header values. The following headers are supported.

   - X-MS-Proxy
   - X-MS-Endpoint-Absolute-Path

- X-MS-Forwarded-Client-IP
- X-MS-Proxy
- X-MS-Target-Role
- X-MS-ADFS-Proxy-Client-IP

4. **Manage end-user traffic** – End-user traffic is routed securely to the desired resources.

> **Note**
>
> Citrix ADC appliance uses form based authentication.

Configure Citrix ADC to work with ADFS server

Prerequisites

- Configure Context Switching (CS) server as front-end with authentication, authorization, and auditing server behind CS. At the command prompt, type:

```
add cs vserver <cs vserver name> SSL 10.220.xxx.xx 443
-cltTimeout 180 -AuthenticationHost <adfs server hostname> -
   Authentication OFF -persistenceType NONE
<!--NeedCopy-->
```

```
add cs action <action name1> -targetLBVserver <lb vserver name>
<!--NeedCopy-->
```

```
add cs action <action name2> -targetLBVserver <lb vserver name>
<!--NeedCopy-->
```

```
add cs policy <policy name1> -rule " http.req.url.contains(\"/
   adfs/services/trust\") || http.req.url.contains(\"
   federationmetadata/2007-06/federationmetadata.xml\")" -action
   <action name1>
<!--NeedCopy-->
```

```
add cs policy <policy name2> -rule "HTTP.REQ.URL.CONTAINS(\"/adfs
   /ls\")" -action <action name2>
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -policyName <policy name1> -
   priority 100
<!--NeedCopy-->
```



```
bind cs vserver <cs vserver name> -policyName <policy name2> -
   priority 110
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -lbvserver <lb vserver name>
<!--NeedCopy-->
```

- Add ADFS service. At the command prompt, type:

```
add service <adfs service name> <adfs server ip> SSL 443
<!--NeedCopy-->
```

```
set ssl service <adfs service name> -sslProfile
   ns_default_ssl_profile_backend
<!--NeedCopy-->
```

- Add a load balanced virtual server. At the command prompt, type:

```
add lb vserver <lb vserver name> SSL 0.0.0.0 0
<!--NeedCopy-->
```

```
set ssl vserver <lb vserver name> -sslProfile
   ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

- Bind service to the load balanced server. At the command prompt, type:

```
bind lb vserver <lb vserver name> <adfs service name>
<!--NeedCopy-->
```

To configure Citrix ADC to work with ADFS server you need to do the following:

1. Create an SSL CertKey profile key to use with ADFS proxy profile
2. Create an ADFS proxy profile
3. Associate the ADFS proxy profile to the LB virtual server

Create an SSL certificate with private key to use with ADFS proxy profile

At the command prompt, type:

```
    add ssl certkey <certkeyname>  - cert <certificate path> -key <
       keypath>
<!--NeedCopy-->
```


Note: The Certificate file and the key file must be present in the Citrix ADC appliance.
Create an ADFS proxy profile using CLI

At the command prompt, type:

```
add authentication adfsProxyProfile  <profile name> -serverUrl <https://<server FQDN or IP address>/> -username <adfs admin user name> -password <password for admin user> -certKeyName <name of the CertKey profile created above>
<!--NeedCopy-->
```

Where;

Profile name – Name of the AFDS proxy profile to be created

ServerUrl – Fully qualified domain name of the ADFS service including protocol and port. For example, `https://adfs.citrix.com`

Username – User name of an admin account that exists on ADFS server

Password – Password of the admin account used as user name

certKeyName – Name of the previously created SSL CertKey profile

Associate the ADFS proxy profile to the load balancing virtual server using CLI

In the ADFS deployment, there are two load balancing virtual severs, one for the client traffic and the other one for metadata exchange. The ADFS proxy profile must be associated with the load balancing virtual server that is front-ending the ADFS server.

At the command prompt, type:

```
set lb vserver <adfs-proxy-lb> -adfsProxyProfile <name of the ADFS proxy profile>
<!--NeedCopy-->
```

Trust renewal support for ADFSPIP

You can renew the trust of the existing certificates that are nearing to expiry or if the existing certificate is not valid. The trust renewal of certificates is done only when the trust is established between Citrix ADC appliance and ADFS server. To renew the trust of the certificate, you must provide the new certificate.

> **Important**
>
> Manual intervention is required for trust renewal of new certificates.

The following example lists the steps involved in the certificate trust renewal:

1. The Citrix ADC appliance sends both old (SerializedTrustCertificate) and new (SerializedReplacementCertificate) certificates in POST request to ADFS server for trust renewal.
2. The ADFS server responds with 200 OK success if trust is renewed successfully.
3. The Citrix ADC appliance updates the state as “ESTABLISHED_RENEW_SUCCESS” if the trust renewal is successful. If the trust renewal fails, the state is updated as “ESTABLISHED_RENEW_FAILED” and Citrix ADC appliance keeps using the old certificate.

> **Note**
>
> You cannot update the certkey if it is already bound to some ADFS proxy profile.

To configure the trust renewal of certificates by using the CLI

At the command prompt, type:

```
set authentication adfsProxyProfile <name> [-CertKeyName <string>]
<!--NeedCopy-->
```

Example:

```
set authentication adfsProxyProfile adfs_2 -CertKeyName ca_cert1
<!--NeedCopy-->
```

Client certificate based authentication on ADFS server

Starting with Windows server 2016, Microsoft introduced a new way of authenticating users when ADFS is accessed through proxy servers. Now, end users can log in with their certificates thereby avoiding the use of password.

End users often access ADFS through a proxy, especially when they are not in the premises. Therefore, ADFS proxy servers are required to support client certificate authentication through ADFSPIP protocol.

When ADFS is load balanced using a Citrix ADC appliance, to support certificate based authentication at the ADFS server, users need to login to the Citrix ADC appliance using the certificate as well. This allows Citrix ADC to pass user certificate to ADFS to provide SSO to the ADFS server.

The following diagram depicts the client certificate authentication flow.



Configure SSO for ADFS server using client certificate

To configure SSO for ADFS server using client certificate, you must first configure the client certificate authentication on the Citrix ADC appliance. You must then bind the certificate authentication policy to authentication, authorization, and auditing virtual server.

At the command prompt, type;

```
add authentication certAction <action name>

add authentication Policy <policy name> -rule <expression> -action <
    action name>

add authentication policylable <label Name>

bind authentication policylabel <label Name> -policyName <name of the
    policy> -priority<integer>

<!--NeedCopy-->
```

Example:


```
add authentication certAction adfsproxy-cert

add authentication Policy cert1 -rule TRUE -action adfsproxy-cert

add authentication policylable certfactor

bind authentication policylabel certfactor  - policyName cert1  -
   priority 100

<!--NeedCopy-->
```

For information on configuring client certificate on Citrix ADC appliance, see Configure client certificate authentication using advanced policies.

Use an on-premises Citrix Gateway as the identity provider for Citrix Cloud

September 14, 2021

Citrix Cloud supports using an on-premises Citrix Gateway as an identity provider to authenticate subscribers signing in to their workspaces.

By using Citrix Gateway authentication, you can:

- Continue authenticating users through your existing Citrix Gateway so they can access the resources in your on-premises Virtual Apps and Desktops deployment through Citrix Workspace.
- Use the Citrix Gateway authentication, authorization, and auditing functions with Citrix Workspace.
- Use features such as pass-through authentication, smart cards, secure tokens, conditional access policies, federation, and many others while providing your users access to the resources they need through Citrix Workspace.

Citrix Gateway authentication is supported for use with the following product versions:

- Citrix Gateway 13.0 41.20 Advanced edition or later
- Citrix Gateway 12.1 54.13 Advanced edition or later

Prerequisites

- Cloud Connectors - You need at least two servers on which to install the Citrix Cloud Connector software.

- Active Directory - Perform the necessary checks.
- Citrix Gateway requirements
  - Use advanced policies on the on-premises gateway due to deprecation of classic policies.
  - When configuring the Gateway for authenticating subscribers to Citrix Workspace, the gateway acts as an OpenID Connect provider. Messages between Citrix Cloud and Gateway conform to the OIDC protocol, which involves digitally signing tokens. Therefore, you must configure a certificate for signing these tokens.
  - Clock synchronization - The Gateway must be synchronized to NTP time.

For details, see Prerequisites.

Create an OAuth IdP policy on the on-premises Citrix Gateway

> **Important:**
>
> You must have generated the client ID, secret, and redirect URL in the **Citrix Cloud > Identity and Access Management > Authentication** tab. For details, see Connect an on-premises Citrix Gateway to Citrix Cloud.

Creating an OAuth IdP authentication policy involves the following tasks:

1. Create an OAuth IdP profile.
2. Add an OAuth IdP policy.
3. Bind the OAuth IdP policy to an authentication virtual server.
4. Bind the certificate globally.

Creating an OAuth IdP profile by using the CLI

At the command prompt, type;

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string> [-undefAction <string>] [-comment <string>][-logAction <
    string>]

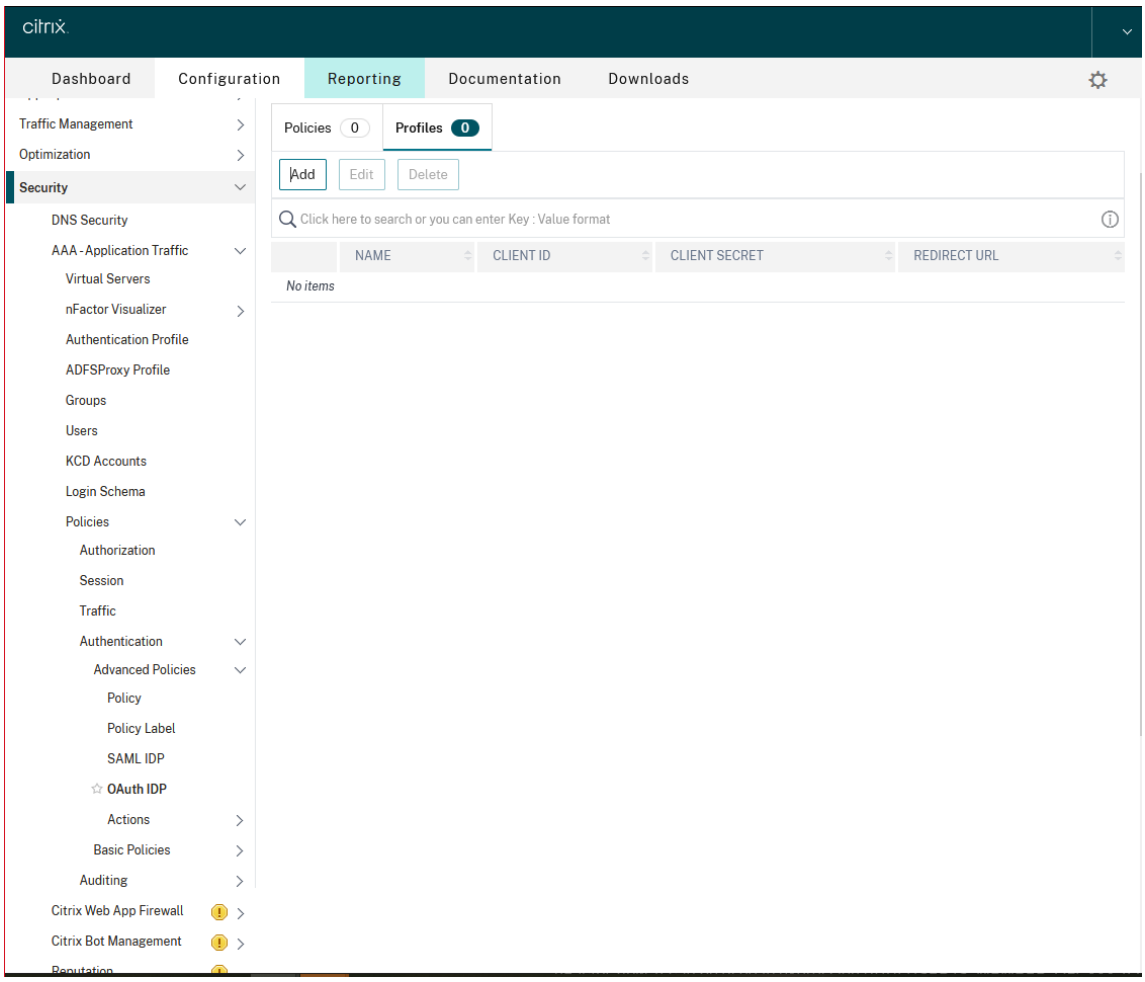
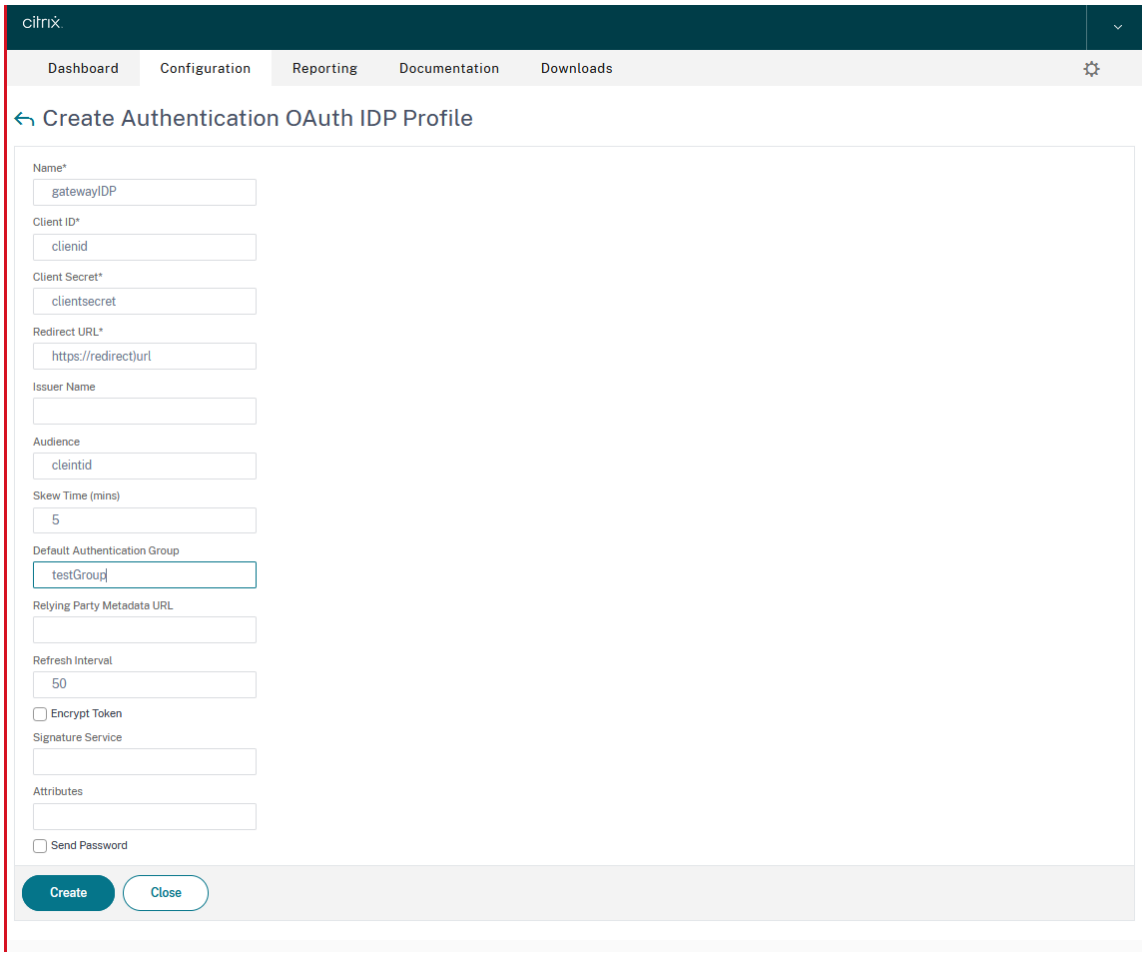
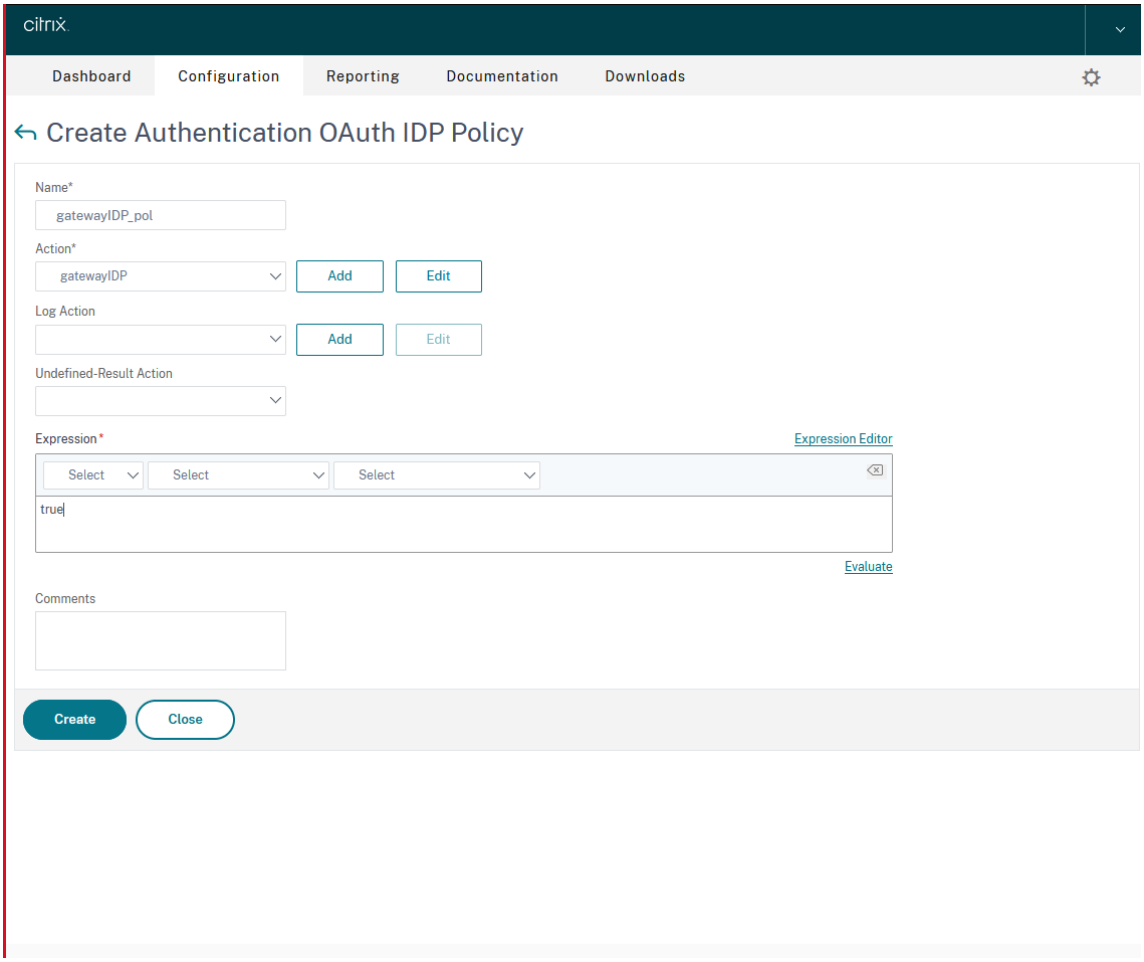
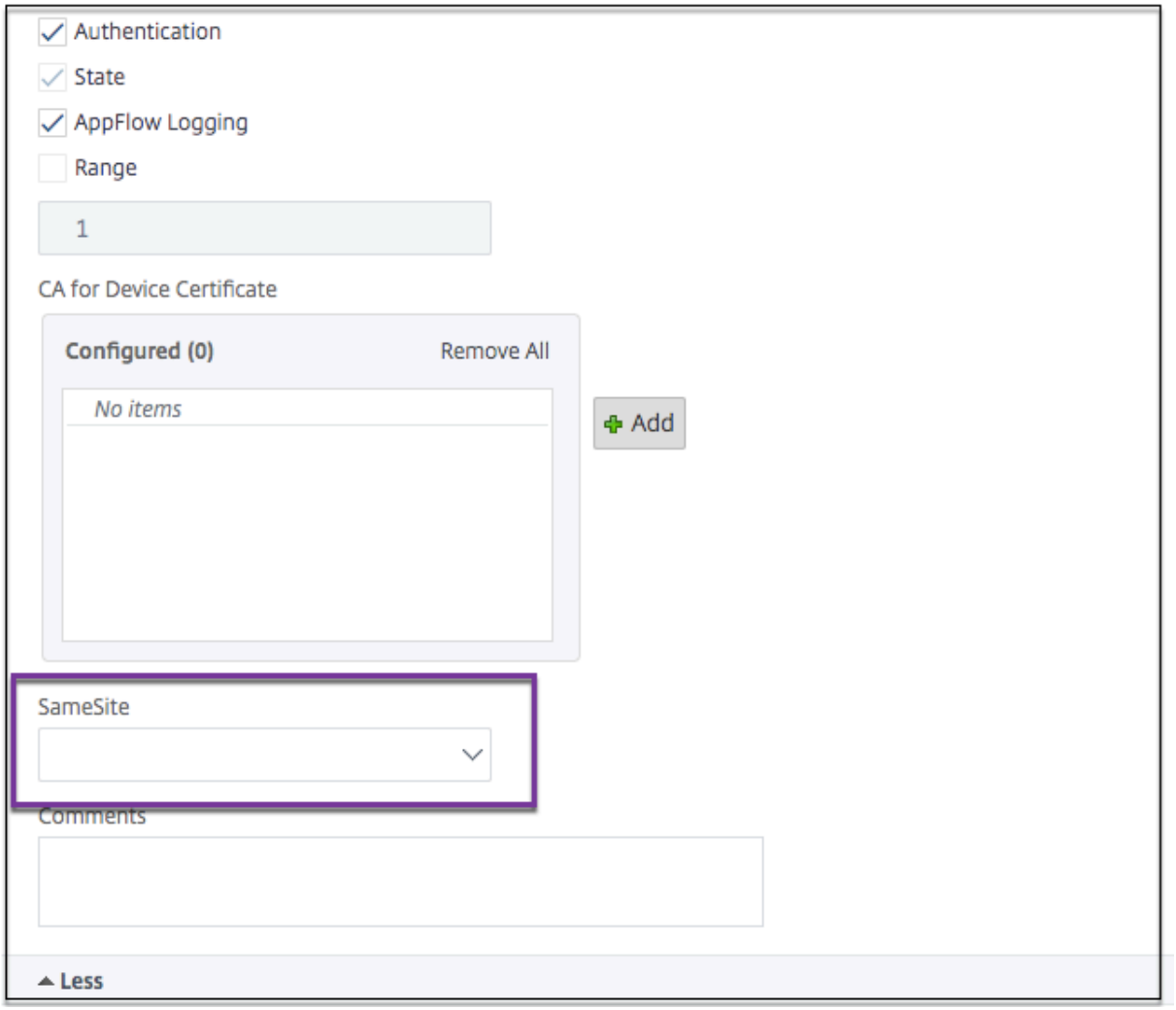
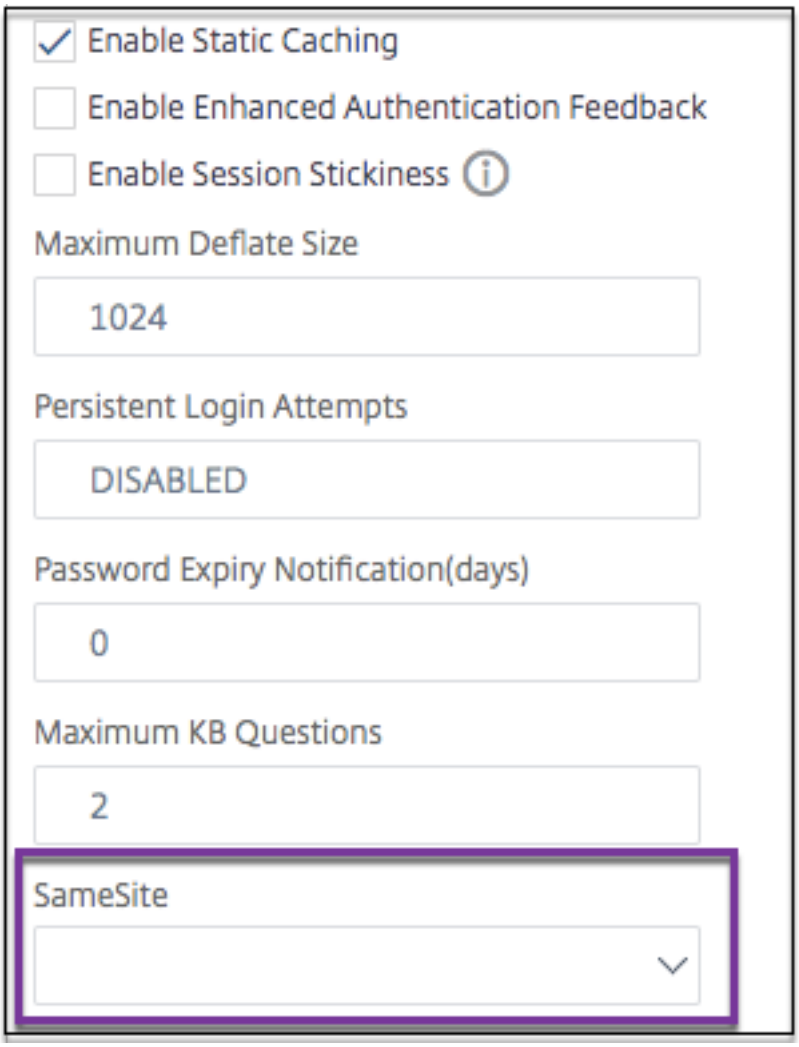
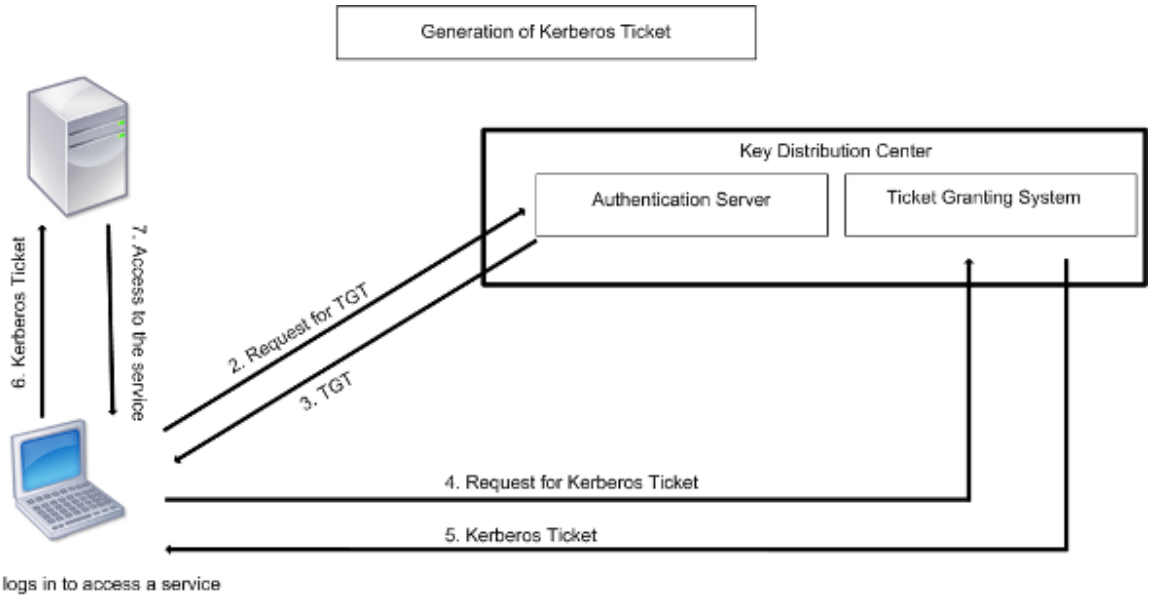
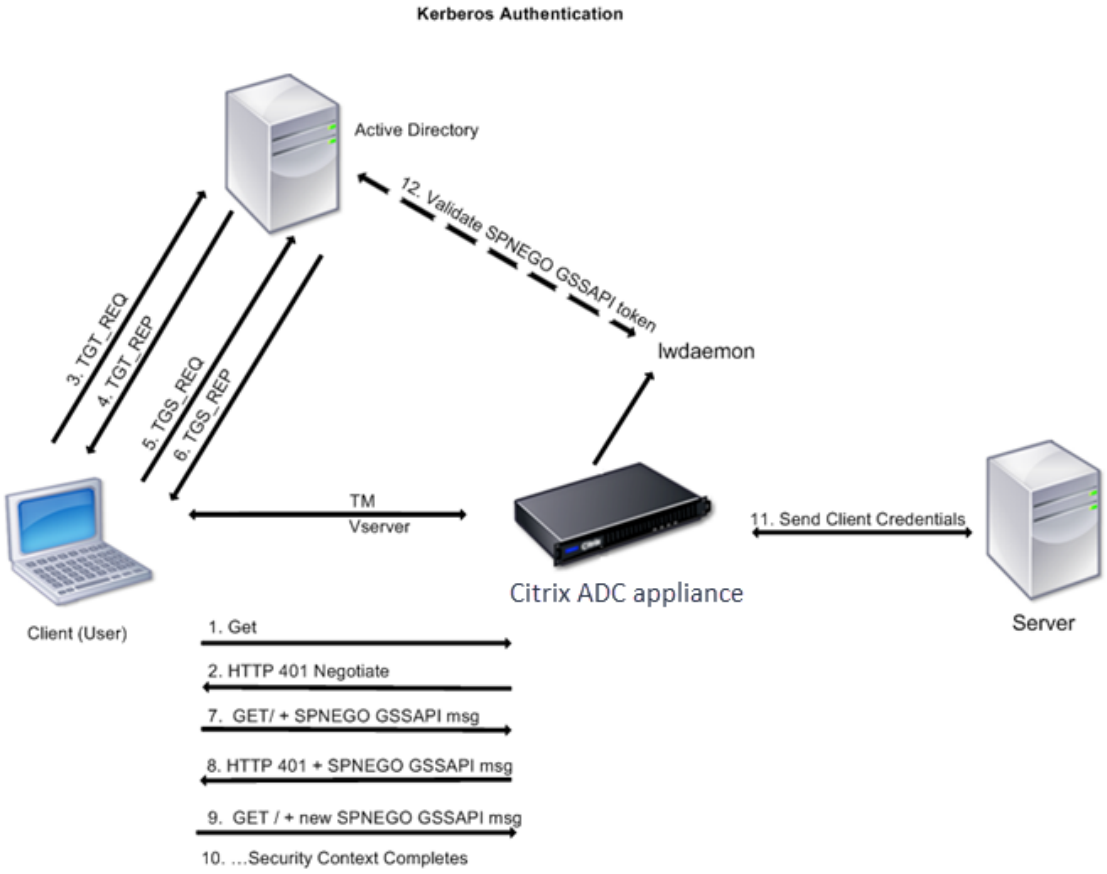
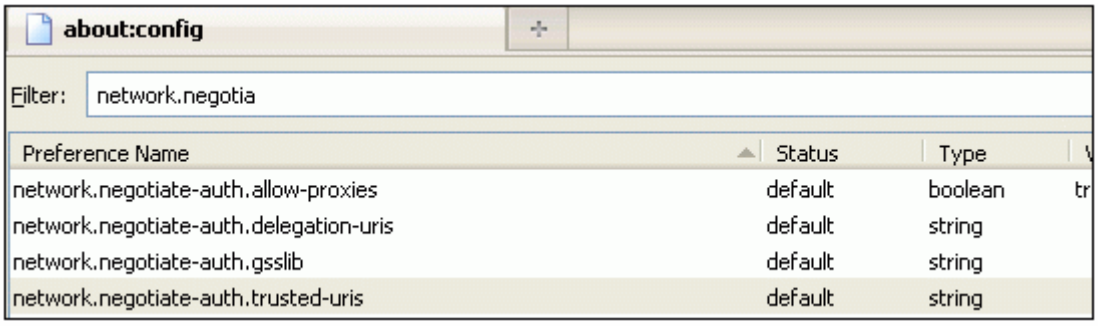
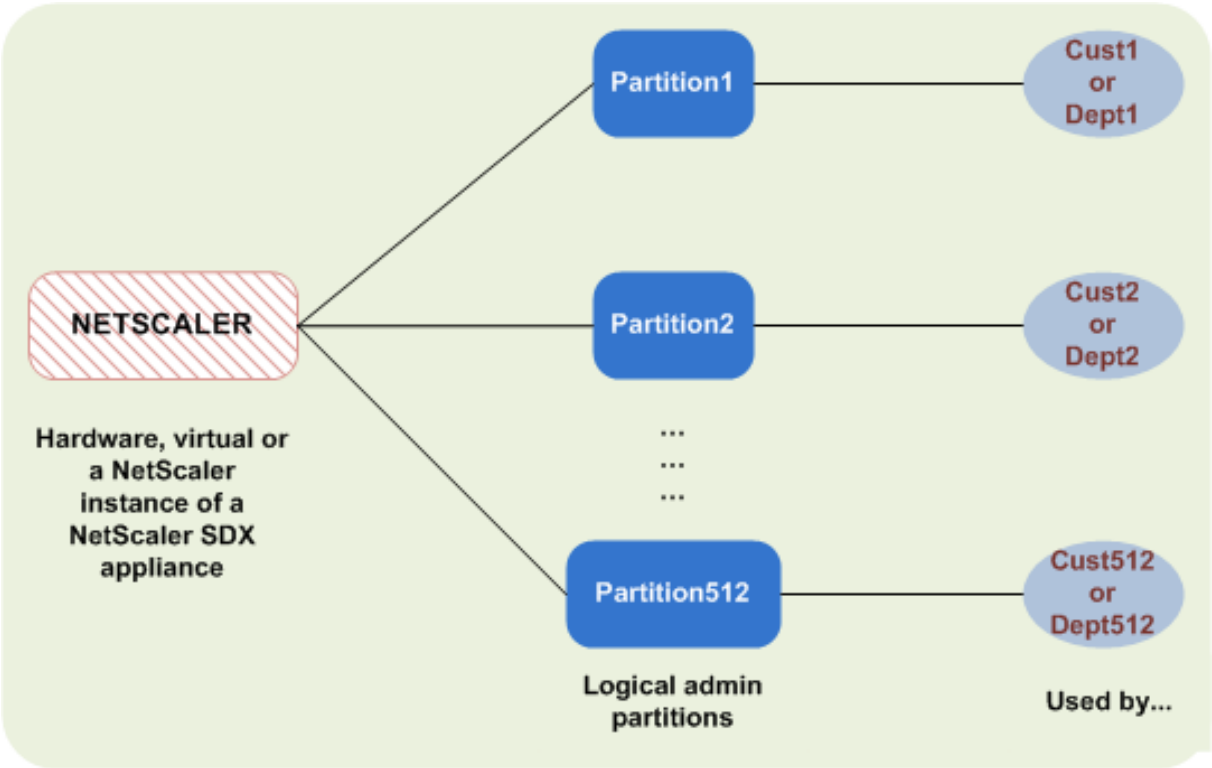
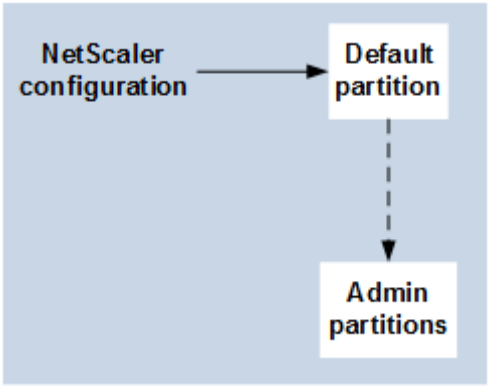
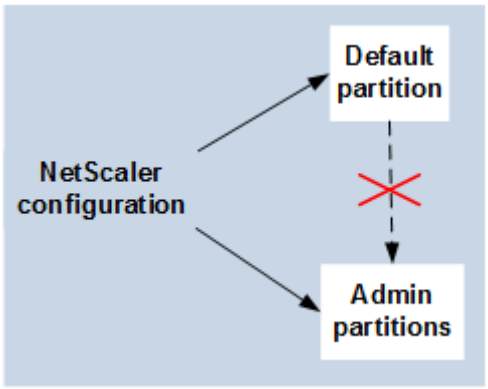
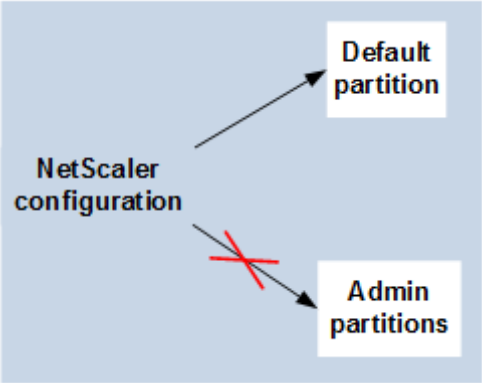
add authentication ldapAction <name> -serverIP <IP> -ldapBase "dc=aaa,
    dc=local"

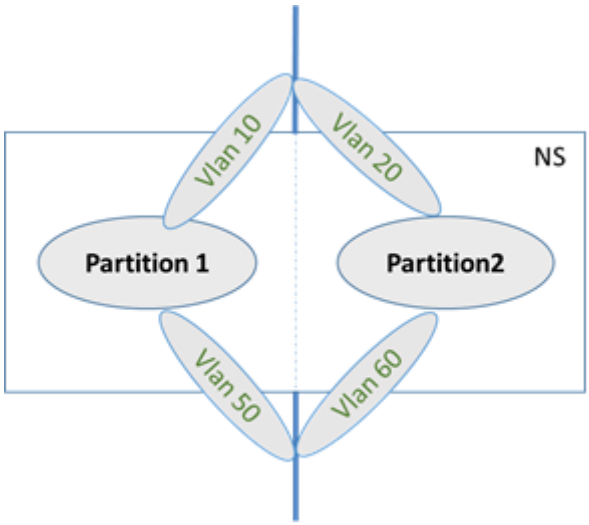
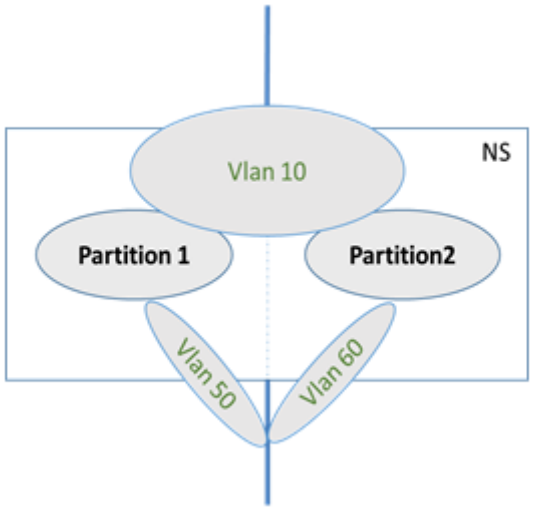
```



```
ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName

add authentication policy <name> -rule <expression> -action <string>

bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority <integer> -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority <integer> -gotoPriorityExpression END

bind vpn global  - certkey <>
<!--NeedCopy-->
```

Creating an OAuth IdP profile by using the GUI

1. Navigate to **Security > AAA – Application Traffic > Policies > Authentication > Advanced Policies > OAuth IDP**.



2. In the **OAuth IDP** page, select the **Profiles** tab and click **Add**.
3. Configure the OAuth IdP profile.

   > **Note:**
   >
   > - Copy and paste the client ID, secret, and Redirect URL values from the **Citrix Cloud > Identity and Access Management > Authentication** tab to establish the connection to Citrix Cloud.
   > - Enter the Gateway URL correctly in the **Issuer Name** Example: https://GatewayFQDN.com
   > - Also copy and paste the client ID in the **Audience** field as well.
   > - **Send Password**: Enable this option for single sign-on support. This option is disabled by default.

4. On the **Create Authentication OAuth IDP Profile** screen, set values for the following parameters and click **Create**.

   - **Name** – Name of the authentication profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created.
   - **Client ID** – Unique string that identifies SP. Authorization server infers client configuration using this ID. Maximum Length: 127.
   - **Client Secret** – Secret string established by user and authorization server. Maximum Length: 239.
   - **Redirect URL** – Endpoint on SP to which code/token has to be posted.
   - **Issuer Name** – Identity of the server whose tokens are to be accepted. Maximum Length: 127. Example: https://GatewayFQDN.com
   - **Audience** – Target recipient for the token being sent by IdP. This might be checked by the recipient.
   - **Skew Time** – This option specifies the allowed clock skew in minutes that Citrix ADC allows on an incoming token. For example, if skewTime is 10, then the token would be valid from (current time - 10) min to (current time + 10) min, that is 20 min in all. Default value: 5.
   - **Default Authentication Group** – A group added to the session internal group list when this profile is chosen by IdP which can be used in nFactor flow. It can be used in the expression (AAA.USER.IS_MEMBER_OF(“xxx”)) for authentication policies to identify relying party related nFactor flow. Maximum Length: 63

A group is added to the session for this profile to simplify policy evaluation and help in customizing policies. This is the default group that is chosen when the authentication succeeds in addition to the extracted groups. Maximum Length: 63.



5. Click **Policies** and click **Add.**
6. On the **Create Authentication OAuth IDP Policy** screen, set values for the following parameters and click **Create**.
   - **Name** – The name of the authentication policy.
   - **Action** – Name of profile created earlier.
   - **Log Action** – Name of the message log action to use when a request matches this policy. Not a mandatory filed.
   - **Undefined-Result Action** – Action to perform if the result of policy evaluation is undefined(UNDEF). Not a mandatory field.
   - **Expression** – Default syntax expression that the policy uses to respond to specific request. For example, true.
   - **Comments** – Any comments about the policy.



> **Note:**
>
> When **sendPassword** is set to ON (OFF by default), user credentials are encrypted and passed through a secure channel to Citrix Cloud. Passing user credentials through a secure channel allows you to enable SSO to Citrix Virtual Apps and Desktops upon launch.

Binding the OAuthIDP policy and LDAP policy to the authentication virtual server

1. Navigate to **Configuration > Security > AAA-Application Traffic > Policies >Authentication > Advanced Policies > Actions > LDAP**.
2. On **LDAP Actions** screen, click **Add**.
3. On the **Create Authentication LDAP Server** screen, set the values for the following parameters, and click **Create**.
   - **Name –** The name of the LDAP action
   - **ServerName/ServerIP –** Provide FQDN or IP of the LDAP server
   - Choose appropriate values **for Security Type, Port, Server Type, Time-Out**
   - Make sure **Authentication** is checked

- **Base DN –** Base from which to start LDAP search. For example, `dc=aaa,dc=local`.
- **Administrator Bind DN:** User name of the bind to LDAP server. For example, admin@aaa.local.
- **Administrator Password/Confirm Password: Password to bind LDAP**
- Click **Test Connection** to test your settings.
- **Server Logon Name Attribute:** Choose **"sAMAccountName"**
- Other fields are not mandatory and hence can be configured as required.

4. Navigate to **Configuration > Security > AAA-Application Traffic > Policies >Authentication > Advanced Policies > Policy.**
5. On the **Authentication Policies** screen, click **Add**.
6. On the **Create Authentication Policy** page, set the values for the following parameters, and click **Create.**

   - **Name –** Name of the LDAP Authentication Policy.
   - **Action Type –** Choose **LDAP.**
   - **Action –** Choose the LDAP action.
   - **Expression –** Default syntax expression that the policy uses to respond to specific request. For example, true**.

Support for active-active GSLB deployments on Citrix Gateway

Citrix Gateway configured as an Identity Provider (IdP) using the OIDC protocol can support active-active GSLB deployments. The active-active GSLB deployment on Citrix Gateway IdP provides the capability to load balance an incoming user login request across multiple geographic locations.

> **Important**
>
> Citrix recommends you to bind CA certificates to the SSL service and enable certificate validation on the SSL service for enhanced security.

For more information on configuring GSLB setup, see Example of a GSLB setup and configuration.

Configuration support for SameSite cookie attribute

September 14, 2021

The SameSite attribute indicates the browser whether the cookie can be used for cross-site context or only for same-site context. Also, if an application intends to be accessed in cross-site context then it can do so only via HTTPS connection. For details, see RFC6265.

Until Feb 2020, the SameSite attribute was not explicitly set in Citrix ADC. The browser took the default value (None). The non-setting of SameSite attribute did not impact the Citrix Gateway and Citrix ADC AAA deployments.

With certain browsers upgrade, such as Google Chrome 80, there is a change in the default cross-domain behavior of cookies. The SameSite attribute can be set to one of the following values. Default value for Google Chrome is set to Lax. For certain version of other browsers, the default value for SameSite attribute might still be set to None.

- **None**: Indicates the browser to use cookie in cross-site context only on secure connections.
- **Lax**: Indicates the browser to use cookie for requests on the same-site context. In cross-site context, only safe HTTP methods like GET request can use the cookie.
- **Strict**: Use the cookie only in the same site context.

If there is no SameSite attribute in the cookie, the Google Chrome assumes the functionality of SameSite = Lax.
As a result, for deployments within an iframe with cross-site context that require cookies to be inserted by the browser, Google Chrome does not share cross site cookies. As a result, the iframe within the website might not load.

Configure SameSite cookie attribute

A new cookie attribute named SameSite is added to the VPN and Citrix ADC AAA virtual servers. This attribute can be set at the global level and at the virtual server level.

To configure SameSite attribute, you must perform the following:

1. Set the SameSite attribute for the virtual server
2. Bind cookies to the patset (if the browser drops cross-site cookies)

Setting the SameSite attribute by using the CLI

To set the SameSite attribute at the virtual server level, use the following commands.

```
set vpn vserver VP1 -SameSite [STRICT | LAX | None]
set authentication vserver AV1 -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

To set the SameSite attribute at the global level, use the following commands.

```
set aaa parameter -SameSite [STRICT | LAX | None]
set vpn parameter -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```


Note: The virtual server level setting takes preference over the global level setting. Citrix recommends setting the SameSite cookie attribute at the virtual server level.

Binding cookies to the patset by using the CLI

If the browser drops cross-site cookies, you can bind that cookie string to the existing ns_cookies_SameSite patset so that the SameSite attribute is added to the cookie.

Example:

```
bind patset ns_cookies_SameSite "NSC_TASS"
bind patset ns_cookies_SameSite "NSC_TMAS"
<!--NeedCopy-->
```

Setting the SameSite attribute by using the GUI

To set the SameSite attribute at the virtual server level:

1. Navigate to **Security > AAA – Application Traffic > Virtual Servers.**
2. Select a virtual server and click **Edit**.
3. Click the edit icon in the **Basic Setting**s section and click **More**.
4. In **SameSite**, select the option as required.



To set the SameSite attribute at the global level:

1. Navigate to **Security > AAA – Application Traffic > Change Authentication Settings.**

2. In the **Configure AAA Parameter** page, click the **SameSite** list, and select the option as required.



Authentication, authorization, and auditing configuration for commonly used protocols

September 14, 2021

Configuring the Citrix ADC appliance for authentication, authorization, and auditing needs a specific setup on the Citrix ADC appliance and clients' browsers. The configuration varies with the protocol used for authentication, authorization, and auditing.

For more information about configuring the Citrix ADC appliance for Kerberos authentication, see Handling Authentication, Authorization and Auditing with Kerberos/NTLM.

Handling authentication, authorization and auditing with Kerberos/NTLM

September 14, 2021

Kerberos, a computer network authentication protocol, provides secure communication over the Internet. Designed primarily for client-server applications, it provides for mutual authentication by which the client and server can each ensure the other's authenticity. Kerberos uses a trusted third party, referred to as Key Distribution Center (KDC). A KDC consists of an Authentication Server (AS), which authenticates a user, and a Ticket Granting Server (TGS).

Each entity on the network (client or server) has a secret key that is known only to itself and the KDC. The knowledge of this key implies authenticity of the entity. For communication between two entities on the network, the KDC generates a session key, referred to as the Kerberos ticket or service ticket. The client makes a request to the AS for credentials for a specific server. The client then receives a ticket, referred to as Ticket Granting Ticket (TGT). The client then contacts the TGS, using the TGT it received from the AS to prove its identity, and asks for a service. If the client is eligible for the service, the TGS issues a Kerberos ticket to the client. The client then contacts the server hosting the service (referred to as the service server), using the Kerberos ticket to prove that it is authorized to receive the service. The Kerberos ticket has a configurable lifetime. The client authenticates itself with the AS only once. If it contacts the physical server multiple times, it reuses the AS ticket.

The following figure shows the basic functioning of the Kerberos protocol.

Figure 1. **Functioning of Kerberos**



Kerberos authentication has the following advantages:

- Faster authentication. When a physical server gets a Kerberos ticket from a client, the server has enough information to authenticate the client directly. It does not have to contact a domain controller for client authentication, and therefore the authentication process is faster.
- Mutual authentication. When the KDC issues a Kerberos ticket to a client and the client uses the ticket to access a service, only authenticated servers can decrypt the Kerberos ticket. If the virtual server on the Citrix ADC appliance is able to decrypt the Kerberos ticket, you can conclude that both the virtual server and client are authenticated. Thus, the authentication of the server happens along with the authentication of the client.
- Single sign-on between Windows and other operating systems that support Kerberos.

Kerberos authentication may have the following disadvantages:

- Kerberos has strict time requirements; the clocks of the involved hosts must be synchronized with the Kerberos server clock to ensure that the authentication does not fail. You can mitigate this disadvantage by using the Network Time Protocol daemons to keep the host clocks synchronized. Kerberos tickets have an availability period, which you can configure.
- Kerberos needs the central server to be available continuously. When the Kerberos server is down, no one can log on. You can mitigate this risk by using multiple Kerberos servers and fallback authentication mechanisms.
- Because all the authentication is controlled by a centralized KDC, any compromise in this infrastructure, such as the user's password for a local workstation being stolen, can allow an attacker to impersonate any user. You can mitigate this risk to some extent by using only a desktop machine or laptop that you trust, or by enforcing preauthentication by means of a hardware-token.

To use Kerberos authentication, you must configure it on the Citrix ADC appliance and on each client.

Optimizing Kerberos authentication on authentication, authorization, and auditing

The Citrix ADC appliance now optimizes and improves the system performance while Kerberos authentication. The authentication, authorization, and auditing daemon remembers the outstanding Kerberos request for the same user to avoid load on Key Distribution Center (KDC), which will avoid duplicate requests.

How Citrix ADC implements Kerberos for client authentication

September 14, 2021

> **Important**
>
> Kerberos/NTLM authentication is supported only in the NetScaler 9.3 nCore release or later, and it can be used only for authentication, authorization, and auditing traffic management virtual

> servers.

Citrix ADC handles the components involved in Kerberos authentication in the following way:

Key Distribution Center (KDC)

In the Windows 2000 Server or later versions, the Domain Controller and KDC are part of the Windows Server. If the Windows Server is UP and running, it indicates that the Domain Controller and KDC are configured. The KDC is also the Active Directory server.

> **Note**
>
> All Kerberos interactions are validated with the Windows Kerberos Domain Controller.

Authentication service and protocol negotiation

A Citrix ADC appliance supports Kerberos authentication on the authentication, authorization, and auditing traffic management authentication virtual servers. If the Kerberos authentication fails, the Citrix ADC uses the NTLM authentication.

By default, Windows 2000 Server and later Windows Server versions use Kerberos for authentication, authorization, and auditing. If you create an authentication policy with NEGOTIATE as the authentication type, the Citrix ADC attempts to use the Kerberos protocol for authentication, authorization, and auditing and if the client's browser fails to receive a Kerberos ticket, the Citrix ADC uses the NTLM authentication. This process is referred to as negotiation.

The client may fail to receive a Kerberos ticket in any of the following cases:

- Kerberos is not supported on the client.
- Kerberos is not enabled on the client.
- The client is in a domain other than that of the KDC.
- The Access Directory on the KDC is not accessible to the client.

For Kerberos/NTLM authentication, the Citrix ADC does not use the data that is present locally on the Citrix ADC appliance.

Authorization

The traffic management virtual server can be a load balancing virtual server or a content switching virtual server.

Auditing

The Citrix ADC appliance supports auditing of Kerberos authentication with the following audit logging:

- Complete audit trail of the traffic management end-user activity
- SYSLOG and high performance TCP logging
- Complete audit trail of system administrators
- All system events
- Scriptable log format

Supported Environment

Kerberos authentication does not need any specific environment on the Citrix ADC. The client (browser) must provide support for Kerberos authentication.

High Availability

In a high availability setup, only the active Citrix ADC joins the domain. In case of a failover, the Citrix ADC lwagent daemon joins the secondary Citrix ADC appliance to the domain. No specific configuration is required for this functionality.

Kerberos authentication process

The following figure shows a typical process for Kerberos authentication in the Citrix ADC environment.

Figure 1. Kerberos Authentication Process on Citrix ADC



The Kerberos authentication occurs in the following stages:

Client authenticates itself to the KDC

1. The Citrix ADC appliance receives a request from a client.
2. The traffic management (load balancing or content switching) virtual server on the Citrix ADC appliance sends a challenge to the client.
3. To respond to the challenge, the client gets a Kerberos ticket.
   - The client sends the Authentication Server of the KDC a request for a ticket-granting ticket (TGT) and receives the TGT. (See 3, 4 in the figure, Kerberos Authentication Process.)
   - The client sends the TGT to the Ticket Granting Server of the KDC and receives a Kerberos ticket. (See 5, 6 in the figure, Kerberos Authentication Process.)

> **Note**
>
> The above authentication process is not necessary if the client already has a Kerberos ticket whose lifetime has not expired. In addition, clients such as Web Services, .NET, or J2EE, which support SPNEGO, get a Kerberos ticket for the target server, create an SPNEGO token, and insert the token in the HTTP header when they send an HTTP request. They do not go through the client authentication process.

Client requests a service.

1. The client sends the Kerberos ticket containing the SPNEGO token and the HTTP request to the traffic management virtual server on the Citrix ADC. The SPNEGO token has the necessary GSSAPI data.
2. The Citrix ADC appliance establishes a security context between the client and the Citrix ADC. If the Citrix ADC cannot accept the data provided in the Kerberos ticket, the client is asked to get a different ticket. This cycle repeats till the GSSAPI data is acceptable and the security context is established. The traffic management virtual server on the Citrix ADC acts as an HTTP proxy between the client and the physical server.

Citrix ADC appliance completes the authentication.

1. After the security context is complete, the traffic management virtual server validates the SPNEGO token.
2. From the valid SPNEGO token, the virtual server extracts the user ID and GSS credentials, and passes them to the authentication daemon.
3. A successful authentication completes the Kerberos authentication.

Configuring kerberos authentication on the Citrix ADC appliance

September 14, 2021

This topic provides the detailed steps to configure Kerberos authentication on the Citrix ADC appliance by using the CLI and the GUI.

Configuring Kerberos authentication on the CLI

1. Enable the authentication, authorization, and auditing feature to ensure the authentication of traffic on the appliance.

   ns-cli-prompt> **enable ns feature** AAA

2. Add the keytab file to the Citrix ADC appliance. A keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. A single keytab file contains authentication details for all the services that are bound to the traffic management virtual server on the Citrix ADC appliance.

   First generate the keytab file on the Active Directory server and then transfer it to the Citrix ADC appliance.

   - Log on to the Active Directory server and add a user for Kerberos authentication. For example, to add a user named “Kerb-SVC-Account”:

net user Kerb-SVC-Account freebsd!@#456 /add

> **Note**
>
> In the **User Properties** section, ensure that the “Change password at next logon option” is not selected and the “Password does not expire” option is selected.

- Map the HTTP service to the above user and export the keytab file. For example, run the following command on the Active Directory server:

  ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456 /mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL

  > **Note**
  >
  > You can map more than one service if authentication is required for more than one service. If you want to map more services, repeat the above command for every service. You can give the same name or different names for the output file.

- Transfer the keytab file to the Citrix ADC appliance by using the unix **ftp** command or any other file transfer utility of your choice.

3. The Citrix ADC appliance must obtain the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, Citrix recommends configuring the Citrix ADC with a DNS server.

   ns-cli-prompt> **add dns nameserver** <ip-address>

   > **Note**
   >
   > Alternatively, you can add static host entries or use any other means so that the Citrix ADC appliance can resolve the FQDN name of the domain controller to an IP address.

4. Configure the authentication action and then associate it to an authentication policy.

   - Configure the negotiate action.

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

     Note: For domain user and domain name configuration, go to client and use the klist command as shown in the following example:

     Client: username @ AAA.LOCAL

     Server: HTTP/onprem_idp.aaa.local @ AAA.LOCAL

     add authentication negotiateAction <name> -domain -domainUser <HTTP/onprem_idp.aaa.local>

- Configure the negotiate policy and associate the negotiate action to this policy.

  ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. Create an authentication virtual server and associate the negotiate policy with it.
   - Create an authentication virtual server.

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - Bind the negotiate policy to the authentication virtual server.

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   > **Note**
   >
   > Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations by doing the following:
   - Access the traffic management virtual server, using the FQDN. For example, Sample
   - View the details of the session on the CLI.

     ns-cli-prompt> **show aaa session**

Configuring Kerberos authentication on the GUI

1. Enable the authentication, authorization, and auditing feature.

   Navigate to **System** > **Settings**, click **Configure Basic Features** and enable the authentication, authorization, and auditing feature.

2. Add the keytab file as detailed in step 2 of the CLI procedure mentioned above.
3. Add a DNS server.

   Navigate to **Traffic Management > DNS > Name Servers**, and specify the IP address for the DNS server.

4. Configure the **Negotiate** action and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, and create a policy with **Negotiate** as the action type. Click **ADD** to create a new authentication negotiate server or click **Edit** to configure the existing details.

5. Bind the negotiate policy to the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the **Negotiate** policy with the authentication virtual server.

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and specify the relevant authentication settings.

   > **Note**
   >
   > Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations as detailed in step 7 of the CLI procedure mentioned above.

Configure kerberos authentication on a client

September 14, 2021

Kerberos support must be configured on the browser to use Kerberos for authentication. You can use any Kerberos-compliant browser. Instructions for configuring Kerberos support on Internet Explorer and Mozilla Firefox follow. For other browsers, see the documentation of the browser.

To configure Internet Explorer for Kerberos authentication

1. In the **Tools** menu select **Internet Options**.
2. On the **Security** tab, click **Local Intranet**, and then click **Sites**.
3. In the **Local Intranet** dialog box, make sure that the Automatically detect intranet network option is selected, and then click **Advanced**.
4. In the **Local Intranet** dialog box, add the web sites of the domains of the traffic management virtual server on the Citrix ADC appliance. The specified sites become local intranet sites.
5. Click **Close** or **OK** to close the dialog boxes.

To configure Mozilla Firefox for Kerberos authentication

1. Make sure that you have Kerberos properly configured on your computer.
2. Type about:config in the URL bar.
3. In the filter text box, type network.negotiate.
4. Change network.negotiate-auth.delegation-uris to the domain that you want to add.
5. Change network.negotiate-auth.trusted-uris to the domain that you want to add.

> **Note:** If you are running Windows, you also need to enter sspi in the filter text box and change the network.auth.use-sspi option to False.

Offload Kerberos authentication from physical servers

September 14, 2021

The Citrix ADC appliance can offload authentication tasks from servers. Instead of the physical servers authenticating the requests from clients, the Citrix ADC authenticates all the client requests before it forwards them to any of the physical servers bound to it. The user authentication is based on Active Directory tokens.

There is no authentication between the Citrix ADC and the physical server, and the authentication offload is transparent to the end users. After the initial logon to a Windows computer, the end user does not have to enter any additional authentication information in a pop-up or on a logon page.

In the current Citrix ADC appliance release, Kerberos authentication is available only for authentication, authorization, and auditing traffic management virtual servers. Kerberos authentication is not supported for SSL VPN in the Citrix Gateway Advanced Edition appliance or for Citrix ADC appliance management.

Kerberos authentication requires configuration on the Citrix ADC appliance and on client browsers.

To configure Kerberos authentication on the Citrix ADC appliance

1. Create a user account on Active Directory. When creating a user account, verify the following options in the User Properties section:
    - Make sure that you do not select the Change password at next logon option.
    - Be sure to select the Password does not expire option.
2. On the AD server, at the CLI command prompt, type:
    - ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

    > **Note**
    >
    > Be sure to type the above command on a single line. The output of the above command is written into the C:\kerbtabfile.txt file.
3. Upload the kerbtabfile.txt file to the /etc directory of the Citrix ADC appliance by using a Secure Copy (SCP) client.
4. Run the following command to add a DNS server to the Citrix ADC appliance.

- add dns nameserver 1.2.3.4

The Citrix ADC appliance cannot process Kerberos requests without the DNS server. Be sure to use the same DNS server that is used in the Microsoft Windows domain.

5. Switch to the command line interface of Citrix ADC.
6. Run the following command to create a Kerberos authentication server:
   - add authentication negotiateAction KerberosServer -domain “crete.lab.net” -domainUser kerbuser -domainUserPasswd Citrix1 -keytab /var/mykcd.keytab

   > **Note**
   >
   > If keytab is not available, you can specify the parameters: domain, domainUser, and -domainUserPasswd.

7. Run the following command to create a negotiation policy:
   - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`
8. Run the following command to create an authentication virtual server.
   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`
9. Run the following command to bind the Kerberos policy to the authentication virtual server:
   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`
10. Run the following command to bind an SSL certificate to the authentication virtual server. You can use one of the test certificates, which you can install from the GUI Citrix ADC appliance. Run the following command to use the ServerTestCert sample certificate.
    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`
11. Create an HTTP load balancing virtual server with the IP address, 192.168.17.200.

    Ensure that you create a virtual server from the command line interface for NetScaler 9.3 releases if they are older than 9.3.47.8.
12. Run the following command to configure an authentication virtual server:
    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`
13. Enter the host name Example in the address bar of the Web browser.

    The Web browser displays an authentication dialog box because the Kerberos authentication is not set up in the browser.

> **Note**
>
> Kerberos authentication requires a specific configuration on the client. Ensure that the client can resolve the hostname, which results in the Web browser connecting to an HTTP virtual server.

14. Configure Kerberos on the Web browser of the client computer.
    - For configuring on Internet Explorer, see Configuring Internet Explorer for Kerberos authentication.
    - For configuring on Mozilla Firefox, see Configuring Internet Explorer for Kerberos authentication.
15. Verify whether you can access the backend physical server without authentication.

To configure Internet Explorer for Kerberos authentication

1. Select **Internet Options** from the **Tools** menu.
2. Activate the **Security** tab.
3. Select **Local Intranet** from the Select a zone to view change security settings section.
4. Click **Sites**.
5. Click **Advanced**.
6. Specify the URL, Example and click **Add**.
7. Restart **Internet Explorer**.

To configure Mozilla Firefox for Kerberos authentication

1. Enter about:config in the address bar of the browser.
2. Click the warning disclaimer.
3. Type **Network.Negotiate-auth.trusted-uris** in the **Filter** box.
4. Double click **Network.Negotiate-auth.trusted-uris.** A sample screen is shown below.



5. In the Enter String Value dialog box, specify www.crete.lab.net.
6. Restart Firefox.

Troubleshoot authentication and authorization related issues

September 14, 2021

Localize error messages

Localize error messages generated by Citrix ADC nFactor system

Troubleshoot authentication issues with aaad.debug module

Troubleshoot authentication issues in Citrix ADC and Citrix Gateway with aaad.debug module

Admin partition

September 14, 2021

A Citrix ADC appliance can be partitioned into logical entities called admin partitions. Each partition can be configured and used as a separate Citrix ADC appliance. The following figure shows the partitions of a Citrix ADC being used by different customers and departments:



A partitioned Citrix ADC appliance has a single default partition and one or more admin partitions. The following table provides further details on the two partition types:

> **Note**
>
> In a partitioned appliance, the mode BridgeBPDUs can be enabled only in the default partition and not in the administrative partitions.

Availability:

The Citrix ADC appliance ships with a single partition, which is called a default partition. The default partition is retained even after the Citrix ADC appliance is partitioned.

Must be explicitly created as described in Configure admin partitions.

Number of Partitions:

One

A Citrix ADC appliance can have one or more (maximum of 512) admin partitions.

User Access and Roles:

All Citrix ADC users, who are not associated with a *partition-specific* command policy, can access and configure the default partition. As always, the associated command policy restricts the operations that a user can perform.

The user access and roles are created by Citrix ADC superusers who also specify the users for that partition. Only superusers and associated users of the partition can access and configure the admin partition.

> **Note**
>
> Partition users do not have shell access.

File Structure:

All files in a default partition are stored in the default Citrix ADC file structure.

For example, the */nsconfig* directory stores the Citrix ADC configuration file and the */var/log/* directory stores the Citrix ADC logs.

All files in an admin partition are stored in directory paths that have the name of the admin partition.

For example, the Citrix ADC configuration file (ns.conf) is stored in the `/nsconfig/partitions/<partitionName>` directory. Other partition-specific files are stored in the `/var/partitions/<partitionName>` directories.

Some other paths in an admin partition:

- Downloaded files: `/var/partitions/<partitionName>/download/`
- Log files: `/var/partitions/<partitionName>/log/`

> **Note**
>
> Currently, logging is not supported at partition-level. Therefore, this directory is empty and all logs are stored in the `/var/log/` directory.

- SSL CRL certificate related files: `/var/partitions/<partitionName>/netscaler/ssl`

Resources Available:

All Citrix ADC resources.

Citrix ADC resources that are explicitly assigned to the admin partition.

User access and roles

In authenticating and authorizing a partitioned Citrix ADC appliance, a root administrator can assign a partition administrator to one or more partitions. The partition administrator can authorize users to that partition without affecting other partitions. The partition users are authorized to access only that partition using the SNIP address. Both the root administrator and the partition administrator can configure role based access (RBA by authorizing users to access different applications.

Administrators and user roles can be described as follows:

Root Administrator. Accesses the partitioned appliance through its NSIP address and can grant user access to one or more partitions. The administrator can also assign partition administrators to one or more partitions. The administrator can create a partition administrator from the default partition using an NSIP address or switch to a partition and then create a user and assign partition admin access using a SNIP address.

Partition Administrator. Accesses the specified partition through an NSIP address assigned by the root administrator. The administrator can assign role-based access to partition user access to that partition and also configure external server authentication using partition specific configuration.

System User. Accesses partitions through the NSIP address. Has access to the partitions and resources specified by the root administrator.

Partition User. Accesses a partition through a SNIP address. The user account is created by the partition administrator and the user has access to resources, only within the partition.

Points to remember

Following are some points to remember when providing role-based access in a partition.

1. Citrix ADC users accessing the GUI through the NSIP address uses the default partition authentication configuration to log on to the appliance.
2. Partition system users accessing the GUI through a partition SNIP address uses partition specific authentication configuration to log on to the appliance.

3. Partition user created in a partition cannot log in using the NSIP address.
4. The Citrix ADC user bound to a partition cannot log in using the partition SNIP address.
5. System users authenticating through an external authentication server (for example, LDAP, RADIUS, TACACS) must access a partition through a SNIP address.

Use case for managing role based access in a partitioned setup

Consider a scenario where an enterprise organization, www.example.com has multiple business units and a centralized administrator who manages all instances in their network. However, they want to provide exclusive user privileges and environment for each business unit.

Following are the administrators and users managed by default partition authentication configuration and partition specific configuration in a partitioned appliance.

John: Root Administrator

George: Partition Administrator

Adam: System User

Jane: Partition User

John, is the root administrator of a partitioned Citrix ADC appliance. John manages all user accounts and administrative user accounts across partitions (for example, P1, P2, P3, P4, and P5) within the appliance. John provides granular role-based access to entities from the default partition of the appliance. John creates user accounts and assigns partition access to each account. George being a network engineer within the organization prefers to have a role based access to few applications running on partition P2. Based on user management, John creates a partition administrator role for George and associates his user account with a partition-admin command policy in the P2 partition. Adam being another network engineer prefers to access an application running on P2. John creates a system user account for Adam and associates his user account to a P2 partition. Once the account is created, Adam can log into the appliance to access the Citrix ADC management interface through the NSIP address and can switch to partition P2 based on user/group binding.

Suppose, Jane who is another network engineer wants to directly access an application running only on partition P2, George (partition administrator) can create a partition user account for her and associate her account with command policies for authorization privileges. Jane's user account created within the partition is now directly associated with P2. Now Jane can access the Citrix ADC management interface through the SNIP address and cannot switch to any other partition.

> **Note**
>
> If Jane's user account is created by a partition administrator in partition P2, the admin can access the Citrix ADC management interface only through the SNIP address (created within the partition). The admin is not permitted to access the interface through the NSIP address. Similarly, if

> Adam's user account is created by a root administrator in the default partition and is bound to a P2 partition. The admin can access the Citrix ADC management interface only through the NSIP address or SNIP address created in the default partition (with management access enabled). And not permitted to access the partition interface through the SNIP address created in the administrative partition.

Configure roles and responsibilities for partition administrators

Following are the configurations performed by a root administrator in a default partition.

Creating administrative partitions and system users – A root administrator creates administrative partitions and system users in the default partition of the appliance. The administrator then associates the users to different partitions. If you are bound to one or more partitions, you can switch from one partition to another based on user bindings. Also, your access to one or more bound partitions is authorized only by the root administrator.

Authorizing the system user as partition administrator for a specific partition – Once a user account is created, the root administrator switches to a specific partition and authorizes the user as the partition administrator. It is done by assigning partition-admin command policy to the user account. Now, the user can access the partition as partition administrator and manage entities within the partition.

Following are the configurations perform by a partition administrator in an administrative partition.

Configuring the SNIP address in an administrative partition- The partition administrator logs on to the partition and creates a SNIP address and provides management access to the address.

Creating and Binding a Partition System User with Partition Command Policy - The partition administrator creates partition users and defines the scope of user access. It is done by binding the user account to partition command policies.

Creating and Binding a Partition System User Groups with Partition Command Policy -The partition administrator creates partition user groups and defines the scope of user group access. It is done by binding the user group account to partition command policies.

Configuring External Server authentication for external users (optional)-This configuration is done for authenticating external TACACS users accessing the partition using the SNIP address.

Following are the tasks performed in configuring role-based access for partition users in an Administrative Partition.

1. Creating an Administrative Partition – Before you create partition users in an administrative partition, you must first create the partition. As a root administrator, you can create a partition from the default partition using the configuration utility or a command line interface.
2. Switching user access from default partition to partition P2 – If you are partition administrator accessing the appliance from the default partition, you can switch from default partition to a specific partition. For example, partition P2 based on user binding.

3. Adding a SNIP address to the partition user account with management access enabled-Once you have switched your access to an administration partition. You create a SNIP address and provide management access to the address.
4. Creating and binding a partition System User with Partition Command Policy-If you are a partition administrator, you can create partition users and define the scope of user access. It is done by binding the user account to partition command policies.
5. Creating and binding partition user group with partition command policy - If you are a partition administrator, you can create partition user groups and define the scope of user access control. It is done by binding the user group account to partition command policies.

Configuring External Server authentication for external users (optional)-This configuration is done for authenticating external TACACS users accessing the partition using a SNIP address.

Benefits of using admin partitions

You can avail the following benefits by using admin partitions for your deployment:

- Allows delegation of administrative ownership of an application to the customer.
- Reduces the cost of ADC ownership without compromising on performance and ease-of-use.
- Safeguards from unwarranted configuration changes. In a non-partitioned Citrix ADC appliance, authorized users of the other application can intentionally or unintentionally change configurations that are required for your application. It can lead to undesirable behavior. This possibility is reduced in a partitioned Citrix ADC appliance.
- Isolates traffic between different applications by the use of dedicated VLANs for each partition.
- Accelerates and allows application deployments to scale.
- Allows application-level or localized management and reporting.

Let us analyze a couple of cases to understand the scenarios in which you can use admin partitions.

User case 1: How admin partition is used in an enterprise network

Let us consider a scenario faced by a company named **Foo.com**.

- **Foo.com** has a single Citrix ADC.
- There are five departments and each department has one application that requires to be deployed with the Citrix ADC.
- Each application must be managed independently by a different set of users or administrators.
- Other users must be restricted from accessing the configurations.
- The application or back-end must be able to share resources like IP addresses.
- The global IT department must be able to control Citrix ADC-level settings which must be common to all partitions.
- Applications must be independent of one another. An error in the configuration of one application must not affect the other.

A non-partitioned Citrix ADC would not be able to satisfy these requirements. However, you can achieve all these requirements by partitioning a Citrix ADC.

Simply create a partition for each of the applications, assign the required users to the partitions, specify a VLAN for each partition, and define global settings on the default partition.

Use case 2: How an admin partition is used by a service provider

Let us consider a scenario faced by a service provider named **BigProvider**:

- BigProvider has 5 customers: 3 small enterprises and 2 large enterprises.
- **SmallBiz**, **SmallerBiz**, and **StartupBiz** need only the most basic Citrix ADC functionality.
- **BigBiz** and **LargeBiz** are larger enterprises and have applications that attract heavy traffic. They would like to use some of the more complex Citrix ADC functionality.

In a non-partitioned approach, the Citrix ADC administrator would typically use a Citrix ADC SDX appliance and provision a Citrix ADC instance for each customer.

The solution suits **BigBiz** and **LargeBiz** because their applications need the undiminished power of the entire non-partitioned Citrix ADC appliance. However, this solution might not be as cost effective for servicing **SmallBiz**, **SmallerBiz**, and **StartupBiz**.

Therefore, **BigProvider** decides on the following solution:

- Using a Citrix ADC SDX appliance to bring up dedicated Citrix ADC instances for **BigBiz** and **LargeBiz**.
- Using a single Citrix ADC which is partitioned into three partitions, one each for **SmallBiz**, **SmallerBiz**, and **StartupBiz**.

The Citrix ADC administrator (superuser) creates an admin partition for each of these customers, and specifies the users for the partitions. And also specifies the Citrix ADC resources for the partitions, and specifies the VLAN to be used by the traffic that is destined for each of the partitions.

Citrix ADC configurations support in admin partition

September 14, 2021

Citrix ADC configurations can be categorized into the following three types of configurations. It depends on the Citrix configuration and the partition in which the configuration is performed.

> **Note**
>
> - Admin partitions cannot be set up on a Citrix ADC cluster. It means that a Citrix ADC cluster cannot be partitioned.

- Admin partitions cannot be set up on a Citrix ADC 14000 FIPS appliance.
- Case 3 lists the Citrix ADC features that are not supported in admin partitions.
- Load balancing templates are not supported in admin partitions.

Case 1 (global configurations)

Configurations that can be performed ONLY in the default partition and which are available or impact all the admin partitions.



- Updates to built-in entities for monitors, TCP profiles, HTTP profiles, and so on.
- Updates to global parameters for syslog, NSLOG, weblog, content switching, IPSEC, SIP, DHCP, Surge protection, TCP buffering, and system collection.
- High availability (HA) configurations
- Interface and VLAN changes
- User configurations

Case 2 (partition-specific configurations)

Configurations that can be performed independently in default and admin partitions. These configurations are applicable only to the partition in which they are performed.



- Getting traffic level statistics for a partition.

- Partition admin can update IP bindings for VLAN which is bound to that partition. But cannot update the interface bindings.
- Clearing Citrix ADC configurations.
- Feature-specific parameters for the following features: AppFlow, AppQoE, HTTP compression, DNS, TCP, HTTP, encryption, responder, rewrite, and SSL.
- Feature-specific configurations such as virtual servers, services, monitors.

Case 3

Configurations that cannot be performed on admin partitions. These features can be configured in the default partition, but there is no impact on admin partitions.



> **Note:**
>
> Configurations that are supported on admin partitions for a particular release are marked as **Yes**.

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Networking	Traffic Domain	No (Not supported from build 60.13 onwards)	No	No	No
Policy	Extensibility	Yes	Yes	Yes	Yes
Load Balancing	DBS Autoscale	Yes	Yes	Yes	Yes
Load Balancing	DNSSEC	No	No	Yes	Yes
Load Balancing	Diameter	Yes	Yes	Yes	Yes

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Load Balancing	RTSP	No	No	No	No
Load Balancing	Sure Connect	Yes	Yes	Yes	Yes
Load Balancing	Autoscale Service Group	Yes	Yes	Yes	Yes
Manageability	RBA External Authentica-tion	Yes	Yes	Yes	Yes
Manageability	RISE Cisco	No	No	No	No
Manageability	ACI-Cisco	Yes	Yes	Yes	Yes
Manageability	AppExpert	Yes	Yes	Yes	Yes
Manageability	HDX Insight	No	No	No	No
Manageability	Insight	No	No	No	No
VPN	Citrix CloudBridge Connector	No	No	No	No
VPN	Citrix Gateway or SSL VPN	No	No	No	No
VPN	SSL VPN ICA Proxy	No	No	No	No
VPN	Web Interface on Citrix ADC	No	No	No	No
SSL	SSL Profile	Yes	Yes	Yes	Yes
SSL	SSL-FIPS	No	No	No	No
SSL	External-HSM	No	No	No	No
Infra	Cache Redirection	No	No	No	No
Infra	Integrated Caching	Yes	Yes	Yes	Yes

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Network	VXLAN	Yes	Yes	Yes	Yes
Network	Graceful Shutdown	Yes	Yes	Yes	Yes
Network	LSN	No	No	No	No
Network	IPv6 Ready Logo	Yes	Yes	Yes	Yes
Network	vPath	Yes	Yes	Yes	Yes
Load Balancing	Datastream	Yes	Yes	Yes	Yes
Logging	Web logging	Yes	Yes	Yes	Yes
Network	L2 Param/L3 Param	Yes	Yes	Yes	Yes
Network	GRE Tunnel	Yes	Yes	Yes	Yes
Loading Balancing	Scriptable Monitoring	Yes	Yes	Yes	Yes
Load Balancing	GSLB	Yes	Yes	Yes	Yes
Infra	Connection Mirroring	Yes	Yes	Yes	Yes
Infra	FEO	Yes	Yes	Yes	Yes
Infra	Ns trace	Yes	Yes	Yes	Yes
Load Balancing	Priority Queuing	Yes	Yes	Yes	Yes
Network	HDOSP	Yes	Yes	Yes	Yes
Network	Net profile	Yes	Yes	Yes	Yes
Network	Networking (Restricted Feature)	Yes	Yes	Yes	Yes
Network	VRRP (Restricted Feature)	Yes	Yes	Yes	Yes

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Logging	Audit Logging (SYSLOG-TCP, LB of syslog servers, SNIP support, and FQDN support for syslog)	Yes	Yes	Yes	Yes
VPN	Citrix Gateway	No	No	No	No
VPN	AAA-TM	Yes	Yes	Yes	Yes
AppFlow	AppFlow	No	Yes (IPFIX only)	Yes (IPFIX only)	Yes
AppFw	Application Firewall	No	No	No	No
URL transformation	URL transformation	No	No	No	No
Load Balancing	TCP Buffering	No	No	No	No
Policies	OCSP Responder	Yes	Yes	Yes	Yes
Audit Log	SYSLOG-TCP	No	Yes	Yes	Yes
Optimization	Front-end-optimization	No	Yes	Yes	Yes
AppQoE	AppQoE	Yes	Yes	Yes	Yes
BOT	BOT Management	NA	NA	NA	NO

The preceding table lists some of the features as **Restricted Features** in the admin partition setup. The following section provides the reason why some of the features are mentioned as **Restricted Features**.

- **VRRP**. The VRRP is Restricted Feature in the admin partition because of the following:
  - VRID addition or deletion can be done only from the default partition context. However,

once a VRID is created, it can be used within non-default partitions.

- VRRP functionality is supported only over the dedicated VLANs.
- VRRP functionality is not supported on shared VLANs, used by the admin partition. It is blocked internally. No error message is shown during configuration. The protocol is blocked on a shared VLAN (tagged or untagged) bound to a default or any administrative partition.

> **Important**
>
> To support active-active deployment using VRRP, main and backup VIP must use the same VRID. Different VRIDs cannot be used.

- **Networking**. Some of the networking configurations (L2 Param and L3 Param) are not supported or valid in the partition context. If you come across any such configurations, the following error message is displayed. "ERROR: This configuration option is not supported on the non-default partition."

Configure admin partitions

September 14, 2021

> **Important**
>
> - Only superusers are authorized to create and configure admin partitions.
> - Unless specified otherwise, configurations to set up an admin partition must be done from the default partition.

By partitioning a Citrix ADC appliance, you are in-effect creating multiple instances of a single Citrix ADC appliance. Each instance has its own configurations and the traffic of each of these partitions is isolated from the other. It is done by assigning each partition a dedicated VLAN or a shared VLAN.

A partitioned Citrix ADC has one default partition and the admin partitions that are created. To set up an admin partition, you must first create a partition with the relevant resources (memory, maximum bandwidth, and connections). Then, specify the users that can access the partition and the level of authorization for each of the users on the partition.

Accessing a partitioned Citrix ADC is the same as accessing a non-partitioned Citrix ADC: through the NSIP address or any other management IP address. As a user, after you provide your valid logon credentials, you are taken to the partition to which you are bound. Any configurations that you create are saved to that partition. If you are associated with more than one partition, you are taken to the first partition with which you were associated. If you want to configure entities on one of your other partitions, you must explicitly switch to that partition.

After accessing the appropriate partition, the configurations that you perform are saved to that partition and are specific to that partition.

> **Note**
>
> - Citrix ADC superusers and other non-partition users are taken to the default partition.
> - Users of all the 512 partitions can log in simultaneously.

Tip

To access a partitioned Citrix ADC appliance over HTTPS by using the SNIP (with management access enabled), make sure that each partition has the certificate of its partition administrator. Within the partition, the partition admin must do the following:

1. Add the certificate to the Citrix ADC.

   ```
   add ssl certKey ns-server-certificate -cert ns-server.cert-key ns-server.key
   ```

2. Bind it to a service named `nshttps-<SNIP>-3009`, where `<SNIP>` must be replaced with the SNIP address, in this case 100.10.10.1.

   ```
   bind ssl service nshttps-100.10.10.1-3009 -certkeyName ns-server-certificate
   ```

Partition resource limiting

In a partitioned Citrix ADC appliance, a network administrator can create a partition with partition resources such as memory, bandwidth, and connection limit configured as unlimited. It is done by specifying Zero as the partition resource value. Where Zero indicates the resource is unlimited on the partition and it can be consumed up to system limits. Partition resource configuration is useful when you migrate a traffic domain deployment to an administrative partition or if you do not know about the resource allocation limit for a partition in a given deployment.

Resource limit for an administrative partition is as follows:

1. **Partition memory**. It is the maximum allocated memory for a partition. You make sure to specify the values when creating a partition.

   > **Note**
   >
   > From NetScaler 12.0 onwards, when you create a partition, you can set the memory limit to Zero. If a partition is already created with a specific memory limit, you can reduce the limit to any value or set the limit as Zero.

   Parameter: maxMemLimit

Maximum memory is allocated in MB in a partition. A zero value indicates the memory is unlimited on the partition and it can consume up to the system limits.

Default value: 10

2. **Partition bandwidth**. Maximum allocated bandwidth for a partition. If you specify a limit, make sure it is within the appliance's licensed throughput. Otherwise, you are not limiting the bandwidth that is used by the partition. The specified limit is accountable for the bandwidth that the application requires. If the application bandwidth exceeds the specified limit, packets are dropped.

   > **Note**
   >
   > From NetScaler 12.0 onwards, when you can create a partition, you can set the partition bandwidth limit to Zero. If a partition is already created with a specific bandwidth, you can reduce the bandwidth or set the limit as Zero.

   Parameter: maxBandwidth

   Maximum bandwidth is allocated in Kbps in a partition. A zero value indicates the bandwidth is unrestricted. That is, the partition can consume up to the system limits.

   Default value: 10240

   Maximum Value: 4294967295

3. **Partition connection**. Maximum number of concurrent connections that can be open in a partition. The value must accommodate the maximum simultaneous flow expected within the partition. The partition connections are accounted from the partition quota memory. Previously, the connections were accounted from the default partition quota memory. It is configured only on the client-side, not on the back-end server-side TCP connections. New connections cannot be established beyond this configured value.

   > **Note**
   >
   > From NetScaler 12.0 onwards, you can create a partition with the number of open connections set to Zero. If you have already created a partition with a specific number of open connections, you can reduce the connection limit or set the limit as Zero.

   Parameter: maxConnections

   Maximum number of concurrent connections that can be open in the partition. A zero value indicates no limit on the number of open connections.

   Default value: 1024

   Minimum value: 0

   Maximum Value: 4294967295

Configure an admin partition

To configure an admin partition, complete the following tasks.

To access in an admin partition by using the CLI

1. Log on to the Citrix ADC appliance.
2. Check if you are in the correct partition. The command prompt displays the name of the currently selected partition.
3. If yes, skip to the next step.
4. If no, get a list of the partitions with which you are associated and switch over to the appropriate partition.
   - `show system user <username>`
   - **`switch`** `ns partition <partitionName>`
5. Now, you can perform the required configurations just as a non-partitioned Citrix ADC.

To access an admin partition by using the GUI

1. Log on to the Citrix ADC appliance.
2. Check if you are in the correct partition. The top bar of the GUI displays the name of the currently selected partition.
   - If yes, skip to the next step.
   - If no, navigate to **Configuration > System > Partition Administration > Partitions**, right-click the partition to which you want to switch, and select **Switch**.
3. Now, you can perform the required configurations just as a non-partitioned Citrix ADC.

Add an admin partition

The root administrator adds an administrative partition from the default partition and binds the partition with VLAN 2.

To create an administrative partition by using the CLI

At the command prompt, type:

```
1 add partition <partitionname>
```


Switch user access from default partition to an admin partition

Now you can switch user access from default partition to partition Par1.

To switch a user account from default partition to an admin partition by using the CLI:

At the command prompt, type:

```
Switch ns partition <pname>
```

Adding SNIP address to a partition user account with management access enabled

In the partition, create a SNIP address with management access enabled.

To add SNIP address to the partition user account with management access enabled by using the command line interface:

At the command prompt, type:

```
> add ns ip <ip address> <subnet mask> -mgmtAccess enabled
```

Create and Bind a partition user with partition command policy

In partition, create a partition system user and bind the user with partition-admin command policies.

To create and bind a partition system user with partition command policy by using the CLI:

At the command prompt, type:

```
> add system user <username> <password>

Done
```

Creating and binding partition user group with partition command policy

In Partition Par1, create a partition system user group and bind the group with partition command policy such as partition admin, partition read-only, partition-operator, or partition-network.

To create and bind a partition user group with partition command policy by using the command line interface:

```
> add system group <groupName>
> bind system group <groupname> (-userName | -policyName <cmdpolicy> <
    priority> | -partitionName)
```


Configuring external server authentication for external users

In partition Par1 you can configure an external server authentication to authenticate external TACACS users accessing the partition through a SNIP address.

To configure external server authentication for external users by using the command line interface:

At the command prompt, type:

```
> add authentication tacacsaction <name> -serverip <IP> -tacacsSecret <
    secret key> -authorization ON -accounting ON
> add authentication policy <policname> -rule true -action <name>
> bind system global <policyname> -priority <value>1
```

Configure a partition system user account in a partition by using the GUI

To configure a partition user account in an administrative partition, you must create a partition user or a partition user group and bind it partition command policies. Also, you can configure the external server authentication for an external user.

To create a partition user account in a partition by using the GUI

Navigate to **System > User Administration**, click **Users** to add a partition system user, and bind the user to command policies (partitionadmin/partitionread-only/partition-operator/partition-network).

To create a partition user group account in a partition by using the GUI

Navigate to **System > User Administration**, click **Groups** to add a partition system user group and bind the user group to command policies (partitionadmin/partitionread-only/partition-operator/partition-network).

To configure external server authentication for external users by using the GUI

Navigate to **System > Authentication > Basic Actions** and click **TACACS** to configure a TACACS server for authenticating external users accessing the partition.

Sample configuration

The following configuration shows how to create a partition user or a partition user group and bind it partition command policies. Also, how to configure the external server authentication for authenticating an external user.


```
> add partition Par1
> switch ns partition Par1
> add ns ip 10.102.29.203 255.255.255.0 -mgmtAccessenabled
> add system user John Password
> bind system user Jane partition-read-only -priority 1
> add system group Retail
> bind system group Retail -policyname partition-network 1 (where 1 is
   the priority number)
> bind system group Retail  - username Jane
> add authentication tacacssaction tacuser  - serverip 10.102.29.200  -
   tacacsSecret Password  - authorization ON  - accounting ON
> add authentication policy polname  - rule true  - action tacacsAction
> bind system global polname  - priority 1
```

Command policies for a partition users and partition user groups in administrative partition

Commands to authorize a user account inside administrative partition	Command policies available inside an administrative partition (built-in policies)	User account access type
add system user	Partition-admin	SNIP (with management access enabled)
add system group	Partition-network	SNIP (with management access enabled)
add authentication <action, policy>, bind system global <policy name>	Partition-read-only	SNIP (with management access enabled)
remove system user	Partition-admin	SNIP(with management access enabled)
remove system group	Partition-admin	SNIP (with management access enabled)
bind system cmdpolicy to system user; bind system cmdpolicy to system group	Partition-admin	SNIP (with management access enabled)

Configure an LACP Ethernet channel on the default admin partition

With the Link Aggregation Control Protocol (LACP), you can combine multiple ports into a single, high-speed link (also called a channel). An LACP-enabled appliance exchanges LACP Data Units (LACPDU) over the channel.

There are three LACP configuration modes that you can enable in the default partition of a Citrix ADC appliance:

1. Active. A port in active mode sends LACPDUs. Link aggregation is formed if the other end of the Ethernet link is in the LACP active or passive mode.
2. Passive. A port in passive mode sends LACPDUs only when it receives LACPDUs. The link aggregation is formed if the other end of the Ethernet link is in the LACP active mode.
3. Disable. Link aggregation is not formed.

> **Note**
>
> By default, the link aggregation is disabled in the default partition of the appliance.

LACP exchanges LACPDU between devices connected by an Ethernet link. These devices are typically referred as an actor or partner.

A LACPDU data unit contains the following parameters:

- LACP Mode. Active, passive, or disable.
- LACP timeout. The waiting period before timing out the partner or actor. Possible values: Long and Short. Default: Long.
- Port Key. To distinguish between the different channel. When the key is 1, LA/1 is created. When the key is 2, LA/2 is created. Possible values: Integer from 1 through 8. 4 through 8 is for cluster CLAG.
- Port Priority. Minimum value: 1. Maximum value: 65535. Default: 32768.
- System Priority. Uses this priority along with the system MAC to form the system ID to uniquely identify the system during LACP negotiation with the partner. Sets system priority from 1 and 65535. The default value is set to 32768.
- Interface. Supports 8 interfaces per channel on NetScaler 10.1 appliance and supports 16 interfaces per channel on NetScaler 10.5 and 11.0 appliances.

After exchanging LACPDUs, the actor and partner negotiate the settings and decide whether to add the ports to the aggregation.

Configure and verify LACP

The following section shows how to configure and verify LACP in the admin partition.

To configure and verify LACP on a Citrix ADC appliance by using the CLI

1. Enable LACP on each interface.

   ```
   set interface <Interface_ID> -lacpMode PASSIVE -lacpKey 1<!--NeedCopy
   -->
   ```

   When you enable LACP on an interface, the channels are dynamically created. Also, when you enable LACP on an interface and set lacpKey to 1, the interface is automatically bound to channel LA/1.

   > **Note**
   >
   > When you bind an interface to a channel, the channel parameters take precedence over the interface parameters, so the interface parameters are ignored. If a channel is created dynamically by LACP, you cannot perform the add, bind, unbind, or remove operations on the channel. A channel dynamically created by LACP is automatically deleted when you disable LACP on all interfaces of the channel.

2. Set the system priority.

   ```
   set lacp -sysPriority <Positive_Integer><!--NeedCopy-->
   ```

3. Verify that LACP is working as expected.

   "'show interface

   ```
   1 ```show channel<!--NeedCopy-->
   ```

   ```
   show LACP<!--NeedCopy-->
   ```

   > **Note**
   >
   > In some versions of Cisco Internetwork Operating System (iOS), running the switchport trunk native VLAN <VLAN_ID> command causes the Cisco switch to tag LACP PDUs. It causes the LACP channel between the Cisco switch and the Citrix ADC appliance to fail. However, this issue does not affect the static link aggregation channels configured in the previous procedure.

Save configuration of all admin partitions from the default partition

Administrators can save the configuration of all the admin partitions at once from the default partition.

Save all admin partitions from default partition by using the CLI

At the command prompt, type:

```
save ns config -all
```


Support for partition and cluster based custom reports

Citrix ADC GUI displays only the custom reports created in the current viewing partition or in the cluster.

Previously, the Citrix ADC GUI used to store the Custom Report names directly to the back end file without mentioning the partition or cluster name to differentiate.

To view the custom reports of the current partition or cluster in the GUI

- Navigate to **Reporting** tab.
- Click **Custom Reports** to view the reports created in the current partition or in the cluster.

Support to bind VPN global certificates in a partitioned setup for OAuth IdP

In a Partitioned setup, you can now bind the certificates to VPN global for OAuth IdP deployments.

To bind the certificates in Partitioned setup by using the CLI

At the command prompt, type:

```
bind vpn global [-certkeyName <string>] [-userDataEncryptionKey <string
    >]
```

VLAN configuration for admin partitions

September 14, 2021

VLANs can be bound to a partition as a “Dedicated” VLAN or a “Shared” VLAN. Based on your deployment, you can bind a VLAN to a partition to isolate its network traffic from other partitions.

Dedicated VLAN – A VLAN bound only to one partition with the “Sharing” option disabled and must be a tagged VLAN. For example, in a client-server deployment, for security reasons a system administrator creates a dedicated VLAN for each partition on the server side.

Shared VLAN – A VLAN bound (shared across) to multiple partitions with the “Sharing” option enabled. For example, in a client-server deployment, if the system administrator do not have control over the client side network, a VLAN is created and shared across multiple partitions.

Shared VLAN can be used across multiple partitions. It is created in the default partition and you can bind a shared VLAN to multiple partitions. By default, a shared VLAN is bound to the default partition implicitly and hence it cannot be bound explicitly.

Note

- A Citrix ADC appliance deployed on any hypervisor (ESX, KVM, Xen, and Hyper-V) platform must comply with both the following conditions in a partition setup and traffic domain:
  - Enable the promiscuous mode, MAC changes, MAC spoofing, or forged transmit for shared VLANs with partition.
  - Enable the VLAN with port group properties of the virtual switch, if the traffic is through a dedicated VLAN.
- In a partitioned (multitenant) Citrix ADC appliance, a system administrator can isolate the traffic flowing to a particular partition or partitions. It is done by binding one or more VLANs to each partition. A VLAN can be dedicated to one partition or Shared across multiple partitions.

Dedicated VLANs

To isolate the traffic flowing into a partition, create a VLAN and associate it with the partition. The VLAN is then visible only to the associated partition, and the traffic flowing through the VLAN is classified and processed only in the associated partition.



To implement a dedicated VLAN for a particular partition, do the following.

1. Add a VLAN (V1).
2. Bind a network interface to VLAN as a tagged network interface.
3. Create a partition (P1).
4. Bind partition (P1) to the dedicated VLAN (V1).

Configure the following by using the CLI

- Create a VLAN

  ```
  add vlan <id>
  ```

Example

```
1     add vlan 100
```

- Bind a VLAN

  ```
  bind vlan <id> -ifnum <interface> -tagged
  ```

Example

```
1     bind vlan 100  - ifnum 1/8 -tagged
```

- Create a partition

  ```
  Add ns partition <partition name> [-maxBandwidth <positive_integer>][-
  maxConn <positive_integer>] [-maxMemLimit <positive_integer>]
  ```

Example

```
1     Add ns partition P1  - maxBandwidth 200  - maxconn 50  - maxmemlimit
         90
2
3     Done
```

- Bind a partition to a VLAN

  ```
  bind partition <partition-id> -vlan <id>
  ```

Example

```
1     bind partition P1  - vlan 100
```

Configure a dedicated VLAN by using the Citrix ADC GUI

1. Navigate to **Configuration > System > Network > VLANs*** and click **Add** to create a VLAN.
2. On the **Create VLAN** page, set the following parameters:
   - VLAN ID
   - Alias Name
   - Maximum Transmission Unit
   - Dynamic Routing

- IPv6 Dynamic Routing
- Partitions Sharing

3. In the **Interface Bindings** section, select one or more interfaces and bind it to the VLAN.
4. In the **IP Bindings** section, select one or more IP addresses and bind to the VLAN.
5. Click **OK** and **Done**.

Shared VLAN

In a shared VLAN configuration, each partition has a MAC address, and traffic received on the shared VLAN is classified by MAC address. Only a Layer3 VLAN is recommended because it can restrict the subnet traffic. A partition MAC address is applicable and important only for a shared VLAN deployment.

> **Note**
>
> Starting from Citrix ADC version 12.1 build 51.16, shared VLAN in a partitioned appliance support dynamic routing protocol.

The following diagram shows how a VLAN (VLAN 10) is shared across two partitions.



To deploy a shared VLAN configuration, do the following:

1. Create a VLAN with the sharing option ‘enabled’, or enable the sharing option on an existing VLAN. By default, the option is ‘disabled’.
2. Bind partition interface to shared VLAN.
3. Create the partitions, each with its own PartitionMAC address.
4. Bind the partitions to the shared VLAN.

Configure a shared VLAN by using the CLI

At the command prompt, type one of the following commands to add VLAN or set the sharing parameter of an existing VLAN:

```
add vlan <id> [-sharing (ENABLED | DISABLED)]

set vlan <id> [-sharing (ENABLED | DISABLED)]

add vlan 100  -sharing ENABLED

set vlan 100  -sharing ENABLED
```

Bind a partition to a Shared VLAN by using the CLI

At the command prompt, type:

```
bind partition <partition-id> -vlan <id>

bind partition P1  -vlan 100

add ns partition P1  -maxBandwidth 200  -maxconn 50    -maxmemlimit 90
   -partitionMAC<mac_addr

Done
```

Configure a Partition MAC Address by using the CLI

```
set ns partition <partition name> [-partitionMAC<mac_addr>]

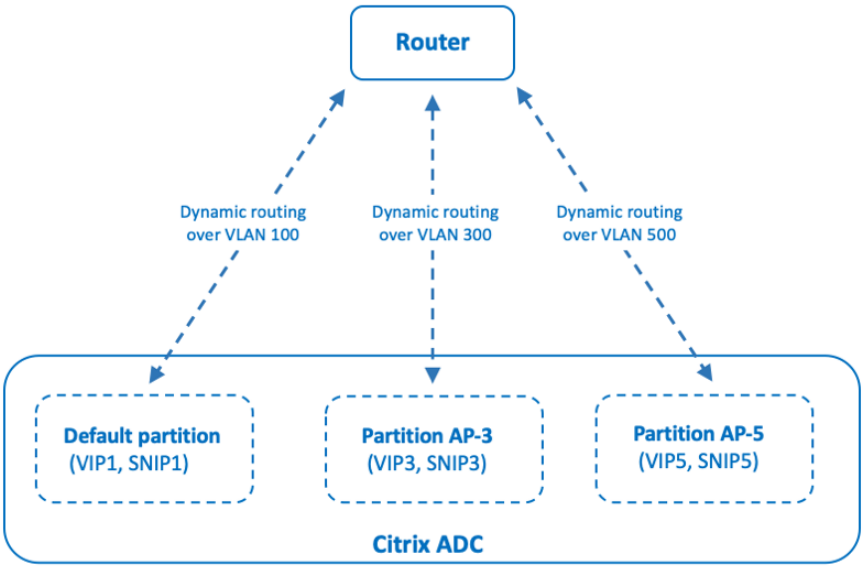
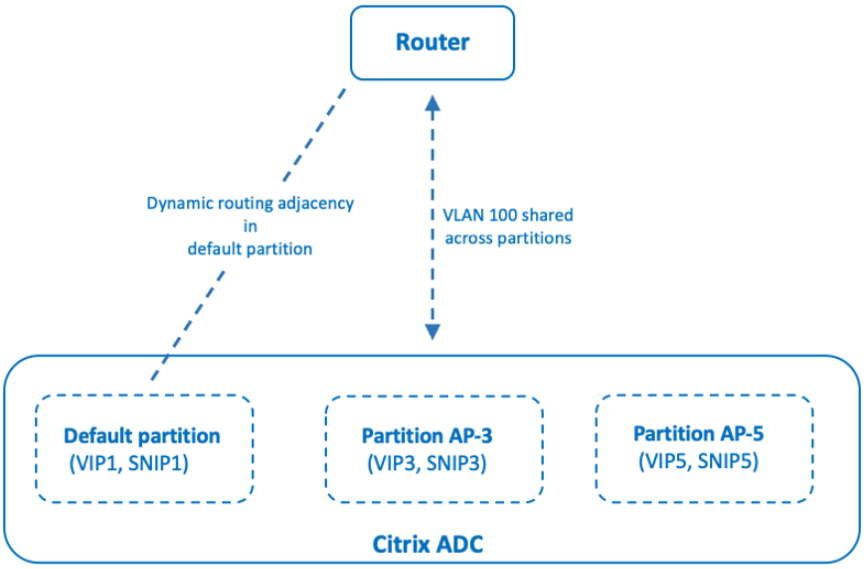
set ns partition P1  -partitionMAC 22:33:44:55:66:77
```

Bind partitions to a shared VLAN by using the CLI

```
bind partition <partition-id> -vlan <id>

bind partition <partition-id> -vlan <id>

bind partition P1  -vlan 100

bind partition P2  -vlan 100

```



```
bind partition P3 -vlan 100

bind partition P4 -vlan 100
```

Configure Shared VLAN by using the Citrix ADC GUI

1. Navigate to **Configuration > System > Network > VLANs** and then select a **VLAN** profile and click **Edit** to set the partition sharing parameter.
2. On the **Create VLAN** page, select the **Partitions Sharing** check box.
3. Click **OK** and then **Done**.

Dynamic routing over a shared VLAN across admin partitions

Admin partitions in a Citrix ADC appliance provide a way to host multiple tenants.

Starting from Citrix ADC version 12.1 build 51.16, a shared VLAN in a partitioned appliance supports the dynamic routing protocol. Routing can be configured in dedicated or shared VLANs associated with admin partitions.

Dedicated VLAN of an admin partition. In a dedicated VLAN, the data path for the tenant is identified using one or more VLANs. It results in strict configuration and data-path isolation for the tenant. For advertising the health of a VIP address, dynamic routing is enabled in each partition and the routing adjacency is established per partition.



Dynamic routing over a dedicated VLAN per partition

A shared VLAN across admin partitions. In a shared VLAN, VIP addresses configured in a non-default partition can be advertised through a single adjacency or peering formed in the default partition. A SNIP address in the non-default partition is used as the next-hop for all the VIP addresses (configured with **advertiseOnDefaultPartition** option) in that non-default partition. The configured SNIP address is marked as a next-hop IP address in the routing advertisements.

Consider an example setup of admin partitions in a Citrix ADC appliance, VLAN 100 is shared across the default partition, and non-default partitions: AP-3 and AP-5. SNIP addresses SNIP1 is added in the default partition, SNIP3 is added in AP-3, and SNIP5 is added in AP-5. SNIP1, SNIP3, and SNIP5 are reachable over the vlan-100. VIP addresses VIP1 is added in the default partition, VIP3 is added in AP-3, and VIP5 is added in AP-5. VIP3 and VIP5 are advertised through the single adjacency or peering formed in the default partition.



Dynamic routing over a shared VLAN across partitions

Before you begin

Before configuring dynamic routing over a shared VLAN in a non-default admin partition, make sure that:

- **Dynamic routing is configured on the shared VLAN in the default partition**. Configuring dynamic routing on the shared VLAN in the default partition consists of the following steps:
  1. Enable dynamic routing on the shared VLAN.
  2. Add a SNIP IP address with dynamic routing enabled. This SNIP IP address is used for dynamic routing with the upstream.
  3. Bind the SNIP IP subnet to the shared VLAN.

- **One or more dynamic routing protocol is configured on the default partition**. For more information, see configure dynamic routing protocols.

Configuration steps

Configuring dynamic routing over a shared VLAN in a non-default admin partition consists of the following steps:

1. **Add a SNIP IP address in the non-default partition**. This SNIP IP address must be in the same subnet of the SNIP IP address that is being used for dynamic routing in the default partition.
2. **Set or enable the following parameters for advertising a VIP address, in a non-default partition, using dynamic routing**.
    - Host route gateway (hostRtGw). Set this parameter to the SNIP address added in the preceding step.
    - Advertise on default partition (advertiseOnDefaultPartition). Enable this parameter.

Sample configuration

Consider an example of an admin partition setup in a Citrix ADC appliance. A non-default admin partition AP-3 is configured on this appliance. A shared VLAN VLAN100 is bound to AP-3. The following sample configuration configures dynamic routing, through VLAN100, in AP-3.

Steps	Sample configuration
On default admin partition	-
Enable dynamic routing on shared VLAN 100.	`set vlan 100 -dynamicRouting enabled`
Add SNIP IP address 192.0.2.10 with dynamic routing enabled. This SNIP IP address is used for dynamic routing with the upstream.	`add ns ip 192.0.2.10 255.255.255.0 -type SNIP -dynamicRouting enabled`
Bind subnet of 192.0.2.10 to shared VLAN 100.	`bind vlan 100 -IPAddress 192.0.2.10 255.255.255.0`
On non-default admin partition AP-3	-
Add SNIP IP address 192.0.2.30. This SNIP IP address is in the same subnet as the SNIP IP address 192.0.2.10 on the default partition.	`add ns ip 192.0.2.30 255.255.255.0 -type SNIP`

Steps	Sample configuration
For advertising VIP address 203.0.113.300 using dynamic routing, enable advertiseOnDefaultPartition parameter and set hostRtGw parameter to 192.0.2.30.	set ns ip 203.0.113.300 255.255.255.255 -hostRoute enabled -advertiseOnDefaultPartition enabled -hostRtGw 192.0.2.30

Dynamic routing of IPv6 over a shared VLAN across admin partition

The enable ns feature IPv6PT and set L3Param -ipv6DynamicRouting ENABLED commands must be enabled for an IPv6 address to dynamically route over a shared VLAN in an admin partition. The following sample configurations help you to configure dynamic routing of IPv6 over shared VLAN.

Sample configuration

The following sample configuration configures dynamic routing, through VLAN 100, in AP-3.

Steps	Sample configuration
On default admin partition	-
Enable dynamic routing on shared VLAN 100.	set vlan 100 -dynamicRouting enabled
Add SNIP IP address 2001:b:c:d::1/64 with dynamic routing enabled. The SNIP IP address is used for dynamic routing with the upstream.	add ns ip6 2001:b:c:d::1/64 -type SNIP -dynamicRouting enabled
Bind subnet of 2001:b:c:d::1/64 to shared VLAN 100.	bind vlan 100 -IPAddress 2001:b:c:d::1/64
On non-default admin partition AP-3	-
Add SNIP IP address 2001:b:c:d::2/64. This SNIP IP address is in the same subnet as the SNIP IP address 2001:b:c:d::2/64 on the default partition.	add ns ip6 2001:b:c:d::2/64 -type SNIP

Steps	Sample configuration
For advertising VIP address 2002::1/128 using dynamic routing, enable `advertiseOnDefaultPartition` parameter and set `ip6hostRtGw` parameter to 2001:b:c:d::2.	`set ns ip6 2002::1/128 -hostRoute enabled -advertiseOnDefaultPartition enabled -ip6hostRtGw 2001:b:c:d::2`

The VIP present in the admin partition must be seen on VTYSH of default partition as a kernel route.

```
> switch partition default
Done

>vtysh
ns#

ns# sh ipv6 route kernel

IPv6 routing table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
   IA - OSPF inter area, E1 - OSPF external type 1,
   E2 - OSPF external type 2, I - IS-IS, B - BGP
Timers: Uptime

K      2002::1/128 via 2001:b:c:d::2, vlan0, 01:24:15
                                >> on Default Partition, VIP : 2002::1
   present in AP known via SNIP6 : 2001:b:c:d::2 is present in AP as a
   Kernel Route
```

It can be advertised to upstream by using “redistribute kernel” option under OSPFv3/BGP+ in default partition.

```
ns# sh run router ipv6 ospf
!
router ipv6 ospf 1
redistribute kernel
!
```

Shared VLAN with admin partition on Citrix ADC SDX appliance

On the SDX appliance, you must generate and configure the PMAC address by using the Management Service user interface, before using the admin partitions with shared VLANs. Management Service

enables you to generate partition MAC addresses by:

- Using a base MAC address
- Specifying custom MAC addresses
- Randomly generating MAC addresses

> **Note**
>
> - The randomly generating MAC addresses are used for other deployments other than high availability.
> - After generating the partition MAC addresses, you must restart the Citrix ADC instance before configuring the admin partitions. For more information on generating partition MAC addresses from the SDX appliance, see Generating Partition MAC Addresses to Configure Admin Partition on a Citrix ADC instance in the SDX Appliance.

VXLAN support for admin partitions

September 14, 2021

In a partitioned Citrix ADC appliance, similar to configuring a VLAN, you can configure a VXLAN in the default partition. After configuring a VXLAN, you can bind it to an administrative partition or if a VXLAN is extending a VLAN that is bound to a partition, the appliance binds the VXLAN to the partition under the same broadcast domain. It is applicable in unbinding a VLAN that unbinds a VXLAN from the partition.

For more information about how VXLAN works in a Citrix ADC appliance, see VXLAN.

Also, for more information on how VLAN works in a partitioned Citrix ADC appliance, see Admin Partitioning.

Points to remember before configuring a VXLAN

Remember the following points before you configure a VXLAN in a partitioned Citrix ADC appliance:

- When you extend a VLAN over VXLAN, make sure VLAN is bound to the partition.
- Only a partition administrator must configure the IP and dynamic routing for the VXAN in the administrative partition.

A shared VXLAN is not supported in a partitioned appliance and so a VXLAN cannot be tagged to a shared VLAN or you cannot make a VLAN a shared one when it is tagged to a VXLAN.

Supportable VXLAN configurations

Following are the supportable VXLAN configurations.

Extending VLAN over a VXLAN in the same broadcast domain

The following CLI steps help you to extend a VLAN over a VXLAN and the opposite way within the same broadcast domain.

1. Add a VLAN in the default partition

```
add vlan <id>
```

2. Extend VLAN over a VXLAN within the same broadcast domain.

```
add vxlan <vxlan id> -vlan <id>
```

3. Configure a peer `vtep` to carry all BUM (broadcast unknown multicast) traffic.

> **Note**
>
> The `vtep` address can be a multicast address.

```
add bridgetable -mac <mac_addr> -vxlan <positive_integer> -vtep <
    ip_addr> [-vni <positive_integer>][-deviceVlan <
    positive_integer>]
```

4. Bind IP addresses to VXLAN.

```
bind vxlan <id> [-srcIP <ip_addr>][-IPAddress <ip_addr|ipv6_addr
    |*> [<netmask>]]
```

5. Bind VLAN to an administrative partition.

```
bind partition <partition-id> -vxlan <id>

add vlan 3000

add vxlan 3000 -vlan 10

add bridgetable -mac 00:00:00:00:00:00 -vxlan 3000 -vtep
    10.102.58.8 -vni 11

bind vxlan 3000 - srcIP 10.102.101.15

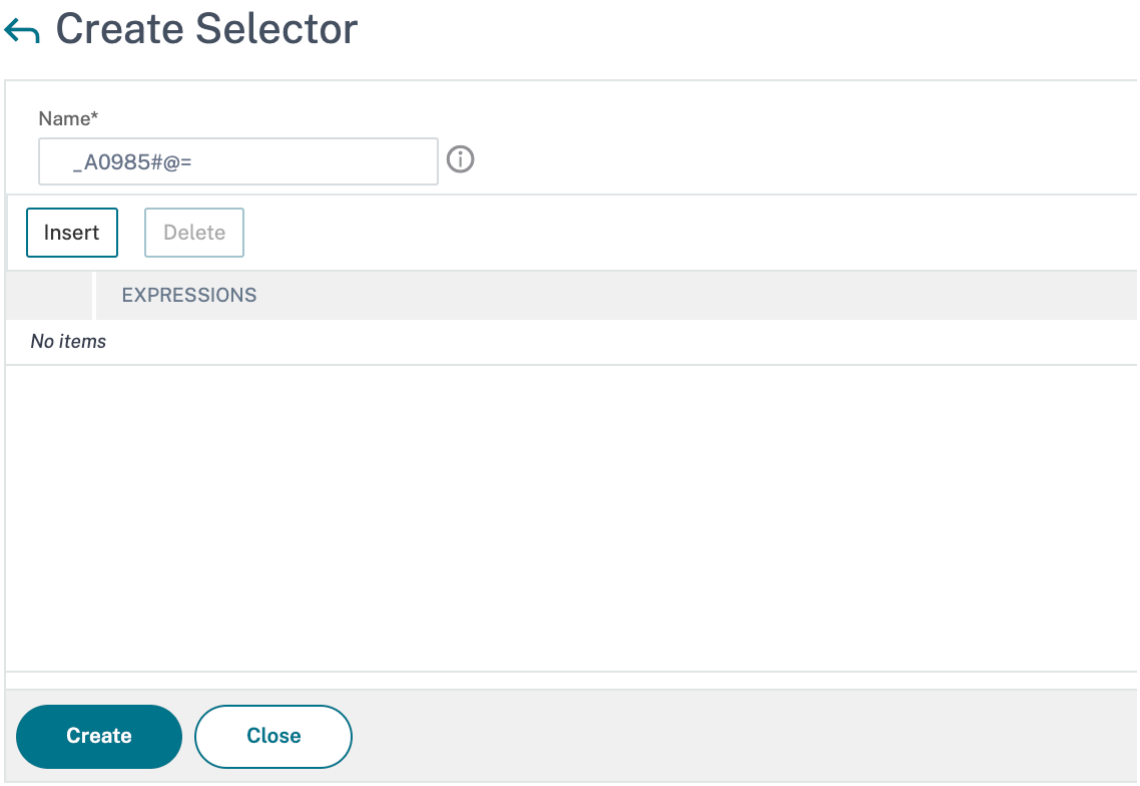
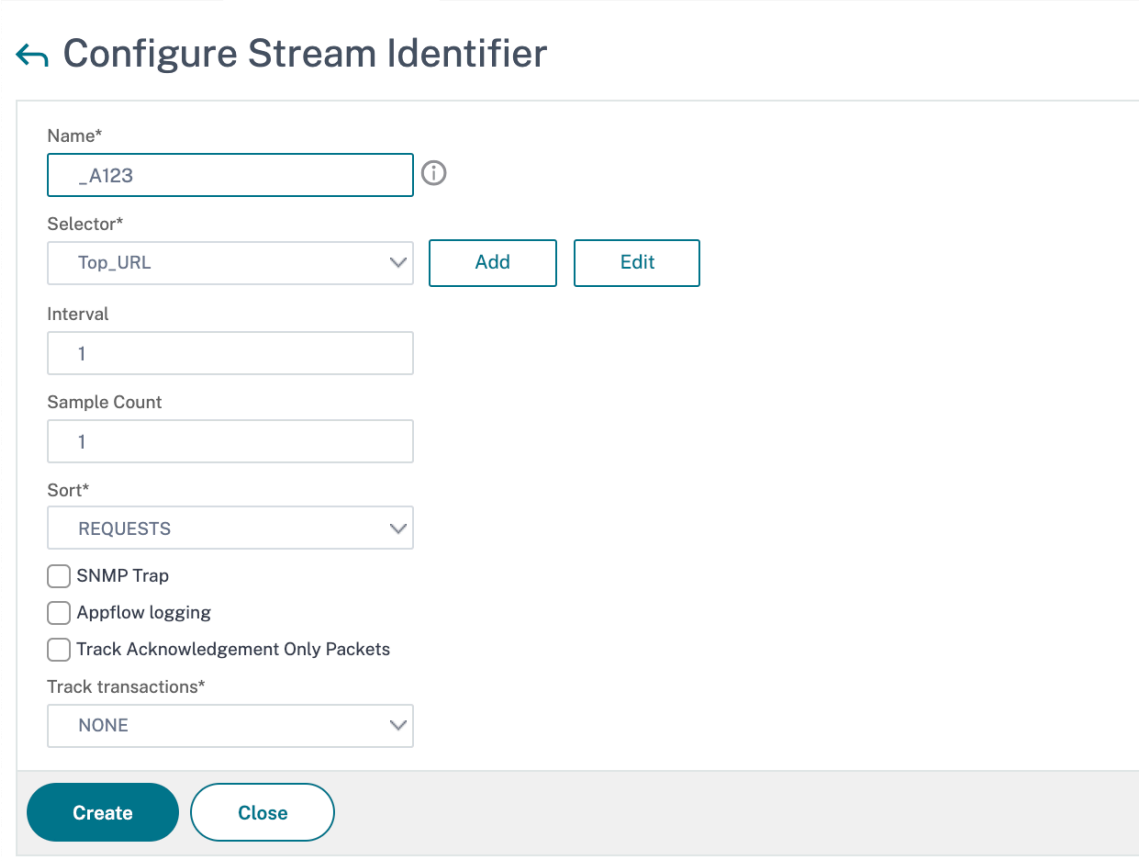
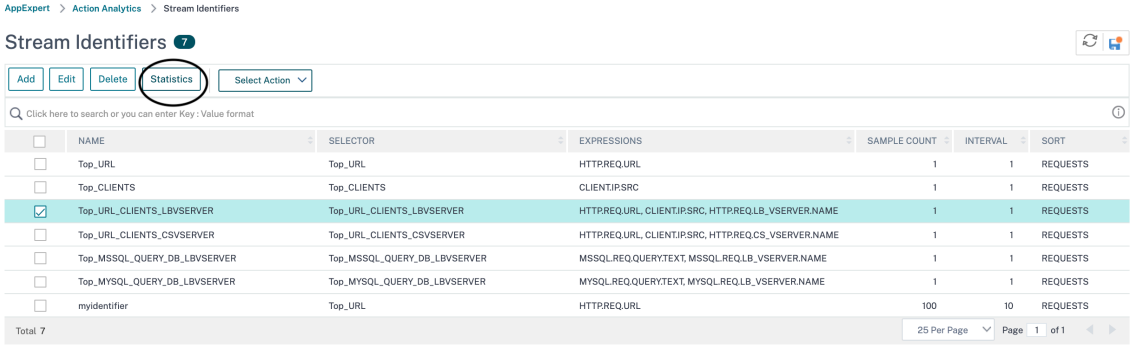
bind partition p1 -vlan 10
```


SNMP support for admin partitions

September 14, 2021

A partitioned Citrix ADC appliance uses the SNMP infrastructure for partition rate limiting and for monitoring partition resource utilization details.

SNMP traps for admin partition rate limiting

On a partitioned Citrix ADC appliance, a PARTITION-RATE-LIMIT alarm can generate nine SNMP traps for notifying a partition resource (such as bandwidth, connection, or memory) has reached its limit or returned to normal.

The following nine SNMP traps are generated when:

- **partitionCONNThresholdReached**. The number of active connections for a partition exceeds its high threshold percentage.
- **partitionCONNThresholdNormal**. The number of active connections is less than or equal to the normal threshold percentage.
- **partitionBWThresholdReached**. The partition's bandwidth usage reaches its high threshold percentage.
- **partitionMEMThresholdReached**. The current memory usage of the partition exceeds its high threshold percentage.
- **partitionMEMThresholdNormal**. The current memory usage of the partition becomes less than or equal to the normal threshold percentage.
- **partitionMEMLimitExceeded.** The current memory usage of the partition exceeds its memory limit percentage.
- **partitionCONNLimitExceeded**. The number of active connections for a partition exceeds its configured limit and new connections are being dropped.
- **partitionCONNLimitNormal**. The number of active connections for a partition goes below its configured limit and the partition can now accept a new connection.
- **partitionBWLimitExceeded**. The current bandwidth usage for a partition has exceeded its configured limit.

The threshold values for the SNMP traps are non-configurable and are the following:

- High threshold = 80% (applicable for all partition rate limit traps)
- Low threshold = 60% (applicable for all partition rate limit traps)
- Memory limit = 95% (applicable only for partition memory traps)

Configuring PARTITION-RATE-LIMIT alarm

To configure the PARTITION-RATE-LIMIT alarm in a specific partition and enable generation of the SNMP trap messages.

1. Enable PARTITION-RATE-LIMIT Alarm
2. Configure PARTITION-RATE-LIMIT Alarm
3. Configure SNMP Trap Destination

To enable PARTITION-RATE-LIMIT alarm by using the CLI

At the command prompt, type the following commands:

```
enable snmp alarm PARTITION-RATE-LIMIT

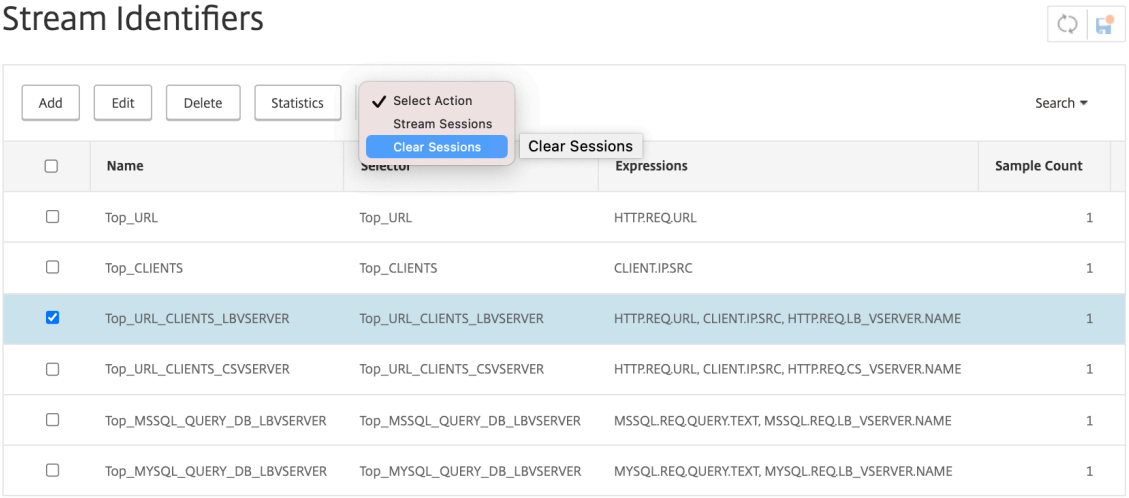
show snmp alarm PARTITION-RATE-LIMIT
```

To configure PARTITION-RATE-LIMIT Alarm by using the CLI

At the command prompt, type the following command:

```
set snmp alarm PARTITION-RATE-LIMIT [-state ( ENABLED | DISABLED )] [-
    severity <severity>] [-logging ( ENABLED | DISABLED )]
```

To configure the SNMP trap destination by using the CLI

At the command prompt, type the following command:

```
add snmp trap <trapClass> <trapDestination> [-version <version>] [-td <
    positive_integer>] [-destPort <port>] [-communityName <string>] [-
    srcIP <ip_addr|ipv6_addr>] [-severity <severity>] [-allPartitions (
    ENABLED | DISABLED )]
```

To configure the PARTITION-Rate-Limit alarm by using the GUI

Navigate to **System > SNMP > Alarms**, select **PARTITION-RATE-LIMIT** alarm, and configure the alarm parameters.

To configure the SNMP trap destination by using the GUI

Navigate to **System > SNMP > Trap**, specify the IP address of the destination device.

SNMP monitoring for partition resource utilization

Using SNMP, you can monitor a partition's resource (such as bandwidth, connection, and memory) utilization details at real time on a Citrix ADC appliance. It is done by sending an SNMP request (such as SNMP GET, SNMP GET BULK, SNMP GETNEXT, or SNMP WALK) from the SNMP Manager.

> **Note**
>
> To monitor partition resources, you must configure the SNMP community in the default partition. Wherein, the *partitionTable* is maintained in the default partition, and the SNMP communication is done through the NSIP address of the appliance.

Consider a scenario where a Citrix ADC administrator wants to know the bandwidth usage of partition P1 on the appliance. The SNMP Manager retrieves this information by sending an SNMP GET request on the corresponding OID (partitionCurrentBandwidth) to the NSIP address of the appliance. The SNMP agent on the default partition retrieves and sends the current bandwidth usage of P1 to the SNMP Manager through the NSIP address.

The following table lists the SNMP counters which are part of *partitionTable* and its description:

SNMP Parameter	SNMP OID	Description
partitionName	1.3.6.1.4.1.5951.4.1.1.88.1.1	Partition name
partitionCurrentBandwidth	1.3.6.1.4.1.5951.4.1.1.88.1.2	Current bandwidth usage of the partition.
partitionCurrentConnections	1.3.6.1.4.1.5951.4.1.1.88.1.3	Current number of active connections of the partition.
partitionMemoryUsagePcnt	1.3.6.1.4.1.5951.4.1.1.88.1.4	Current Memory usage (in percentage) of the partition.

Audit log support for admin partitions

September 14, 2021

On a partitioned Citrix ADC appliance, for enhanced data security, you can configure audit logging in an administrative partition by using advanced policies. For example, you might want to view logs (states and status information) of a specific partition. It has multiple users accessing different sets of features based on their levels of authorization in the partition.

Points to remember

1. The audit logs generated from the partition is stored as a single log file (/var/log/ns.log).
2. Configure the audit log server's (syslog or ns log) subnet address as the source IP address in the partition for sending the audit-log messages.
3. The default partition uses the NSIP as the source IP address for the audit log messages by default.
4. You can display the audit-log message by using the "show audit messages" command.

For information on audit-log configuration, see Configuring the NetScaler Appliance for Audit Logging.

Configuring audit logging in partitioned Citrix ADC appliance

Complete the following tasks to configure audit logging in an administrative partition.

1. Configure partition subnet IP address. An IPv4 SNIP address of an administrative partition.
2. Configure audit-log (syslog and ns log) action. An Audit action is a collection of information that specifies the messages to be logged and how to log the messages on the external log server.
3. Configure audit-log (syslog and ns log) policies. Audit-log policies define log messages for the source partition to the syslog or ns log server.
4. Bind audit-log policy to sysGlobal and nsGlobal entity. Bind an audit-log policy to a system global entity.
5. Review audit-log statistics. Display the audit-log statistics and evaluate the configuration.

Configure the following by using the CLI

1. Create a partition's subnet IP address

   ```
   add ns ip <ip address> <subnet mask>
   ```

2. Create a syslog action

   ```
   add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )] [-transport ( TCP |
   UDP )]
   ```

3. Create an ns log action

   ```
   add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
   <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )]
   ```

4. Create a syslog audit-log policies

   ```
   add audit syslogpolicy syslog-pol1 true audit-action1
   ```

5. Create an ns log audit-log policies

   ```
   add audit nslogpolicy nslog-pol1 true audit-action1
   ```


6. Bind an audit-log policy to syslogGlobal entity

   ```
   bind audit sysloggglobal -policyName <name> -priority <priority_integer>
    -globalBindType SYSTEM_GLOBAL
   ```

7. Bind an audit-log policy to nslogGlobal entity

   ```
   bind audit nslogglobal -policyName <name> -priority <priority_integer>
   -globalBindType SYSTEM_GLOBAL
   ```

8. Display an audit-log statistics

   ```
   stat audit -detail
   ```

Example

```
1 add ns ip 10.102.1.1 255.255.255.0
2 add audit syslogAction syslog_action1 10.102.1.2 -logLevel
     INFORMATIONAL -dateFormat MMDDYYYY -transport UDP
3 add audit syslogpolicy syslog-pol1 true syslog_action1
4 bind audit sysloggglobal -policyName syslog-pol1 -priority 1 -
     globalBindType SYSTEM_GLOBAL
```

Storing logs

When the SYSLOG or NSLOG server collects log information from all partitions, it is stored as log messages in the ns.log file. The log messages contain the following information:

- Partition Name.
- The IP address.
- A time stamp.
- The message type
- The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency)
- The message information.

Display configured PMAC addresses for shared VLAN configuration

September 14, 2021

To use a partition setup with shared VLAN configuration, you need a virtual MAC address called as partition MAC (PMAC) address. The partition uses the PMAC address for its communication on the shared VLAN. A unique PMAC address is configured for each partition and it is used across all shared

VLANs bound to that partition. In the case of a non-SDX platform (VPX or MPX) platform, the PMAC address can be either user specified or internally generated by a Citrix ADC appliance. If the PMAC address is not specified for a partition, it is internally generated when the partition is bound to the first shared VLAN. While in the case of an SDX platform, the PMAC addresses always need to be configured from the SVM tool first and then assigned to a partition.

To display a list of configured PMACs, you can use the **Show ns PartitionMAC** command. The command enables you to verify the configured PMACs either through the Citrix ADC CLI or GUI. The command displays all the PMAC addresses and the corresponding partitions (if assigned). In the case of a non-SDX platform, the command displays all the PMAC addresses and their corresponding partitions because the PMAC address is assigned to a partition only on need basis (when a partition bound a shared VLAN). However in the case of an SDX platform, you might have some unassigned PMACs in the list.

For information on how to generate PMAC for the SDX platform, see Generating Partition MAC Addresses topic.

Display PMACs by using the Citrix ADC CLI

At the command prompt, type the following command:

```
show ns partitionMAC
```

```
Partition MAC Partition Name

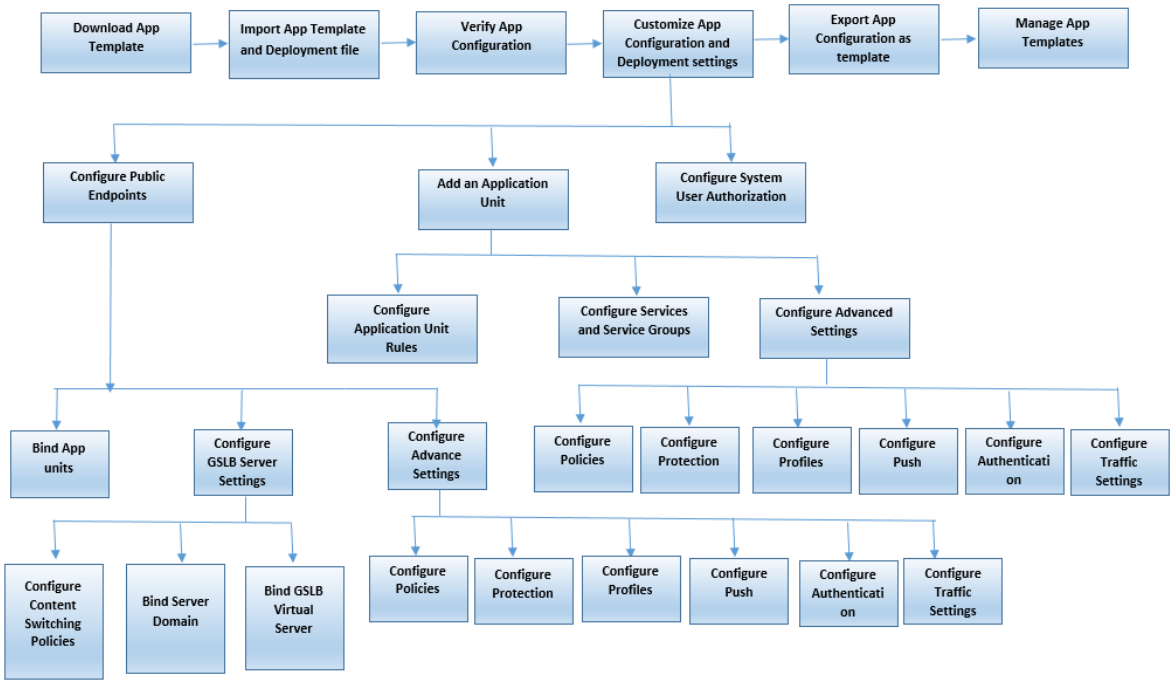
1) f2:0c:64:da:f6:d7

2) b4:0c:43:da:f6:d2

3) a6:e7:b2:6c:48:e0

Done
```

Display PMAC Addresses by using the Citrix ADC GUI

1. Sign in to the Citrix ADC appliance and navigate to the **Configuration > System > Partition MAC**.
2. The Partition MAC page displays a list of PMACs and its partitions.

AppExpert

September 14, 2021

The following topics provide a conceptual reference and configuration instructions for the AppExpert and other features of the Citrix ADC appliance.

> **Note**
>
> For information about policy extensions, see Policy Extensions.

- Action Analytics: Collects run-time statistics on the basis of pre-defined criteria. When used with policies, the feature also provides you with the infrastructure for automatic, real-time traffic optimization.
- AppExpert Applications and Templates: Simplify configuration steps for the Citrix® NetScaler® appliance by using applications, application templates, Citrix Gateway applications, and entity templates.
- AppQoE: Application level Quality of Experience (AppQoE) integrates several existing policy-based security features of the Citrix ADC appliance into a single integrated feature that takes advantage of a new queuing mechanism, fair queuing.
- Entity Template: Describes how to use entity templates to set up and configure individual Citrix ADC entities, such as a policy or virtual server. An entity template provides a specification and a set of defaults for the object.
- HTTP Callouts: An HTTP request that the Citrix ADC appliance generates and sends to an external application when certain criteria are met during policy evaluation.
- Pattern Sets: Allow string matching during the evaluation of a default syntax policy.
- Policies and Expressions: Rules that determine the operations that the Citrix ADC appliance must perform.
- Rate Limiting: Defines the maximum load for a given network entity or virtual entity on the Citrix ADC appliance.
- Responder: Bases responses on who sends the request, where it is sent from, and other criteria with security and system management implications.
- Rewrite: Rewrites information in the requests or responses handled by the Citrix ADC appliance.
- String Maps: Perform pattern matching in all Citrix ADC features that use the default policy syntax.

Action analytics

September 14, 2021

The performance of your website or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources help you determine whether those resources need to be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

If the website or application does not change frequently, you can use products that collect statistical data, and then manually analyze the statistics and optimize the delivery of content. However, if you do not want to perform manual optimizations, or if your website or application is dynamic in nature, you need infrastructure that can not only collect statistical data but can also automatically optimize the delivery of resources on the basis of the statistics. On the Citrix ADC appliance, this functionality is provided by the action analytics feature. The feature operates on a single Citrix ADC appliance and collects run-time statistics on the basis of criteria that you define. When used with Citrix ADC policies, the feature also provides you with the infrastructure that you need for automatic, real-time traffic optimization.

When configuring the action analytics feature, you specify the request attributes for which you want to collect statistical data, for example, URLs and HTTP methods by configuring default syntax expressions in an entity called a selector. Then, you configure an identifier to configure settings such as the sampling interval and sample count. You also configure a policy that enables the appliance to evaluate traffic as specified by the selector-identifier pair. Finally, you bind the policy to a bind point to begin collecting statistics.

The appliance also provides you with a set of built-in selectors, identifiers, and responder policies that you can use to get started with the feature.

The appliance aggregates the following statistics:

- The number of requests.
- The bandwidth consumed by the requests.
- The response time.
- The number of concurrent connections.

You can configure the feature to perform run-time sorting of the records on an attribute of your choice.

You can view the statistical data by using either the command-line interface or the Stream Sessions tool in the configuration utility.

Configure a selector

September 14, 2021

A selector is a filter for identifying requests. It consists of up to five individual default syntax expressions that identify request attributes such as the client IP address and the URL in the request. Each expression is a non-compound default syntax expression and is considered to be in an AND relationship with the other expressions. Following are some examples of selector expressions:

- `HTTP.REQ.URL`
- `CLIENT.IP.SRC`
- `HTTP.RES.BODY(1000).AFTER_STR("<string>").BEFORE_STR("<string>")"`
- `CLIENT.IP.SRC.SUBNET(24)`

Selectors are used in rate limiting and action analytics configurations. A selector is optional in a rate limiting configuration but is required in an action analytics configuration.

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the Citrix ADC considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the Citrix ADC, again, can count the same transaction more than once.

If you modify an expression in a selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a selector named myLimitSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myLimitSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

The Citrix ADC appliance provides built-in selectors pdf for some of the most common use cases. Refer to the pdf.

You can also configure a selector with expressions that identify the request attributes of your choice. For example, you might want to create a record for a request that arrives with a specific header. To evaluate the header, you can add `HTTP.REQ.HEADER("<header_name>")` to the selector that you intend to use.

To configure a selector by using the command line interface:

At the command prompt, type the following commands to configure a selector and verify the configuration:

- `add stream selector <name> <rule> ...`
- `show stream selector`

Example

```
> add stream selector myselector HTTP.REQ.URL CLIENT.IP.SRC
 Done
> show stream selector myselector
    Name: myselector
    Expressions:
        1) HTTP.REQ.URL
        2) CLIENT.IP.SRC
 Done
>
<!--NeedCopy-->
```

To modify or remove a selector by using the command line interface:

- To modify a selector, type the set stream selector command, the name of the selector, and the rule parameter with the expressions. Enter the existing expressions that you want to retain, along with the new expressions that you want to add.
- To remove a selector, type the rm stream selector command and the name of the selector.

To configure a selector by using the GUI:

1. Navigate to **AppExpert** > **Action Analytics** > **Selectors**.
2. In the details pane, do one of the following:
   - To create a selector, click **Add**.
   - To modify a selector, select the selector, and then click **Edit**.
3. In the **Create Selector** or **Configure Selector** page, set the following parameters:
   - Name. To add a name for the selector,enter the name in the **Name** filed. The Name must begin with ASCII, alphanumeric or underscore character. The name must contain only ASCII alphanumeric, underscore, hash, period, space, colon, at, equals, and hyphen characters.
   - Expressions. To add the expression to the selector configuration, click **Insert**. To remove an expression from the selector configuration, in the Expression box, select the expression, and then click **Delete**. Note: In the Expressions box, enter a valid parameter. For example, enter HTTP. Then, enter a period after this parameter. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. To select the next keyword in this expression prefix, double-click the selec-

tion in the drop-down menu. The **Expressions** text box displays both the first and second keywords for the expression prefix, for example, HTTP.REQ. Continue adding expression components until the complete expression is formed.

4. Click **Insert**.
5. Continue adding up to five non-compound expressions.
6. Click **Create** and then **Close**.



Configure a stream identifier

September 14, 2021

You configure a stream identifier to specify parameters for collecting statistical data from requests identified by a given selector. An identifier specifies the selector to be used, the statistics collection interval, the sample count, and the field on which the records are to be sorted.

The Citrix ADC appliance includes the following built-in stream identifiers for common use cases. All the built-in identifiers specify a sample count of 1 and an interval of 1 minute. Additionally, they sort the data on the
REQUESTS attribute. They differ only in being associated with different built-in selectors. Each built-in identifier is associated with a built-in selector of the same name (for example, the built in identifier

Top_URL is associated with the built-in selector
Top_URL). Following are the built-in identifiers:

- Top_URL
- Top_CLIENTS
- Top_URL_CLIENTS_LBVSERVER
- Top_URL_CLIENTS_CSVSERVER
- Top_MSSQL_QUERY_DB_LBVSERVER
- Top_MYSQL_QUERY_DB_LBVSERVER

For more information about the built-in selectors, see Configuring a Selector.

Note: The maximum length for storing string results of selectors (for example, HTTP.REQ.URL) is 60 characters. If the string (for example, URL) is 1000 characters long, of which 50 characters are enough to uniquely identify a string, use an expression to extract only the required 50 characters.

You cannot modify a built-in identifier's configuration. However, you can create an identifier with a configuration of your choice.

To configure a stream identifier by using the command line interface

At the command prompt, type the following commands to configure a stream identifier and verify the configuration:

- `add stream identifier <name> <selectorName> [-interval <positive_integer>] [-SampleCount <positive_integer>] [-sort <sort>]`
- `show stream identifier <name>`

Example

```
1 > add stream identifier myidentifier Top_URL -interval 10 -sampleCount
     100
2  Done
3 <!--NeedCopy-->
```

To configure a stream identifier by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.
2. In the details pane, do one of the following:
   - To create a stream identifier, click **Add**.
   - To modify a stream identifier, select the identifier, and then click **Edit**.
3. In the Configure Stream Identifier page, set the following parameters:
   - Name
   - Selector
   - Interval

- Sample Count
- Sort

4. Click **Create**, and then click **Close**.



View statistics

September 14, 2021

You can view the collected statistics in tabular format in the command-line interface and in graphical format in the configuration utility.

The following table describes the collected statistics:

Statistics	Column name in the output of the stat stream identifier <identifier name> command	Description
Number of requests	**Req**	The number of requests for which records were created in the last <interval> number of minutes.
Bandwidth consumed	**BandW**	The total bandwidth consumed by the requests that were received in the last <interval> number of minutes. The total bandwidth of a request is the bandwidth consumed by the request and its response. The value is rounded off to the next higher or next lower integer value. So, it might differ slightly from the expected value. For example, if a request's total bandwidth consumption is 2.2 KB. One instance of the request might be shown as having consumed 2 KB. Two instances might be shown as having consumed 4 KB, but three instances might be shown as having consumed 7 KB.
Response time	**RspTime**	The average response time for all the requests received in the last <interval> number of minutes.
Concurrent connections	**Conn**	The total number of concurrent connections that are currently open.

To view the statistical data collected for a stream identifier by using the command line

At the command prompt, type:

```
stat stream identifier <name> [<pattern> ...] [-detail] [-fullValues] [-
ntimes <positive_integer>] [-logFile <input_filename>] [-sortBy <sortBy> [<
sortOrder>]
```

Examples

Example 1 sorts the output on the BandW column, in the descending order. Example 2 sorts the output in Example 1, on the **Req** column, and in the ascending order

Example 1

```
1  > stat stream identifier myidentifier -sortBy BandW Descending -
       fullValues
2  Stream Session statistics
3                       Req                 BandW
4  User1                508                125924
5  User2               5020                 12692
6  User3               2025                  4316
7
8                   RspTime                  Conn
9  User1               5694                     0
10 User2                109                     0
11 User3                  3                     0
12  Done
13 <!--NeedCopy-->
```

Example 2

```
1  > stat stream identifier myidentifier -sortBy Req Ascending  -
       fullValues
2  Stream Session statistics
3                       Req                 BandW
4  User1                508                125924
5  User3               2025                  4316
6  User2               5020                 12692
7
8                   RspTime                  Conn
9  User1               5694                     0
10 User3                  3                     0
11 User2                109                     0
12  Done
13 <!--NeedCopy-->
```


To view the statistical data collected for a stream identifier by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.
2. Select the stream identifier whose sessions you want to view, and then click Statistics For information about how you can group the output on the basis of the values collected for various selector expressions.



Grouping records on attribute values

September 14, 2021

Statistical information such as the number of times a particular URL has been accessed overall and per client, and the total number of GET and POST requests per client can provide valuable insights into whether any of your resources need to be expanded to meet the demand or be optimized for delivery. To obtain such statistics, you must use an appropriate set of selector expressions, and then use the pattern parameter in the stat stream identifier command. The grouping is based on the pattern that is specified in the command. Grouping can be performed concurrently on the values of multiple expressions.

In the command-line interface, you can group the output by using patterns of your choice. In the configuration utility, the pattern depends on the choices you make when drilling down through the values of various selector expressions. For example, consider a selector that has the expressions `HTTP.REQ.URL`, `CLIENT.IP.SRC`, and `HTTP.REQ.LB_VSERVER.NAME`, in that order. The statistics home page displays icons for each of these expressions. If you click the icon for `CLIENT.IP.SRC`, the output is based on the patterns *?. The output displays statistics for each client IP address. If you click an IP address, the output is based on the patterns * <IP address> ? and ? <IP address> * where <IP address> is the IP address you selected. In the resulting output, if you click a URL, the pattern used is <URL> <IP address> ?.

To group the records on the values of selector expressions by using the command line interface

At the command prompt, enter the following command to group the records on the basis of a selector expression:

```
stat stream identifier <name> [<pattern> ...]
```

The following examples use a different pattern to demonstrate the effect of the pattern on the output of the stat stream identifier command. The selector expressions are HTTP.REQ.URL and HTTP.REQ.HEADER("UserHeader"), in that order. The requests contain a custom header whose name is UserHeader. Note that in the examples, a given statistical value changes as determined by the grouping, but the sum total of the values for a given field remains the same.

Example 1

In the following command, the pattern used is ? ?. The appliance groups the output on the values collected for both selector expressions. The row headers consist of the expression values separated by a question mark (?). The row with the header /mysite/mypage1.html?Ed displays statistics for requests made by user Ed for the URL /mysite/mypage1.html.

> **Note:**
>
> You must ensure to type the following command with "\?" instead of "?". For example, If selector uses an expression - client.ip.src and client.tcp.srcport. The Stat command to group the output on the values collected for the selector is 'stat stream identifier myidentifier \? \? -fullValues' as given below.

```
> stat stream identifier myidentifier ? ? -fullValues
Stream Session statistics
                                                Req                  BandW
/mysite/mypage2.html?Grace                        1                   2553
/mysite/mypage1.html?Grace                        2                      4
/mysite/mypage1.html?Ed                           8                     16
/mysite/mypage2.html?Joe                          1                   2554
/mysite/mypage1.html?Joe                          5                     10
/mysite/?Joe                                      1                      4

                                             RspTime                  Conn
/mysite/mypage2.html?Grace                        0                      0
/mysite/mypage1.html?Grace                        0                      0
/mysite/mypage1.html?Ed                           0                      0
/mysite/mypage2.html?Joe                          0                      0
/mysite/mypage1.html?Joe                          0                      0
/mysite/?Joe                                      6                      0
 Done
```



```
19 <!--NeedCopy-->
```

Example 2

In the following command, the pattern used is * ?. The appliance groups the output on the values accumulated for the second expression HTTP.REQ.HEADER(“UserHeader”). The rows display statistics for all requests made by users Grace, Ed, and Joe.

> **Note:**
>
> Ensure to type the following command with “\?” instead of “?”.

```
1 > stat stream identifier myidentifier * ?
2 Stream Session statistics
3                   Req    BandW  RspTime     Conn
4 Grace               3     2557        0        0
5 Ed                  8       16        0        0
6 Joe                 7     2568        6        0
7  Done
8 <!--NeedCopy-->
```

Example 3

In the following command, the pattern used is ? *, which is the default pattern. The output is grouped on the values collected for the first selector expression. Each row displays statistics for one URL.

> **Note:**
>
> Ensure to type the following command with “\?” instead of “?”.

```
1  > stat stream identifier myidentifier ? * -fullValues
2  Stream Session statistics
3                                          Req                 BandW
4  /mysite/mypage2.html                      2                  5107
5  /mysite/mypage1.html                     15                    30
6  /mysite/                                  1                     4
7
8                                       RspTime                  Conn
9  /mysite/mypage2.html                      0                     0
10 /mysite/mypage1.html                      0                     0
11 /mysite/                                  6                     0
12  Done
13 <!--NeedCopy-->
```

Example 4

In the following command, the pattern used is * *. The appliance displays one set of collective statistics for all the requests received, with no row title.


```
> stat stream identifier myidentifier * *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   18     5141        6        0
 Done
<!--NeedCopy-->
```

Example 5

In the following command, the pattern is /mysite/mypage1.html *. The appliance displays one set of collective statistics for all the requests received for the URL /mysite/mypage1.html, with no row title.

```
> stat stream identifier myidentifier /mysite/mypage1.html *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   15       30        0        0
 Done
<!--NeedCopy-->
```

Clearing a stream session

September 14, 2021

You can flush all the records that have been accumulated for a stream identifier.

To clear a stream session by using the command line interface

At the command prompt, enter the following commands to clear a stream session and verify the results:

- clear stream session
- stat stream identifier

Example

This example uses the stat stream identifier command first, so that a comparison can be made with the stat stream identifier command that is used for verifying the result of the clear stream session command.

```
>stat stream identifier myidentifier
Stream Session statistics
```



```
                  Req     BandW  RspTime      Conn
/aed....html        2         0         0         0
/                 636       303        12         0
 Done
>clear stream session myidentifier
 Done
>stat stream identifier myidentifier
 Done
<!--NeedCopy-->
```

To clear a stream session by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.
2. Select the stream identifier whose sessions you want to clear, and then click **Clear Sessions**.



Configure policy for optimizing traffic

September 14, 2021

To put the selector-identifier pair in your action analytics configuration into effect, you must associate the pair with the point in the traffic flow at which you want to collect statistics. You can do so by configuring a default syntax policy and referencing the stream identifier from the policy rule. You can use compression policies, caching policies, rewrite policies, application firewall policies, responder policies, and any other policies whose action is based on a Boolean expression.

The action analytics feature introduces a set of default syntax expressions and functions for collecting and evaluating data. The expression `ANALYTICS.STREAM(<identifier_name>)` is used for referencing the identifier that you want to use. The expression COLLECT_STATS is used to collect statistical

data. Functions such as `IS_TOP(<uint>)` and `IS_TOP_FREQUENTS(<uint>)` are used for making automatic, real-time traffic optimization decisions.

- **IS_TOP(<number>).** Finds if a given object is in the top <number> of elements. For example, is the element among the top 10 elements. When multiple elements have the count, they are considered to be similar in nature. The sort function must be turned on to avoid an undef condition.
- **IS_TOP_FREQUENTS(<frequency>).** Finds if a given object is in the top <frequency> of the elements that are in the top elements. For example, is the element among the top 50% of all the top elements maintained. Elements having the same values are considered similar in nature. The sort function must be turned on to avoid an undef condition.

It is your policy configuration that determines whether the Citrix ADC appliance must only collect data from traffic or also perform an action. If the appliance must only collect statistical data, you can configure a policy with the rule `ANALYTICS.STREAM(<identifier_name>).COLLECT_STATS` and the action NOOP. The NOOP policy must be the policy with the highest priority at the bind point. This policy is sufficient if you are only collecting statistics. Traffic optimization decisions, such as what to compress or cache, must be based on manual, periodic evaluation of the statistical data.

If, in addition to collecting statistics, the appliance must also perform an action on the traffic, you must configure the gotoPriorityExpression parameter of the NOOP policy such that another policy that has the desired rule and action is evaluated subsequently. This second policy must have a rule that begins with the `ANALYTICS.STREAM(<identifier_name>)` prefix and a function that evaluates the data.

Following is an example of two responder policies that are configured and bound globally. The policy responder_stat_collection enables the appliance to collect statistics based on the identifier, myidentifier. The policy responder_notify evaluates the data that is collected.

Example

```
1  > add responder action send_notification respondwith '"You are in the
       Top 10 list for bandwidth consumption"'
2   Done
3  > add responder policy responder_stat_collection' ANALYTICS.STREAM("
       myidentifier").COLLECT_STATS' NOOP
4  Done
5  > add responder policy responder_notify 'ANALYTICS.STREAM("myidentifier
       ").BANDWIDTH.IS_TOP(10)' send_notification
6   Done
7  > bind responder global responder_stat_collection 10 NEXT
8   Done
9  > bind responder global responder_notify 20 END
10  Done
```



```
<!--NeedCopy-->
```

How to limit bandwidth consumption per user or client device

September 14, 2021

Your web site, application, or file hosting service has finite network and server resources available to it to serve all its users. One of the most important resources is bandwidth. Substantial bandwidth consumption by only a subset of the user base can result in network congestion and reduced resource availability to other users. To prevent network congestion, you might have to limit a client's bandwidth consumption by using temporary service denial techniques such as responding to a client request with an HTML page if it has exceeded a preconfigured bandwidth value over a fixed time period leading up to the request.

In general, you can regulate bandwidth consumption either per client device or per user. This use case demonstrates how you can limit bandwidth consumption per client to 100 MB over a time period of one hour. The use case also demonstrates how you can regulate bandwidth consumption per user to 100 MB over a time period of one hour, by using a custom header that provides the user name. In both cases, the tracking of bandwidth consumption over a moving time period of one hour is achieved by setting the interval parameter in the stream identifier to 60 minutes. The use cases also demonstrate how you can import an HTML page to send to a client that has exceeded the limit. Importing an HTML page not only simplifies the configuration of the responder action in these use cases, but also simplifies the configuration of all responder actions that need the same response.

To limit bandwidth consumption per user or client device by using the command line interface

In the command-line interface, perform the following tasks to configure action analytics for limiting a client's or user's bandwidth consumption. Each step includes sample commands and their output.

1. **Set up your load balancing configuration.** Configure load balancing virtual server mysitevip, and then configure all the services that you need. Bind the services to the virtual server. The following example creates ten services and binds the services to mysitevip.

```
> add lb vserver mysitevip HTTP 192.0.2.17 80
 Done
> add service service[1-10] 192.0.2.[240-249] HTTP 80
service "service1" added
service "service2" added
service "service3" added
.
.
.
```



```
service "service10" added
 Done
> bind lb vserver vserver1 service[1-10]
service "service1" bound
service "service2" bound
service "service3" bound
.
.
.
service "service10" bound
 Done
<!--NeedCopy-->
```

2. **Configure the stream selector.** Configure one of the following stream selectors:

   - To limit bandwidth consumption per client, configure a stream selector that identifies the client IP address.

     ```
     > add stream selector myselector CLIENT.IP.SRC
      Done
     <!--NeedCopy-->
     ```

   - To limit bandwidth consumption per user on the basis of the value of a request header that provides the user name, configure a stream selector that identifies the header. In the following example, the name of the header is UserHeader.

     ```
     > add stream selector myselector HTTP.REQ.HEADER( “UserHeader
        ” )
      Done
     <!--NeedCopy-->
     ```

3. **Configure a stream identifier.** Configure a stream identifier that uses the stream selector. Set the interval parameter to 60 minutes.

   ```
   > add stream identifier myidentifier myselector -interval 60 -
      sampleCount 1 -sort BANDWIDTH
    Done
   <!--NeedCopy-->
   ```

4. **Configure the responder action.** Import the HTML page that you want to send to users or clients that have exceeded the bandwidth consumption limit, and then use the page in responder action crossed_limits.

   ```
   > import responder htmlpage http://.1.1.1/stdpages/wait.html
      crossed-limits.html
   ```



```
This operation may take some time, Please wait...

 Done
> add responder action crossed_limits respondwithhtmlpage crossed-
    limits.html
 Done
<!--NeedCopy-->
```

5. **Configure the responder policies.** Configure responder policy myrespol1 with the rule ANALYTICS.STREAM(“myidentifier”).COLLECT_STATS and the action NOOP. Then, configure policy myrespol2 for determining whether a client or user has crossed the 100 MB limit. The policy myrespol2 is configured with the responder action crossed_limits.

```
> add responder policy myrespol1 'ANALYTICS.STREAM("myidentifier")
    .COLLECT_STATS' NOOP
 Done
> add responder policy myrespol2 'ANALYTICS.STREAM("myidentifier")
    .BANDWIDTH.GT(104857600)' crossed_limits
 Done
<!--NeedCopy-->
```

6. **Bind the responder policies to the load balancing virtual server.** The policy myrespol1, which only collects statistical data, must have the higher priority and a GOTO expression of NEXT.

```
> bind lb vserver mysitevip -policyName myrespol1 -priority 1 -
    gotoPriorityExpression NEXT
 Done
> bind lb vserver mysitevip -policyName myrespol2 -priority 2 -
    gotoPriorityExpression END
 Done
<!--NeedCopy-->
```

7. **Test the configuration.** Test the configuration by sending test HTTP requests, from multiple clients or users, to the load balancing virtual server and using the stat stream identifier command to view the statistics that are collected for the specified identifier. The following output displays statistics for clients.

```
> stat stream identifier myidentifier -sortBy BandW  - fullValues
Stream Session statistics
                            Req           BandW
192.0.2.30                  5000          3761
192.0.2.31                  29            2602
192.0.2.32                  25            51
```



```
7
8                                RspTime       Conn
9  192.0.2.30                    2             0
10 192.0.2.31                    0             0
11 192.0.2.32                    0             0
12  Done
13 >
14 <!--NeedCopy-->
```

AppExpert applications and templates

September 14, 2021

> **Warning**
>
> The application template functionality is deprecated from Citrix ADC 13.0 build 82.x onwards and as an alternative Citrix recommends you to use the Style Books. For more information, see Style Books topic.

An AppExpert application is a collection of configuration that you set up on the Citrix ADC appliance. Managing AppExpert applications is simplified by a GUI (GUI) that allows you to specify application traffic subsets and a distinct set of security and optimization policies for processing each traffic subset. Also, it consolidates deployment steps in one view, so you can quickly configure target IP addresses for clients and specify host servers.

To get started with an AppExpert application, you must first obtain the appropriate application template and import the template to the Citrix ADC appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If necessary, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the counters for various application components, statistics, and the Application Visualizer. You can also configure authentication, authorization, and auditing (authentication, authorization, and auditing) policies for the application.

AppExpert application terminology

Following are the terms used in the AppExpert applications feature and the descriptions of the entities for which the terms are used:

Public endpoint. The IP address and port combination at which the Citrix ADC appliance receives client requests for the associated web application. A public endpoint can be configured to receive either HTTP or secure HTTP (HTTPS) traffic. All client requests for the web application must be sent

to a public endpoint. An AppExpert application can be assigned multiple endpoints. You configure public endpoints after you import a template.

Application unit. An AppExpert application entity that processes a subset of web application traffic and load balances a set of services that host the associated content. The subset of traffic that an application unit must manage is defined by a rule. Each application unit also defines its own set of traffic optimization and security policies for the requests and responses that it manages. The Citrix ADC services associated with these policies are Compression, Caching, Rewrite, Responder, and application firewall.

By default, every AppExpert application with at least one application unit includes a default application unit, which cannot be deleted. The default application unit is not associated with a rule for identifying requests and is always placed last in the order of application units. It defines a set of policies for processing any request that does not match the rules that are configured for the other application units. Thereby ensuring all client requests are processed.

Application units and their associated rules, policies, and actions are included in AppExpert application templates.

Service. The combination of the IP address of the server that hosts the web application instance and the port to which the application is mapped on the server, in the format `\<IP address\>:\<Port\>`. A web application that serves many requests is hosted on multiple servers. Each server is said to host an instance of the web application, and each such instance of the web application is represented by a service on the Citrix ADC appliance. Services are deployment-specific and are therefore not included in templates. You must configure services after you import a template.

Application unit rule. Either a classic expression or a default syntax expression that defines the characteristics of a traffic subset for an application unit. The following example rule is a default syntax expression that identifies a traffic subset that consists of four image types:

```
HTTP.REQ.URL.SUFFIX.EQ("bmp")|| HTTP.REQ.URL.SUFFIX.EQ("gif")|| HTTP.REQ.
URL.SUFFIX.EQ("png")|| HTTP.REQ.URL.SUFFIX.EQ("jpg")
```

For more information about default syntax expressions and classic policy expressions, see Policies and Expressions.

Traffic Subset. A set of client requests that require a common set of traffic optimization and security policies. A traffic subset is managed by an application unit and is defined by a rule.

How appExpert application works

September 14, 2021

When the endpoint receives a client request, the Citrix ADC appliance evaluates the request against the rule that is configured for the topmost application unit. If the request satisfies this rule, the request is processed by the policies that are configured for the application unit, and then forwarded to a service. The choice of service depends on which services are configured for the application, and on settings such as the load balancing algorithm and persistence method configured for the application unit.

If the request does not satisfy the rule, the request is evaluated against the rule for the next topmost application unit. In this order, the request is evaluated against each application unit rule until the request satisfies a rule. If the request does not satisfy any of the configured rules, it is processed by the default application unit, which is always the last application unit.

You can configure multiple public endpoints for an AppExpert application. In such a configuration, by default, each application unit processes requests received by all the public endpoints and load balances all the services that are configured for the application. However, you can specify that an application unit processes traffic from only a subset of the public endpoints and load balances only a subset of the services that are configured for the AppExpert application.

The following flow diagram illustrates the AppExpert Application flow sequence for using a built-in application template.



If you prefer to create a customized application without using a template, do the following:

1. Create a custom application.
2. Configure application and deployment settings.
3. Export the configuration to new template files (optional).

4. Import the template files to other Citrix ADC appliances that require a similar AppExpert application configuration

Get started with appExpert

September 14, 2021

To get started with an AppExpert application, you must first obtain an application template and import the template to a Citrix ADC appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

The process of setting up an application can be done in two ways:

- Using a prebuilt application template
- Creating a custom application without using a template.

If you prefer to set up the application by using a prebuilt application template, do the following:

1. Download an application template.
2. Import template files to Citrix ADC appliance.
3. Verify application setup.
4. Configure application and deployment settings.
5. Export the configuration to new template files (optional).
6. Import the template files to other Citrix ADC appliances that require a similar AppExpert application configuration.

Citrix ADC's video tutorials enable you to understand Citrix ADC features in easy and simple way. Watch https://www.youtube.com/watch?v=aqayflvCR_0 video to learn how to set up an application using AppExpert Application template.

Downloading an application template

September 14, 2021

Note: Citrix no longer supports AppExpert application templates and does not allow you to download a copy. If you are using Citrix ADC version 13.0 or earlier, you can reach out to Citrix support to get a copy of an application template.

To set up AppExpert application, you must first download an application template from the Citrix community website at http://community.citrix.com to your local computer or Citrix ADC appliance. The application templates are imported and exported, so that you can easily share application-specific configurations within an organization or across organizations. An application template includes the following set of entities:

1. Application components (for example, webpages, files, archives, and web services)
2. Traffic management entities (for example, virtual server IP addresses and associated load-balancing algorithms, and SSL offload settings) for the application components.
3. Citrix ADC policies used for optimizing the application traffic.

Note: Application templates are available in different versions for configuring different types of Citrix ADC appliances.

Importing an application template

September 14, 2021

For Citrix ADC software version 9.3 or later, each AppExpert template has two XML files: a Template file and a Deployment file. You must import both files from your local computer to a Citrix ADC appliance. You can either import the template files from your computer to the AppExpert application templates directory in Citrix ADC appliance or upload files to a Citrix ADC appliance and then import them from the appliance.

Note: When you import a template from an appliance, you have to provide the variable value available in the template. By default, the pre-configured value is display `/en-us/citrix-adc/13/appexpert/appexpert-application-templates/creating-managing-templates/citrix-adc-application-template-deployment-files.html`

After you import the template files, the application-configuration and deployment information populates the target application automatically. The appliance imports all the configuration from the template files through the NITRO API. If you do not import the deployment file, the system generates an application populated with content switching virtual server configuration. For more information about the format of application templates and deployment files, see Understanding Citrix ADC Application Templates and Deployment Files.

When you import a template, if you do not include a deployment file, you have to configure the public endpoints in the application that the system automatically generates from the template. One endpoint for HTTP and another endpoint for HTTPS. When configuring a public endpoint of type HTTPS, make sure you enable the SSL feature, bind the server certificate, and include the server-certificate and certificate-key files.

For more information about configuring endpoints after you import a template, see Configuring Public Endpoints.

To import AppExpert application template files to a Citrix ADC appliance by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click **Import Template**.
3. On the **Import** page, set the following parameters:
   a) Application Name (mandatory)
   b) Template File (mandatory)
   c) Use deployment file
4. Click **Continue** to auto populate application-configuration and deployment information into an application.

Citrix ADC's video tutorials enable you to understand Citrix ADC features in an easy and simply way. Watch https://www.youtube.com/watch?v=AR9TwSD9uJM video to learn how to import an application template.

Verifying and testing application configuration

September 14, 2021

The GUI includes icons that indicate the states of the entities in the AppExpert application. These icons are displayed for applications and application units and are based on the health checks that the Citrix ADC appliance performs periodically on services and entities. The following table lists the icons and describes their meanings.

Icon	Entity	Indicates that
	Application	At least one public endpoint is up. The application will accept client requests from the public endpoints that are up.
	Application unit	The application unit is up. The application unit is up when at least one service or service group is up.

Icon	Entity	Indicates that
	Application	The public endpoint is out of service (disabled). This indicator is displayed when only one public endpoint is configured for the AppExpert application.
	Application	All the endpoints that are configured for the application are out of service. This indicator is displayed only when multiple endpoints are configured for the application.
	Application unit	All the services configured for the application unit are down.

You must ensure that the icons for each application and its application units are green at all times. If the icon that is displayed for an application is not green, verify that you have configured the public endpoints correctly. If the icon that is displayed for an application unit is not green, verify that the services are configured correctly. However, note that a green indicator does not mean that the state of all associated entities is UP. It only means that the application has sufficient resources (endpoints and services) to serve client requests. To verify that the state of all associated entities is UP, check the health of all the entities on the statistics page for the application.

Customizing the configuration

September 14, 2021

After you verify that the AppExpert application is working correctly, you can customize the configuration to suit your requirements.

After you verify that the AppExpert application configuration is working correctly, you can configure the application and the deployment settings to suit your requirements. When you import an application template and deployment file, the system automatically populates the target application with the available configuration settings (such as application units, application unit rules, policies, persistence settings, load balancing methods, profiles, and traffic settings). In this application, you can configure

deployment settings such as public endpoints, services, and service groups for each traffic subset. If you want the AppExpert application to manage a traffic subset that is not included in the template, you can either add an application unit for a traffic subset or modify the existing application unit. After you customize the configuration, you can also specify the order of evaluation for each traffic subset that the application manages.

Configuring an AppExpert application consists of the following steps:

1. Configuring Public Endpoints
2. Configuring Application Units
3. Specifying the Order of Evaluation
4. Viewing Application Configuration using Visualizer

Also, you can configure the policies that the template provided. If the AppExpert application template does not include policies for a particular Citrix ADC feature, such as Rewrite or application firewall, you can configure your own policies.

Configure public endpoints

September 14, 2021

If you did not specify a public endpoint when importing an AppExpert application, you can specify public endpoints after you create the application. You can configure one public endpoint of type HTTP and one public endpoint of type HTTPS for your AppExpert application.

If endpoints are already configured for the application, you can dissociate endpoints from the AppExpert application and delete any endpoints that you no longer need. Note that when you dissociate a public endpoint from the AppExpert application, the endpoint is automatically unbound from the associated application unit, but it is not deleted from the system.

To configure public endpoints for an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click Edit.
3. In the **Applications** page, go to **Public Endpoint** section and click the pencil icon.
4. In the **Public Endpoint** slider, set the following parameters.
   a) Public endpoint type. Select the radio button to define the endpoint type.
   b) Name. Name of the public endpoint.
   c) IP address. IP address of the public endpoint.
   d) Port. Port number of the public endpoint.
   e) Protocol. Select a protocol type as HTTP or HTTPS.
5. Click **Continue.**

6. In the **Application Units** section, select an application unit from the list.
7. Click **Continue** to set the policy and server details.
8. Click **OK** and then Done.
9. Click Close.

For more information about the parameters in the **Configure Public Endpoint** dialog box, see Content Switching.

Configure services and service groups for an application unit

September 14, 2021

When you configure a service or service group, you either modify an existing service or service group, or add new services to the AppExpert application. You add services or service groups if you did not specify them when you imported the application template. You also add services and service groups when you increase the number of servers that host instances of the application. You can configure a service and service group for an application unit only after you configure the service or service group for the AppExpert application.

To configure a service or service group for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application and then click **Edit.**
3. In the **Applications** page, select an application unit and then click **Continue**.
4. In the **Services and Service Groups** section, do the following:
    a) In the Service Binding slider, set the following parameters.
        i. Service. Select a load balancing service from the list or create a new service.
        ii. Weight. Provide a weight value for the service.
    b) Click **Bind** and then **Done.**
    c) In the ServiceGroup Binding slider, set the following parameters:
        i. Service Group Name. Select a load balancing service group or create a new service group.
        ii. Click **Bind** and then **Done**.
    d) Click **Done**.
5. Click **Continue** to set other configurations.

Create application units

September 14, 2021

You might need to add application units for traffic subsets that are either specific to your web application implementation or not defined in the template. When creating an application unit, you must configure a rule for the application unit.

To create an application unit for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications** page, go to **Application Units** section and click the **pencil** icon.

To configure policy expressions for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications** page, go to **Application Units** section and click the **+** icon. to create a unit and add policy expressions.
4. To specify the format of the new expression, do one of the following:
   a) To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
   b) To specify that you want to configure an advanced expression in the Rule box, click Default Syntax.
   c) In the Rule box, configure the expression.
5. Click **OK**.

Configuring application unit rules

September 14, 2021

You might want to configure an application unit rule to include or exclude certain types of traffic. When you configure the rule, you can also define the syntax of the expression.

To configure an application unit rule:

1. In the navigation pane of the GUI, expand AppExpert, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to modify the rule, and then click **Open**.
3. In the Configure Application Unit dialog box, do the following:
   a) To specify the format of the new expression, do one of the following:
      - To specify that you want to configure a classic expression in the Rule box, click **Classic Syntax**.

- To specify that you want to configure an advanced expression in the Rule box, click **Default Syntax**.

b) In the Rule box, configure the expression.

4. Click **OK**.

Configuring policies for application units

September 14, 2021

For an AppExpert application, you can configure policies for Compression, Caching, Rewrite, Responder, and Application Firewall. The templates that you download from the Citrix Community web site provide you with a set of policies that fulfill the most common application management requirements. You might want to fine-tune or customize these policies. If the set of policies provided for a given application unit does not include policies for a particular feature, you can create and bind your own policies for that feature.

If you create an AppExpert application without using a template, you must configure all the policies that the web application needs.

The GUI uses various icons to indicate whether or not policies are configured for a feature. For an application unit, if a policy is configured for a given feature, an icon that represents the feature is displayed. For example, if a compression policy is configured for an application unit, a compression icon is displayed in the Compression column for the application unit. For features for which no policy is configured, an icon depicting a plus sign (+) is displayed.

Note: When configuring policies for application units, you might need to configure policies and expressions that are either in the classic or default syntax. Additionally, when you configure default syntax policies, you might need to specify parameters such as Goto expressions and invoke policy banks.

For information about configuring policies and expressions in both formats, see Policies and Expressions.

Configuring compression policies

You can use either classic policies or advanced policies to configure compression, but you cannot bind compression policies of both types to the same application unit.

To configure a compression policy for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Compression column.

3. In the Configure Compression Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click Switch to Default Syntax if you want to configure a default syntax compression policy. If you want to bind or configure classic compression policies, and if you are in the default syntax view, you can click Switch to Classic Syntax to return to the classic policy view and begin modifying bound classic policies or create and bind new classic compression policies.
     Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you are in the default syntax view, when you click Insert Policy, the list that appears in the Policy Name column will include only default syntax policies. You cannot bind policies of both types to an application unit.
   - If you want to configure classic policies, click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
     You can configure both request-time and response-time classic compression policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
   - To modify a compression policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Compression Policy dialog box, modify the policy, and then click OK.
     For information about modifying a compression policy, see Compression.
   - To unbind a policy, click the name of the policy, and then click Unbind Policy.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click Regenerate Priorities.
   - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the Policy Name column, click **New Policy**. Then, in the Create Compression Policy dialog box, configure the policy, and then click **Create**.
     For information about modifying a compression policy, see Compression.
   - If you are configuring a default syntax expression, do the following:
     - In the Goto Expression column, select a Goto expression.
     - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Caching Policies

You can use only default syntax policies and expressions to configure Caching policies.

To configure Caching policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.

2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Caching column.
3. In the Configure Cache Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
     You can configure both request-time and response-time Caching policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
   - To modify a Caching policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the **Configure Cache Policy** dialog box, modify the policy, and then click **OK**.
     For information about modifying a Caching policy, see Integrated Caching.
   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.
   - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the Policy Name column, click **New Policy**. Then, in the **Create Cache Policy** dialog box, configure the policy, and then click **Create**.
     For information about modifying a Caching policy, see Integrated Caching.
   - In the Goto Expression column, select a Goto expression.
   - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring rewrite policies

You can use only default syntax policies and expressions to configure Rewrite policies.

To configure Rewrite policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Rewrite column.
3. In the **Configure Rewrite Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
     You can configure both request-time and response-time Rewrite policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance

evaluates response-time policies.

- To modify a Rewrite policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the Configure Rewrite Policy dialog box, modify the policy, and then click **OK**.
  For information about modifying a Rewrite policy, see Rewrite.
- To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
- To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
- To regenerate assigned priorities, click **Regenerate Priorities**.
- To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy.** Then, in the **Create Rewrite Policy** dialog box, configure the policy, and then click **Create**.
  For information about modifying a Rewrite policy, see Rewrite.
- In the Goto Expression column, select a Goto expression.
- In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click **Apply Changes**, and then click **Close**.

Configuring responder policies

You can use only default syntax policies and expressions to configure Responder policies.

To configure Responder policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Responder column.
3. In the **Configure Responder Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - To modify a Filter policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the Configure Responder Policy dialog box, modify the policy, and then click **OK**.
     For information about modifying a Responder policy, see Responder.
   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.
   - To insert a new policy, click Insert Policy and, in the list that is displayed in the Policy Name column, click New Policy. Then, in the Create Responder Policy dialog box, configure the policy, and then click Create.
     For information about modifying a Responder policy, see Responder.

- In the Goto Expression column, select a Goto expression.
- In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click **Apply Changes**, and then click **Close**.

Configuring Application Firewall Policies

You can configure both classic and default syntax policies and expressions for Application Firewall. However, if a policy of one type is already bound globally or to a virtual server that is configured on the appliance, you cannot bind a policy of the other type to an application unit. For example, if a default syntax policy is already bound either globally or to a virtual server, you cannot bind a classic policy to an application unit.

To configure Application Firewall policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the **Application Firewall** column.
3. In the **Configure Application Firewall Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Classic Expression or Advanced Expression depending on the type of expression you want to configure for the Application Firewall policy.
     Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you select Advanced Expression, when you click **Insert Policy**, the list that appears in the **Policy Name** column will include only default syntax policies. You cannot bind policies of both types to an application unit. This option is not available if a policy of either type is already bound either globally or to a virtual server.
   - To modify an application firewall policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Application Firewall Policy dialog box, modify the policy, and then click OK.
     For information about modifying a application firewall policy, see Policies.
   - To unbind a policy, click the name of the policy, and then click Unbind Policy.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click Regenerate Priorities.
   - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click New Policy. Then, in the **Create Application Firewall Policy** dialog box, configure the policy, and then click **Create**.
     For information about modifying a application firewall policy, see Policies.
4. Click **Apply Changes**, and then click **Close**.

Configuring Application Units

September 14, 2021

To configure an application unit by using the GUI:

1. Navigate to **AppExpert** > **Applications** > **Application Unit** section and then click the plus icon to add a new application unit for a traffic subset.
2. In the **Application Unit** slider, set the following parameters:
   - Name
   - Expression

   You can insert an expression either by adding the expression components manually or by using the Expression Editor link. To manually add an expression, enter a selector component and then type a period (.) to display a list from which you can select the next component. For example, type HTTP and then type a period. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. Select a component from the drop-down menu. The Expression* text box now displays the components that you have added to the expression (for example, HTTP.REQ). Continue adding components until the complete expression is formed.

   If you prefer assistance to form the expression, you can use the Expression Editor link. On the Expression Editor page, you can form an expression by selecting components from the drop-down boxes. Select the components and click Done to insert the expression on the Application Unit page.
3. Click **Continue** to bind services and service groups.
4. Click the **Service** section to select or add a virtual service and bind it to the application unit.
5. Click **Continue** and click the **Service Group** section to select or add a virtual service group and bind it to the application unit.
6. Click **Bind** and **Continue** to configure Advanced Settings (such as Policies, Method, Persistence, Protection, Profiles, Push, Authentication, and Traffic Settings) for the application unit.
7. Click the **plus** icon in each section to set the configuration parameters.
8. Click **OK** and then **Done**.

To edit an application unit for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, select an entity, click the edit icon and modify the application unit settings.

Note: You cannot modify the name and rule expression for an existing application unit.

Citrix ADC's video tutorials enable you to understand Citrix ADC features in an easy and simple way. Watch https://www.youtube.com/watch?v=bJ5_i8fV2hc video to learn how configure an application unit.

Configuring Public Endpoints for an application

September 14, 2021

To configure public endpoints for an application by using the GUI:

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Public Endpoint** section, click **+** to configure a new public endpoint.
3. In the **Public Endpoint** slider, do one of the following:
    a) Click **New** to create a new endpoint.
    b) Click **Existing Public Endpoint** to select an endpoint from the drop-down list.
4. Set the following endpoint parameters:
    a) Name
    b) IP address
    c) Protocol
    d) Port
5. Click **Continue** to configure additional settings such as application units, GSLB server bindings, policies, profiles, push, traffic settings, and authentication.
6. Click **OK** and then **Done**.
7. Click **Continue** and then **Done**.

To edit a public endpoint for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application, and click **Edit**. In the **Public Endpoint** section, select an endpoint, click the pen icon, and modify the endpoint settings.

To delete a public endpoint for an application by using the GUI:

Navigate to **AppExpert** > **Applications** > **Public Endpoint,** click the pen icon to view the delete icon next to the entity.

Citrix ADC's video tutorials enable you to understand Citrix ADC features in an easy and simply way. Watch https://www.youtube.com/watch?v=z4v-edQiVpw video to learn how to configure a public endpoint.

Specifying the order of evaluation of application units

September 14, 2021

Application unit rules are evaluated in the order in which they are placed in the GUI. The rule that is configured for the topmost application unit is always configured first, followed by the rule that is configured for the second topmost application unit, and so on. The default application unit is always evaluated last.

When a request matches the rule that is configured for an application unit, the request is processed by the application unit, and no further matching is performed. Therefore, the order of evaluation of application units becomes an important factor if the traffic subsets for two or more application units overlap. If the traffic subsets for two or more application units overlap, you must specify the order in which an incoming request is matched against the application unit rules.

To specify the order of evaluation of application units:

1. Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the **Pencil** icon and then hover the cursor over the check box to the left of the name of the application unit. Click the icon that appears next to the check box and hold down the mouse to drag the application up or down to a new location in the priority list.

Configuring persistency groups for application units

September 14, 2021

You can configure a persistency group for the application units in an AppExpert application. In the context of an AppExpert application, a persistency group is a group of application units that you can treat as a single entity for the purpose of applying common persistence settings. When the application is exported to an application template file, the persistency group settings are included, and they are automatically applied to the application units when you import the AppExpert application.

To configure a persistency group for an application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the **Applications View** dialog box, click the name of the application for whose application units you want to configure a persistency group, and then click Configure **Persistency Groups**.
3. In the **Configure Persistency Groups** dialog box, do one of the following:
   - To add a persistency group, click **Add**.
   - To modify a persistency group, click **Open**.
4. In the **Create Persistency Group** or **Configure Persistency Group** dialog box, set the following parameters:

- Group Name—Name of the persistency group. For the Citrix ADC appliance to recognize the persistency group as part of the application's configuration, the name of the AppExpert application must be included in the name of the persistency group, as a prefix. Therefore, by default, the appliance displays the prefix in the Group Name box, and you cannot remove that prefix. Enter a name of your choice after the prefix.
- Persistence—Type of persistence for the virtual server. If you select SOURCEIP, in the IPv4 Netmask box, enter a network mask that specifies the number of bits that the appliance must consider when creating persistence sessions. If you select COOKIEINSERT, in the Cookie Domain and Cookie Name boxes, specify a domain attribute to send in the Set-Cookie directive, and a name for the cookie, respectively.
- Timeout—Time period for which a persistence session is in effect.
- Backup Persistence—Type of backup persistence for the group.
- Backup Timeout—Time period, in minutes, for which backup persistence is in effect.
- Application Units—To add an application unit to the persistency group, in the Available Application Units box, click the application unit, and then click Add. To remove an application unit from the persistency group, in the Configured Application Units box, click the application unit, and then click **Remove**.

5. Click **OK**.

Viewing AppExpert applications and configuring entities by using the application visualizer

September 14, 2021

The Visualizer feature shows you a graphical representation of an application's configuration. It includes the name of the public endpoint, application units assigned to the public endpoint, and the number of policies and services bound to the application. You can use the Visualizer to obtain a visual overview of an AppExpert application's configuration and configure some of the displayed entities. By default, the Visualizer displays application units, services, and monitors for the selected application.

To view an AppExpert application by using the Application Visualizer:

1. Navigate to **AppExpert** > **Applications**, select an application entity, and click **Visualizer**.

Configuring user authentication, authorization, and auditing

September 14, 2021

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure AAA users and AAA user groups for an application by using the configuring utility

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Advanced Settings** section, click **Authorization**, and configure authorized users and user groups.
3. Click the **AAA** user section to bind authorized users to the application.
4. In the **AAA User** slider, set the parameters .
5. Click **Continue**, and then click **Authorization Policies** in the **Advanced Settings** section.
6. In the **Authorization Policy** slider, bind an authorization policy to the application.
7. Click **Continue**, and then click the **Authorization Group** section in the **Advanced Settings** section.
8. In the **AAA Group Binding** slider, bind an authorization user group to the application.
9. Click **Continue**, and then click **Policies** in the **Advanced Settings** section.
10. In the **Policies** slider, bind an **Audit Syslog** or **Audit NSlog** policy to the application.
11. Click **Continue** and then **Done**.

To edit AAA users and AAA user groups for an application by using the GUI:

Navigate to **AppExpert** > **Applications** > **Advanced Settings** and click **Authorization**. Then click the edit icon and specify values for user or user-group authorization settings.

To delete AAA users and AAA user groups by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Applications** page, click **Advanced Settings** and click **Authorization**. Click the delete icon next to the entity.

Monitoring a Citrix ADC application

September 14, 2021

After you customize the AppExpert application, you can view application statistics to make sure that the application and all its entities are working correctly. You can also use the Application Visualizer to monitor statistics associated with certain entities such as policies and virtual servers.

You can also view the hit counters for various entities at regular intervals to make sure that counters are being updated.

View application statistics

In the **Applications** node, you can select an application and view the Statistics page for the application. On the Statistics page, you can monitor the health and states of public endpoints and application units, and view the following statistical information:

- Requests and responses per second for each of the public endpoints and application units.
- Bytes per second, at each endpoint, for incoming and outgoing traffic.
- Application unit hit counters and the number of client and server connections for each application unit.
- Statistics for the services that are bound to the application units.

On the Statistics page, you can also view CPU usage, memory usage, and system logs.

To view statistics for an application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the application for which you want to view statistics, and then click **Statistics**.

Monitoring an Application by Using the Application Visualizer

You can use the Application Visualizer to monitor the number of requests received per second at a given point in time by the vservers and the number of hits per second at a given point in time for Rewrite, Responder, and Cache policies.

To view statistical information for vservers, Rewrite policies, Responder policies, and Cache policies in the Visualizer:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, select the application for which you want to view statistical information, and then click **Visualizer**.
3. In the **Application Visualizer** window, do the following:
   - To view the statistics, click **Show Stats**.
     The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time and has to be refreshed manually.
   - To refresh the statistical information, click **Refresh Stats**.

Viewing Hits

The hit counters that are provided for various AppExpert application entities enable you to monitor the functioning of public endpoints and application units. For an application, the Hits dialog box displays the total number of requests received by each configured public endpoint. For an application unit, the Hits dialog box displays the number of requests that the application unit processed from each

of the public endpoints and the total hit count. For instructions on viewing hit counters, see Verifying and Testing the Configuration.

Deleting an application

September 14, 2021

If you no longer need an application and its application units, you can delete it. When you delete an AppExpert application, backend services are not deleted, and any public endpoints that the application used become available for use by other applications.

When deleting an application, you are also prompted to specify whether you want to delete any bound policies and actions that are not used elsewhere.

To delete an application unit for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the delete icon next to the entity

Configure application authentication, authorization, and auditing

September 14, 2021

You can configure Authentication, Authorization, and Auditing (AAA) for the applications that you configure on the appliance. An authentication policy that is configured for an application defines the type of authentication to apply when a user or group attempts to access the application. If external authentication is used, the policy also specifies the external authentication server. Authorization policies configured for an application specify whether a particular user or group can access the application. Auditing policies define the audit log type, the level at which logging is performed, and other audit server settings. Authentication and auditing policies use the classic policy format.

Authentication policies, authorization policies, and auditing policies can be configured in any order. However, before you configure AAA for an application, you must configure a public endpoint for the application.

Configuring authentication for an application involves specifying an authentication FQDN, an authentication virtual server, a server certificate, and authentication and session policies. Authentication policies are automatically bound to the authentication virtual server specified for the application.

To configure authentication for an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.

2. In the details pane, do one of the following:
   a) Click Add to add an authentication for a new application.
   b) Click Edit to modify an existing application.
3. In the **Applications** page, select an Application Unit.
4. In the **Application Unit** slider page, click Authentication from the **Advanced Settings** section.
5. In the **Authentication** section, select the authentication type as follows:
   a) Form based authentication
   b) 401 based authentication
   c) None
6. Click **OK** and then click **Done**.

Configure application authorization

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure permissions for a AAA user or group to access an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the AppExpert application for which you want to configure a user or group access.
3. In the **Applications** page, and then click Authorization. from the **Advanced Settings** section.
4. Do one of the following:
   - If the AAA user or group for which you want to configure permissions are already in the Groups/Users tree, drag the user or group from the Groups/Users tree to the Users or Groups node in the application tree. Then, right-click the user or group and click Allow.
   - If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the application tree, right-click Users or Groups, and then click Add. In the Create AAA Group or Create AAA User dialog box, fill in the values, click Create, and then click Close.
     The user or group is created with the permission set to Allow. To change the permission setting, right-click the group or user, and then click the permission setting.
5. Click **Done** and then click **Close**.

Configure application auditing

When you configure auditing policies for an application, you must specify the server to which the log messages must be directed, the format of the messages logged, and the log level. Optionally, you can configure other settings, such as the log facility and date format. Auditing policies are automatically bound to all the AppExpert application's public endpoints.

To configure auditing policies for an application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the application for which you want to configure auditing policies.
3. In the Application Unit slider page, click + icon in the **Policies** section to configure the auditing policies.
4. In the **Policies** slider page, select policy type as Syslog auditing or Nslog auditing and click **Continue**.
5. In the Policy binding section, set the following parameters.
    a) Select a policy for binding. If you do not have a policy for binding. click + to create a new policy.
    b) To create a new auditing policy, under Policy Name, click **New Policy,** and then, in the **Policy** page do the following:
        i. In the Name box, type a name for the policy.
        ii. The Name box already contains the string that is required at the beginning of the server name. You cannot modify the string.
        iii. From the Auditing Type list, select the auditing type (either SYSLOG or NSLOG).
        iv. If the audit server you want to specify is already listed in the Server list, select the server from the list, and then, if you want to modify the server settings, click Modify. In the Configure Auditing Server dialog box, modify the settings as appropriate, and then click OK. For more information about the settings in the Configure Auditing Server dialog box, see Auditing Authenticated Sessions.
        v. If you want to configure a new audit server, click New, and then, in the Create Auditing Server dialog box, type a name for the server, specify the server IP address, port number, and other settings as appropriate. When finished, click **OK**.
        vi. Click **Create**.
    c) To change the priorities for the new auditing policies you created, under Priority, for each policy for which you want to change the priority, double-click the priority value and type new priority value.
    d) To regenerate priorities, click **Regenerate Priorities**.
    e) To unbind a policy, click the policy, and then click **Unbind Policy**.
    f) To modify a policy, click the policy, and then click **Modify Policy.**
6. Click **Apply Changes,** and then click **Close**.

Disabling AAA for an Application

After you configure AAA for an application, you can disable the AAA configuration for that application. When you disable AAA for an application, the configuration is not lost. You can enable AAA for the application when you want to reapply the configuration.

To enable or disable AAA for an application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the application for which you want to enable or disable AAA, and then do one of the following:
3. To disable AAA for the application, click **Turn Off AAA.**
4. To enable AAA for the application, click **Turn On AAA**.

Setting up a custom Citrix ADC application

September 14, 2021

If an AppExpert application template is not available for the Web application that you want to manage through the Citrix ADC appliance, or if available AppExpert application templates do not suit your requirements, you can create an AppExpert application without a template.

To create an AppExpert application without a template, you must first create an application and application units. Then, you configure public endpoints, services, and service groups. Finally, you configure the policies that determine how application traffic is evaluated and processed.

After you create the application and application units and configure policies, you must verify the configuration and test it to make sure that it is working correctly, just as you would when you configure an application by using a prebuilt AppExpert application template. Then, you must monitor the application to make sure that the application and its entities are working correctly.

Creating an application

When you create an AppExpert application, the appliance creates a container to which you can add application units. The default application unit is not created until you create the first application unit.

To create an AppExpert application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click **Applications**, and then click **Add**.
3. In the **Create Application** dialog box, in Name, enter a name for the application, and then click **OK**.

Creating application units

For each subset of traffic associated with your web application, you must create an application unit.

To create an application unit for the AppExpert application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.

2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. Click **Create**.

Configuring public endpoints for an AppExpert application

After you have created all the application units that you require, you must configure one or more public endpoints to enable clients to access the web application through the Citrix ADC appliance.

To configure public endpoints for an AppExpert application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click **Configure Public Endpoints**.
3. In the Choose Public Endpoints dialog box for the application, do one of the following:
   - If the endpoints you want are listed in the dialog box, click the corresponding check boxes.
   - If you want to specify all the public endpoints, click **Activate All**.
   - If you want to dissociate endpoints from the AppExpert application, clear the corresponding check boxes.
   - If you want to create a new public endpoint, click **Add**. Then, in the Create public endpoint dialog box, configure endpoint settings, and then click **OK**.
     In the **Create public endpoint** dialog box, you can specify only the name, IP address, port, and protocol for the endpoint. You can specify additional endpoint settings after you create the public endpoint. To specify additional endpoint settings, after you create the endpoint, in the Choose Public Endpoints dialog box, click the endpoint, and then click **Open**. Then, in the **Configure Public Endpoint** dialog box, provide additional settings, and then click **OK**.
     For more information about the parameters in the **Create public endpoint** and **Configure Public Endpoint** dialog boxes, see Content Switching.
   - If you want to modify a public endpoint, click the endpoint, and then click **Open**. Then, in the **Configure Public Endpoint** dialog box, modify settings for the endpoint, and then click **OK**.
     For more information about the parameters in the Configure Public Endpoint dialog box, see Content Switching.
4. Click **Close**.

Configuring public endpoints for an application unit

For an application unit, you specify public endpoints in the same way as you would specify public endpoints for an application that is created from an AppExpert application template. For more information about specifying a subset of the endpoints for an application unit, see Configuring Endpoints

for an Application Unit.

To configure endpoints for an application unit by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click **Configure Public Endpoints**.
3. In the **Choose Public Endpoints** dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click the corresponding check boxes.
4. Click **OK**.

Configuring Services and Service Groups for an AppExpert Application

Services and service groups are available for application units only after you configure the services and service groups for the AppExpert application. Therefore, you must configure services and service groups for the AppExpert application before you configure the services for the application units. All the services and service groups that you configure for an AppExpert application must use the same protocol (either HTTP or HTTPS). The procedure for configuring services and service groups for an AppExpert application that is not created from a template is the same as that for an application created from a template.

To configure a service or service group for the AppExpert application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to configure services or service groups, and then click **Configure Backend Services**.
3. In the Configure Backend Services dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. On the **Service** or **Service Groups** tab, do one of the following:
   - If the services or service groups that you want are listed on the tab, click the corresponding check boxes.
   - If you want to specify all the services or service groups, click **Activate All**.
   - If you want to create a new service or service group, click **Add**. Then, in the **Create Service** dialog box or **Create Service Group** dialog box, configure settings for the service or service group, respectively, and then click **Create**.
   - If you want to modify a service, click the service, and then click Open. Then, in the **Configure Service** dialog box or **Create Service Group** dialog box, configure settings for the

service or service group, respectively, and then click **OK**.

For information about the settings in the Create Service, Configure Service, and **Create Service Group** dialog boxes, see Load Balancing.

Configuring services and service groups for an application unit

After you configure services and service groups, you must configure services and service groups for each application unit. However, this step is not necessary if each backend service hosts all the content associated with the web application. You configure services and service groups for an application unit if the content associated with the application unit is hosted on only a subset of the backend servers.

To configure services or service groups for an application unit by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. In the **Services** or **Service Groups** tab, do one of the following:
   - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the Weight column, specify the weight that you want to assign to each configured service.
   - To specify all services or service groups, click **Activate All**.
5. On the **Method** and **Persistence** and **Advanced** tabs, specify the desired parameters.
6. Click **OK**.

Configuring policies

The procedures for configuring policies for an AppExpert application that is created without using a template are the same as those for an AppExpert application that was created from a template. For more information, see Configuring Policies for Application Units.

Creating and managing template files

September 14, 2021

After you set up an AppExpert application and customize it to suit your requirements, you can create a template from the configuration and then share the template with other administrators. Or, you can create a template and then import the template to other Citrix ADC appliances that require a similar AppExpert application configuration. This simplifies and expedites the process of setting up similar applications on other appliances.

AppExpert application template files can be exported either to the template directory on the Citrix ADC appliance or to a folder on your local computer. You can then upload and download the templates to and from the Citrix ADC appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

AppExpert application template files can be exported either to the template directory on the Citrix ADC appliance or to a folder on your local computer. You can then upload and download the templates to and from the Citrix ADC appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

Exporting an AppExpert Application to a Template File

September 14, 2021

When you export an AppExpert application, all application-configuration information is exported to a template file, and all deployment-specific information is exported to a deployment file. The string _deployment is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the Citrix ADC appliance, the template file is stored in the /nsconfig/nstemplates/applications directory and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory. If you have configured a Citrix Gateway application, you can choose to include the Citrix Gateway policies in the template.

To export an AppExpert application to a template file by using the GUI:

1. Navigate to **AppExpert** > **Application**, select an application entity, and then click **Edit**.
2. On the **Applications** page, click the **Export** as a Template link to export the application configuration and deployment settings as a template.
3. In the **Export Application** slider, set the following parameters:
    a) Template Filename
    b) Deployment Filename
4. Click **Continue** and **Done**.
5. Navigate to **AppExpert** > **Application** and click **Manage Templates** to show the exported configuration as files on the **Template File** and **Deployment File** tabs.

Exporting a Content Switching Virtual Server Configuration to a Template File

September 14, 2021

You can also export a content switching configuration as an application template. You can export a content switching virtual server configuration to an application template either from the Content Switching Virtual Servers pane or from the Content Switching Visualizer. Configuration information, which includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies, is exported to a template file and all deployment-specific information is exported to a deployment file. The string "_deployment" is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the Citrix ADC appliance, the template file is stored in the /nsconfig/nstemplates/applications directory on the Citrix ADC appliance and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory.

For more information about the format of application templates and deployment files, see Understanding Citrix ADC Application Templates and Deployment Files. The configuration information that is exported includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies.

However, if the content switching virtual server is already configured as the public endpoint for an AppExpert application, you cannot export the configuration to a template file . In this scenario, you must export the associated AppExpert application to a template.

For more information about exporting an AppExpert application to a template file, see Exporting an AppExpert Application to a Template File.

To export a content switching configuration to an application template file from the Content Switching Visualizer by using the GUI:

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click Visualizer.
3. In the Content Switching Visualizer, click the icon for the content switching vserver, click Related Tasks, and then click Create Template.
4. In the Export…as Template dialog box, enter a name for the template file, and then do one of the following:
    - To export the template file to the appliance, make sure that Browse (Appliance) is displayed.
    - To export the template file to your computer, click the Browse (Appliance) drop-down menu, click Local, browse to the location to which you want to save the file, and then click Save.

5. Provide the following information:
   - **Introduction Description**—Any text that introduces the AppExpert application template during import. This text is displayed on the Specify Application Name page of the AppExpert Template Wizard when the template is imported.
   - **Summary Description**—Any summary that you might want to display on the Summary page of the AppExpert Template Wizard when the template is imported.
   - **Author**—The name of the author of the template.
   - **Major**—The major version number of the template.
   - **Minor**—The minor version number of the template. This number is appended to the major version number and displayed on the Summary page of the AppExpert Template Wizard, during import, in the format Major.Minor.
6. Click OK.

To export a content switching configuration to an application template file from the Content Switching Virtual Servers pane by using the GUI:

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click Create AppExpert Template.
3. Perform steps 4 through 6 described in **To export a content switching configuration to an application template file from the Content Switching Visualizer** procedure.

Creating Variables in Application Templates

September 14, 2021

Application templates support the declaration of variables in the policy expressions and actions that are configured for an application. The ability to declare variables in policy expressions and actions enables you to replace preconfigured values in expressions (for example, configurable parameters such as the host name of a server or the target for a Rewrite action) with values that suit the environment into which you are importing the template. If variables have been configured for an AppExpert application template, the AppExpert Template Wizard, which appears when you import an AppExpert application template, includes a Specify Variable Values page on which you can specify appropriate values for the variables that are configured for the template.

As an example, consider the following policy expression that is configured to evaluate the value of the Host header in an HTTP request:

```
1 HTTP.REQ.HEADER("Host").CONTAINS("server1")
2 <!--NeedCopy-->
```


If you want the server name to be configurable at import time, you can specify the string “server1” as a variable. When importing the template, you can specify a new value for the variable on the Variables tab.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

In the export application template wizard, you can define variables in certain fields for the following entities:

- Cache policies
- Rewrite policies
- Rewrite actions
- Responder policies
- Responder actions

To configure a variable in a policy expression or action by using the GUI:

1. Navigate to **AppExpert > Variables**.
2. In the **Variables** page, click **Add**.
3. In the **Create Variables** page, set the following parameters.
   Name. Name of the variable.

   Scope. Select the scope as Global or transaction.

   Type. Select the variable type as text, ulong, map.

   Expires in. Enter the expiry date.

   If Full*. Action to perform if an assignment to a map exceeds its configured max-entries:
   lru - (default) reuse the least recently used entry in the map.
   undef - force the assignment to return an undefined (Undef) result to the policy executing the assignment.
   Possible values: undef, lru
   Default value: lru.

   if no value. Value expiration in seconds. If the value is not referenced within the expiration period it will be deleted. 0 (the default) means no expiration. Minimum value: 0, Maximum value: 31622400

Init Value. Initialization value for this variable, to which a singleton variable or map entry will be set if it is referenced before an assignment action has assigned it a value. If the singleton variable or map entry already has been assigned a value, setting this parameter will have no effect on that variable value. Default: 0 for ulong, NULL for text Maximum Length: 127

Comments. A brief description about the variable.

4. Click **Close**.

Uploading and Downloading Template Files

September 14, 2021

Template files can be uploaded from your local computer to the Citrix ADC appliance or downloaded from the appliance to your local computer. On the appliance, AppExpert application templates are always stored in the AppExpert application templates directory, which is `/nsconfig/nstemplates/applications/`.

To upload an AppExpert application template from your local computer to the Citrix ADC appliance:

1. Navigate to **AppExpert > Applications**.
2. In the details pane, click **Manage Templates**.
3. In the Application Templates dialog box, click **Upload**.
4. Browse to the directory in which the template file is stored, click the template file, and then click **Select**.

   The template file is uploaded to the AppExpert application template directory on the appliance.

To download an AppExpert application template from the Citrix ADC appliance to your local computer:

1. Navigate to **AppExpert > Applications**.
2. In the details pane, click **Manage Templates**.
3. In the Application Templates dialog box, click the AppExpert application template that you want to download, and click **Download**.
4. Browse to the directory to which you want to save the file, and then click **Save**.

Understanding Citrix ADC Application Templates and Deployment Files

September 14, 2021

When you export a Citrix ADC application, the following two files are automatically created:

- **Citrix ADC application template file**. Contains application-configuration information such as application units, rules, and configured policies.
- **Deployment file**. Contains deployment-specific information such as public endpoints, services, associated IP addresses, and configured variables.

In a template file or deployment file, each unit of application-configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, each public endpoint and associated endpoint details are encapsulated within the <appendpoint> and </appendpoint> tags, and all the endpoint elements are encapsulated within the <appendpoint_list> and </appendpoint_list> tags.

Note: After you export a Citrix ADC application, you can add elements, remove elements, and modify existing elements before importing the application to a Citrix ADC appliance.

Example of a Citrix ADC Application Template

Following is an example of a template file that was created from a Citrix ADC application called **SharePoint_Team_Site**:

```
<?xml version="1.0" encoding="UTF-8" ?>
<template>
<template_info>
   <application_name>SharePoint_Team_Site</application_name>
   <templateversion_major>1</templateversion_major>
   <templateversion_minor>1</templateversion_minor>
   <author>Ed</author>
   <introduction>An application for managing a SharePoint team site
      with images, reports, and, XML content.</introduction>
   <summary>This template includes variables</summary>
   <version_major>9</version_major>
   <version_minor>3</version_minor>
   <build_number>38</build_number>
</template_info>
<apptemplate>
   <rewrite>
      <rewriteaction_list>
         <rewriteaction>
            <name>Rw_name</name>
            <type>replace</type>
            <target>HTTP.REQ.BODY(10000).AFTER_REGEX(re/number/).
               BEFORE_REGEX(re/address/)</target>
            <stringbuilderexpr>"NA"</stringbuilderexpr>
            <allow_unsafe_pi1>NO</allow_unsafe_pi1>
         </rewriteaction>
```



```
            <rewriteaction>
               .
               .
               .
            </rewriteaction>
            .
            .
            .
         </rewriteaction_list>
         <rewritepolicy_list>
            <rewritepolicy>
               <name>Rw_number_NA</name>
               <rule>HTTP.REQ.BODY(100000).CONTAINS("admin")</rule>
               <action>Rw_name</action>
            </rewritepolicy>
            <rewritepolicy>
               .
               .
               .
            </rewritepolicy>
            .
            .
            .
         </rewritepolicy_list>
      </rewrite>
      <appunit_list>
         <appunit>
            <name>SharePoint_Team_Sitedefault</name>
            <rule />
            <expressiontype>PE</expressiontype>
            <servicetype>HTTP</servicetype>
            <ipv46>0.0.0.0</ipv46>
            <ipmask>*</ipmask>
            <port>0</port>
            <range>1</range>
            <persistencetype>NONE</persistencetype>
            <timeout>2</timeout>
            <persistencebackup>NONE</persistencebackup>
            <backuppersistencetimeout>2</backuppersistencetimeout>
            <lbmethod>LEASTCONNECTION</lbmethod>
            <persistmask>255.255.255.255</persistmask>
            <v6persistmasklen>128</v6persistmasklen>
            <pq>OFF</pq>
            <sc>OFF</sc>
            <m>IP</m>
```



```
            <datalength>0</datalength>
            <dataoffset>0</dataoffset>
            <sessionless>DISABLED</sessionless>
            <state>ENABLED</state>
            <connfailover>DISABLED</connfailover>
            <clttimeout>180</clttimeout>
            <somethod>NONE</somethod>
            <sopersistence>DISABLED</sopersistence>
            <redirectportrewrite>DISABLED</redirectportrewrite>
            <downstateflush>DISABLED</downstateflush>
            <gt2gb>DISABLED</gt2gb>
            <ipmapping>0.0.0.0</ipmapping>
            <disableprimaryondown>DISABLED</disableprimaryondown>
            <insertvserveripport>OFF</insertvserveripport>
            <authentication>OFF</authentication>
            <authn401>OFF</authn401>
            <push>DISABLED</push>
            <pushlabel>none</pushlabel>
            <l2conn>OFF</l2conn>
         </appunit>
         <appunit>
            .
            .
            .
         </appunit>
         .
         .
         .
      </appunit_list>
      </apptemplate>
      <parameters>
         <property_list>
            <property>
               <variable_definition_list>
                  <variable_definition>
                     <name>body_size</name>
                     <defaultvalue>10000</defaultvalue>
                     <description>Evaluation Scope</description>
                     <startindex>14</startindex>
                     <length>5</length>
                  </variable_definition>
                  .
                  .
                  .
               </variable_definition_list>
```



```
            <object_type>rewriteaction</object_type>
            <object_name>Rw_name</object_name>
            <name>target</name>
         </property>
         .
         .
         .
      </property_list>
   </parameters>
</template>
<!--NeedCopy-->
```

Example of a Deployment File

Following is the deployment file associated with the **SharePoint_Team_Site** application in the preceding example:

```
<?xml version="1.0" encoding="UTF8" ?>
<template_deployment>
   <template_info>
      <application_name>SharePoint_Team_Site</application_name>
      <templateversion_major>1</templateversion_major>
      <templateversion_minor>1</templateversion_minor>
      <author>Ed</author>
      <introduction>An application for managing a SharePoint team site
         with images, reports, and, XML content.</introduction>
      <summary>This template includes variables</summary>
      <version_major>9</version_major>
      <version_minor>3</version_minor>
      <build_number>38</build_number>
   </template_info>
   <appendpoint_list>
      <appendpoint>
         <ipv46>10.111.111.1</ipv46>
         <port>80</port>
         <servicetype>HTTP</servicetype>
      </appendpoint>
   </appendpoint_list>
   <service_list>
      <service>
         <ip>10.102.29.5</ip>
         <port>80</port>
         <servicetype>HTTP</servicetype>
      </service>
```



```
27          <service>
28             .
29             .
30             .
31          </service>
32          .
33          .
34          .
35       </service_list>
36       <variable_list>
37          <variable>
38             <name>body_size</name>
39             <description>Evaluation Scope</description>
40             <value>10000</value>
41          </variable>
42          <variable>
43             .
44             .
45             .
46          </variable>
47          .
48          .
49          .
50       </variable_list>
51    </template_deployment>
52    <!--NeedCopy-->
```

Deleting a Template File

September 14, 2021

If you no longer need an application template and its configuration, you can delete it. When you delete a template, the template XML file that is stored in the application template directory gets deleted. When you delete a template file, you are prompted to confirm the deletion. Click **Yes** to confirm and delete the selected file from the directory.

To delete a template file from the application template directory by using the GUI:

1. Navigate to **AppExpert > Applications** and then click **Manage Template**. Select a file from **Template Files** tab page or **Deployment Files** tab page and click **Delete**.

Citrix gateway applications

September 14, 2021

When you configure an AppExpert application to manage a web application through the Citrix® Citrix ADC® appliance, you also create a set of application units and configure a set of traffic optimization and security policies for each unit. The policies that you configure for each application unit (policies for features such as Compression, Caching, and Rewrite) evaluate traffic that is meant only for that unit. In addition to these policies, you might want to configure Access Gateway policies for the application as a whole to optimize the application traffic when accessed through the Access Gateway. The Access Gateway Applications feature enables you to configure Access Gateway policies (Authorization, Traffic, Clientless Access, and TCP Compression) for an AppExpert application. After you configure Citrix Gateway policies for AppExpert applications, you can include the policy configuration in the AppExpert application templates that you create.

You can also configure Citrix Gateway policies for intranet subnets, file shares, and other network resources. Finally, you can create bookmarks for AppExpert applications and certain resources if you want users to be able to access them from the Citrix Gateway home page.

You can configure the entities in the Citrix Gateway Applications feature only by using the GUI.

How an Citrix Gateway application works

When you create an AppExpert application in the Applications node in the GUI, a corresponding Access Gateway application is automatically created in the Access Gateway Applications node. Additionally, a rule that uses the AppExpert application's configured public endpoint is automatically created for the Access Gateway application entry. If multiple endpoints are configured for the AppExpert application, the rule includes all the configured public endpoints. The Citrix ADC appliance uses this rule to apply any configured Access Gateway policies to the traffic received at the AppExpert application's public endpoint. Traffic received at the AppExpert application's public endpoint is first evaluated against the Citrix Gateway policies and then evaluated against the policies configured for AppExpert application's application units.

The rule that is created for the Clientless Access policies for an Access Gateway application is an advanced expression that also uses the public endpoint that is configured for the AppExpert application. Therefore, before you configure Citrix Gateway policies for an AppExpert application, you must configure public endpoints for the AppExpert application.

When you include the Citrix Gateway configuration in an application template, deployment-specific information, such as IP address and port information, and the rule that is created from this information are not included in the template.

How a Citrix ADC configuration for a file share works

On the Citrix ADC appliance, you can configure Authorization policies for a file share that is hosted on your organization's network.

When you create a file share, you specify a name for the file share and the network path to the file share. In the network path, you can specify either the name of the server or the server IP address. A rule that uses the components of the file share path is automatically created for the file share. This rule enables the appliance to identify requests for files hosted on the file share server. Any Authorization policies that are configured for the file share are applied to incoming requests.

The Citrix ADC configuration for a file share cannot be saved in AppExpert application templates.

How a Citrix ADC configuration for an intranet subnet works

For the intranet subnets that form a part of your network, you can configure policies for Authorization, Traffic, and TCP Compression on the Citrix ADC appliance. When adding an intranet subnet, you specify the IP address and the netmask of the intranet subnet. A rule that uses these two parameters is automatically created for the intranet subnet. The appliance applies the configured policies to any request that has a destination IP address and netmask set to the subnet's IP address and netmask, respectively.

The Citrix ADC configuration for an intranet subnet cannot be saved in AppExpert application templates.

How other resources category works

The Other Resources category enables you to configure Access Gateway policies for any network resource by using a rule of your choice. When you configure the Citrix ADC appliance to process requests for the network resource, you configure a classic expression to identify the requests that are associated with the network resource. You can configure Authorization, Traffic, Clientless Access, and TCP Compression policies for a network resource in Other Resources. The Citrix ADC appliance applies the configured Citrix Gateway policies to any requests that match the configured rule.

The Citrix ADC configuration for a network resource in Other Resources cannot be saved in AppExpert application templates.

Entity naming conventions

The Citrix Gateway Applications feature enforces a naming convention for some of the entities that you create in this feature. For example, the names of the profiles that you create for Traffic policies for an intranet subnet always begin with a string that consists of the name of the intranet subnet followed by an underscore (_). The name that you provide for the entity is appended to this string. If the name of a

subnet is “subnet1,” the name of the profile begins with “subnet1_.” When such a naming convention is required (in the text box in which you type the name of an entity, for example), the user interface automatically inserts the string with which the name of the entity must begin and does not allow you to modify it.

Adding intranet subnets

September 14, 2021

You can specify authorization and Traffic policies for traffic that is bound for the intranet subnets that are configured in your network. The rules for these policies are automatically created by using the parameters you specify for the subnet.

To configure an intranet subnet by using the GUI:

1. In the navigation pane of the GUI, expand **AppExpert**, and then click Access Gateway Applications.
2. In the details pane, do one of the following:
   - To add an intranet subnet, click **Intranet Subnets**, and then click **Add**.
   - To modify an intranet subnet, click an intranet subnet, and then click **Open**.
3. In the **Create Intranet Subnet** or **Configure Intranet Subnet** dialog box, do the following:
   a) In the Name box, type a name for the intranet subnet you are adding. This parameter cannot be changed for an existing intranet subnet.
   b) In the IP Address box, type the IP address of the intranet subnet.
   c) In the Netmask box, type the netmask that will be used for the intranet subnet.
   d) Click **Create** or **OK**, and then click **Close**.

Adding other resources

September 14, 2021

For a network resource that you add to Other Resources, you must configure a classic expression that identifies the subset of traffic associated with the resource. For more information about configuring a classic expression, see the .

To configure a resource in other Resources by using the GUI:

1. In the navigation pane of the GUI, expand AppExpert, and then click **Access Gateway Applications**.
2. In the details pane, do one of the following:

- To add a resource, click **Other Resources**, and then click **Add**.
- To modify a resource, click a resource, and then click **Open**.

3. In the **Create Resource** or **Configure Resource** dialog box, do the following:
   a) In the Name box, type a name for the resource you are adding. This parameter cannot be changed for an existing resource.
   b) In the Rule box, type the rule that will identify the subset of traffic that is associated with the resource you are adding.
      Alternatively, click **Configure**, and then create the rule in the **Create Expression** dialog box.
   c) Click **Create** or **OK**, and then click **Close**.

Configuring authorization policies

September 14, 2021

You can configure Citrix Gateway authorization policies for AAA users and groups to access a resource.

To configure permissions for a AAA user or group to access a resource by using the GUI:

1. In the navigation pane of the GUI, expand AppExpert, and then click **Access Gateway Applications**.
2. In the details pane, in the Authorization column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure authorization policies for AAA users and groups.
3. Do one of the following:
   - If the AAA user or group for which you want to configure permissions is already in the Groups/Users tree, drag the user or group from the Groups/Users tree to the Users or Groups node in the <application name> tree. Then, right-click the user or group and click **Allow**.
   - If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the <application name> tree, right-click Users or Groups, and then click **Add**. In the Create **AAA Group** or **Create AAA User** dialog box, fill in the values, click **Create**, and then click **Close**.
     The user or group is created with the permission set to Allow. To change the permission setting, right-click the group or user, and then click the permission setting.
4. Click **Close**.

Configuring traffic policies

September 14, 2021

The traffic policies that you configure for the resources in the Citrix Gateway Applications node control client connections to the application. You do not have to configure a rule for the resource. The rule created automatically when you create the resource. You only need to associate a request profile with the traffic policy. In the traffic profile, you specify parameters such as the protocol, application timeout, and file type association.

To configure traffic policies for a resource

1. In the navigation pane of the GUI, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the Traffic column, click the icon provided for the application, file share, intranet subnet, or resource for which you want to configure traffic policies.
3. In the **Configure Traffic Policies** dialog box, do the following:
   - To specify an existing traffic policy, click **Insert Policy**, and then, in the Policy Name column, click the name of the policy.
   - To configure a new policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create Traffic Policy dialog box, in the Name box, after the underscore (_), type a name for the policy. Then, in Request Profile, either select an existing request profile or click New to configure a new request profile. You can also select an existing profile and then click Modify to modify the profile.
     For more information about configuring a traffic policy or profile, see Citrix Gateway.
   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click Modify Policy. To modify only the associated profile, in the Profile column, click the name of the profile, and then click **Modify Profile**.
   - To regenerate the priorities assigned to the policies, click **Regenerate Priorities**.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes**, and then click **Close**.

Configuring clientless access policies

September 14, 2021

Clientless access, when configured for a resource on the Citrix ADC appliance, allows end-users to access the resource without using the Citrix Gateway client software. Users can use web browsers

to access resources such as Outlook Web Access. You configure clientless access for a resource by configuring a clientless access policy that is associated with a clientless access profile.

To configure a clientless access policy for a resource in the Citrix Gateway Applications node:

1. In the navigation pane of the GUI, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, in the **Clientless Access** column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a clientless access policy.
3. In the **Configure Clientless Access Policies** dialog box, do the following:
   - To specify an existing clientless access policy, click **Insert Policy**, and then, in the **Policy Name** column, click the name of the policy.
   - To configure a new clientless access policy, click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**. In the **Create Clientless Access Policy** dialog box, in the Name box, after the underscore (_), type a name for the policy. Then, in Profile, either select an existing profile or click New to configure a new profile. You can also select an existing profile and then click **Modify** to modify the profile.
     For more information about configuring a clientless access policy or profile, see Citrix Gateway.
   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click **Modify Policy**. To modify only the associated profile, in the Profile column, click the name of the profile, and then click Modify Profile.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes**, and then click **Close**.

Configuring TCP compression policies

September 14, 2021

You can configure TCP compression policies for an application to increase the performance of the application. TCP compression reduces network latency, reduces bandwidth requirements, and increases the speed of transmission. When configuring a TCP compression policy, you associate a compression action with the policy. The compression action specifies either Compress, GZIP, Deflate, or NoCompress as the compression type. For more information about the compression policies, and compression actions, see Citrix Gateway.

To configure a TCP compression policy for a resource in the Citrix Gateway Applications node

1. In the navigation pane of the GUI, expand **AppExpert**, and then click **Access Gateway Applica-**

tions.

2. In the details pane, in the TCP Compression column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a TCP compression policy.
3. In the **Configure TCP Compression Policies** dialog box, do the following:
   - To specify an existing TCP compression policy, click **Insert Policy**, and then, in the **Policy Name** column, click the name of the policy.
   - To create a new TCP compression policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create TCP Compression Policy dialog box, in the Policy Name box, after the underscore ("_"), type a name for the policy. Then, in Action, either select an existing action or click New and configure a new action. You can also click View to view the configured compression type.
     For more information about configuring a TCP compression policy or action, see Citrix Gateway , Advanced Edition at Citrix Gateway.
   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click **Modify Policy**.
   - To regenerate the priorities assigned to the policies, click **Regenerate Priorities**.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes,** and then click **Close**.

Configure bookmarks

September 14, 2021

You can configure bookmarks for internal applications or resources that are available for an entitled user. You can then bind the bookmark to a user, user group or virtual server globally and enable it for the user in the Access Interface. The bookmark links that you create appear on the web sites panes under enterprise web sites.

For more information, see Creating and Applying Web Links topic.

AppQoE

September 14, 2021

Application level Quality of Experience (AppQoE) integrates several existing policy-based security features of the Citrix ADC appliance into a single integrated feature that takes advantage of a new queuing

mechanism, fair queuing. Fair queuing manages requests to load-balanced web servers and applications at the virtual server level instead of at the service level, allowing it to handle the queuing of all requests to a website or application as one group before load balancing, instead of as separate streams after load balancing.

The features that are integrated into AppQoE are HTTP Denial-of-Service Protection (HDOSP), and Priority Queuing (PQ). Collectively these services provide protection against various problems:

- **Simple overload.** Any server, no matter how robust, can accept only a limited number of connections at one time. When a protected website or application receives too many requests at once, the Surge Protection feature detects the overload and queues the excess connections til the server can accept them. The Priority Queuing feature ensures that whoever most needs access to a resource is provided access without having to wait behind other lower-priority requests. The AppQoE feature displays an alternate webpage that notifies users that the resource that they requested is not available.
- **Denial-of-Service (DOS) attacks.** Any public-facing resource is vulnerable to attacks whose purpose is to bring that service down and deny legitimate users access to it. The Surge Protection, and Priority Queuing features help manage DOS attacks in addition to other types of high load. In addition, the HTTP Denial-of-Service Protection feature targets DOS attacks against your websites, sending challenges to suspected attackers and dropping connections if the clients do not send an appropriate response.

Until the current version of the Citrix ADC operating system, these features were implemented at the service level, which means that each service was assigned its own queues. While service-level queues work, they also have some disadvantages, most of which are due to the Citrix ADC appliance having to load balance requests before implementing any of the protection features that rely on queuing. Implementing protection features before queuing has various advantages, some of which are listed below:

- Absolute priority of connections as configured in the priority queuing feature can be maintained.
- Connections are not flushed if a service transitions state, as they are in a service-level queue.
- During periods of high load, such as a denial-of-service attack, and HTTP DoS come into play before load balancing, allowing these features to detect and divert unwanted or lower-priority traffic from the load balancer before the load balancer must cope with it.

In addition to implementing fair queuing, AppQoE integrates a set of features that each provide a different set of tools to achieve a common goal: protecting your networked resources from excessive or inappropriate demand. Putting these features into a common framework enables you to configure and implement them more easily.

Enabling AppQoE

September 14, 2021

To configure AppQoE, you must first enable the feature.

To enable AppQoE by using the command line

At the command prompt, type the following commands:

- enable ns feature appqoe
- show ns feature

Example:

```
> enable ns feature appqoe
 Done
> show ns feature

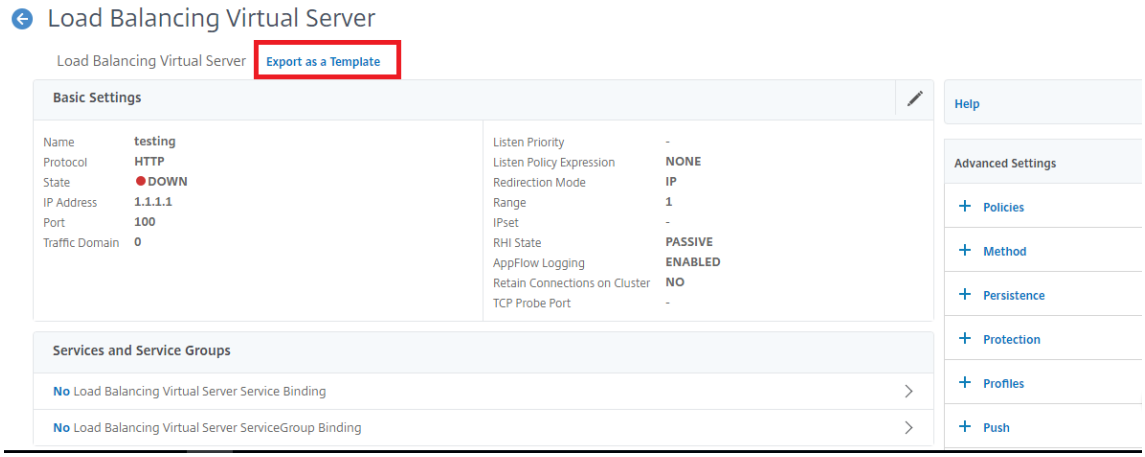
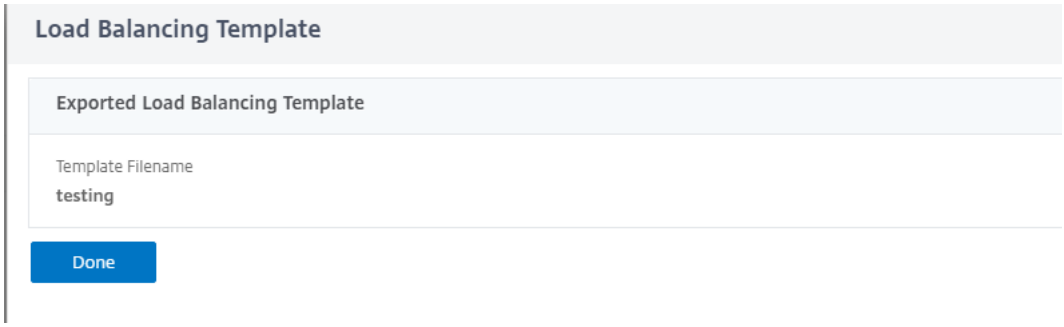
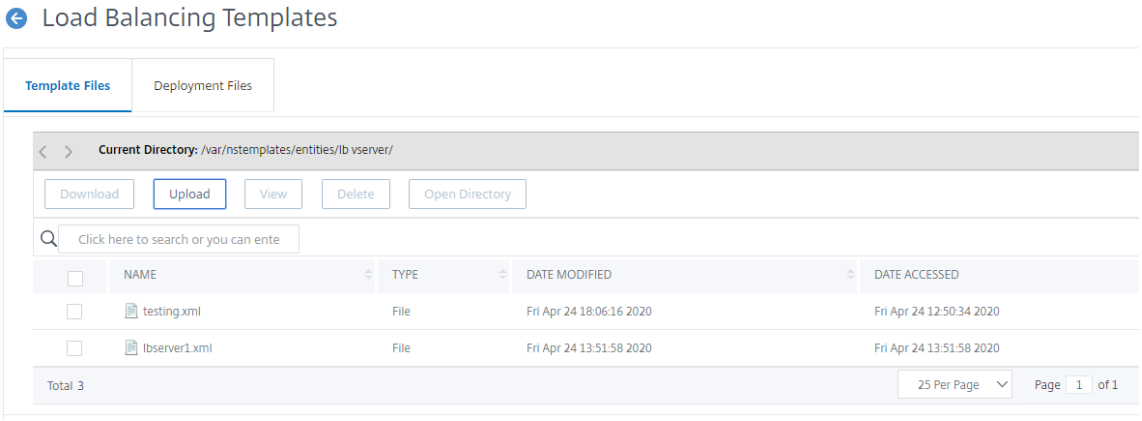
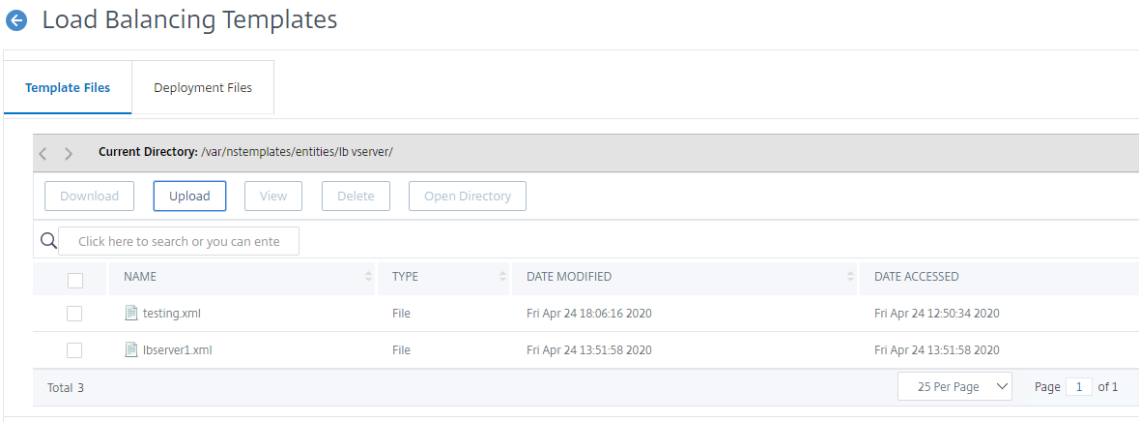
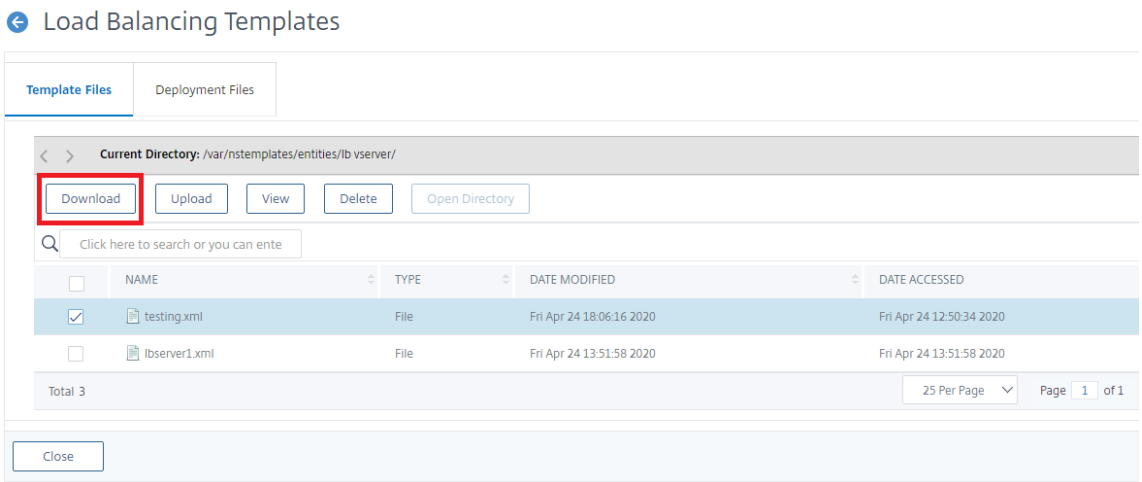
        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 ...
 1)     AppQoE                          AppQoE              ON
 Done
<!--NeedCopy-->
```

To enable AppQoE by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Advanced Features**.
3. In the **Configure Advanced Features** dialog box, select the **AppQoE** check box.
4. Click **OK**.

AppQoE actions

September 14, 2021

After enabling the AppQoE feature, you must configure one or more actions for handling request.

> **Important:**
>
> No specific individual parameters are required to create an action, but you must include at least one parameter or you cannot create the action.

To configure an AppQoE action by using the command line

At the command prompt, type the following commands:

- `add appqoe action <name> [-priority <priority>] [-respondWith (ACS|NS)[<customfile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction (**SimpleResponse** | **HICResponse**)]`
- `show appqoe action`

Example

To configure priority queuing with policy queue depths of 10 and 1000 for medium and lowest priority queues, respectively:

```
> add appqoe action appqoe-act-basic-prhigh -priority HIGH
 Done

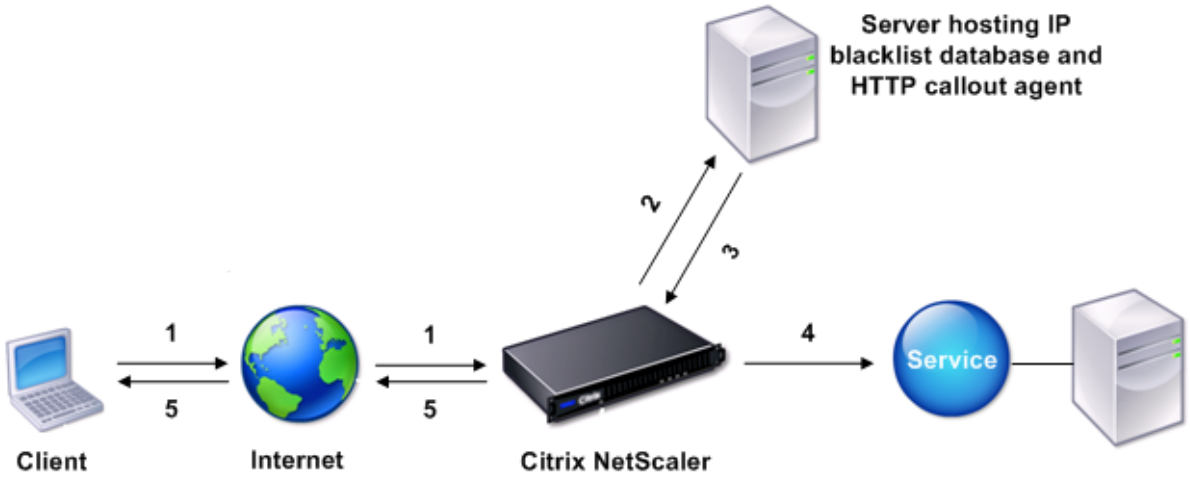
> add appqoe action appqoe-act-basic-prmedium -priority MEDIUM -
   polqDepth 10
 Done

> add appqoe action appqoe-act-basic-prlow -priority LOW -polqDepth
   1000
 Done

> show appqoe action

1.       Name: appqoe-act-basic-prhigh
        ActionType: PRIORITY_QUEUING
        Priority: HIGH
        PolicyQdepth: 0
        Qdepth: 0

1.      Name: appqoe-act-basic-prmedium
        ActionType: PRIORITY_QUEUING
        Priority: MEDIUM
        PolicyQdepth: 10
        Qdepth: 0
```



```
23
24 1.       Name: appqoe-act-basic-prlow
25          ActionType: PRIORITY_QUEUING
26          Priority: LOW
27          PolicyQdepth: 1000
28          Qdepth: 0
29 Done
30 <!--NeedCopy-->
```

To modify an existing AppQoE action by using the command line

At the command prompt, type the following commands:

- `set appqoe action <name> [-priority <priority>] [-altContentSvcName <string>] [-altContentPath <string>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-maxConn <positive_integer>] [-delay <usecs>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse | HICResponse )]`
- `show appqoe action`

To remove an AppQoE action by using the command line

At the command prompt, type the following commands:

- `rm appqoe action <name>`
- `show appqoe action`

Parameters for configuring an AppQoE action

- name. A name for the new action, or the name of the existing action that you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.
- priority. The priority queue to which the request is assigned. When a protected web server or application is heavily loaded and cannot accept additional requests, specifies the order in which waiting requests are to be fulfilled when resources are available. The choices are:

1. **HIGH.** Fulfills the request as soon as resources are available.
2. **MEDIUM.** Fulfills the request after it has fulfilled all requests in the HIGH priority queue.
3. **LOW.** Fulfills the request after it has fulfilled all requests in the HIGH and MEDIUM priority queues.
4. **LOWEST.** Fulfills the request only after it has fulfilled all requests in higher-priority queues.

If priority is not configured, then the Citrix ADC appliance assigns the request to the LOWEST priority queue by default.

- respondWith. Configures the Citrix ADC to take the specified Responder action when the specified threshold is reached. Must be used with one of the following settings:
  - **ACS:** Serves content from an alternate content service. Threshold: maxConn (maximum connections) or delay.
  - **NS:** Serves a built-in response from the Citrix ADC. Threshold: maxConn (maximum connections) or delay.
  - **NO ACTION:** Serves no alternative content. Assigns connections to the LOWEST priority queue if the maxConn (maximum connections) or delay threshold is reached.
- altContentSvcName. If -responseWith ACS is specified, the name of the alternative content service, usually an absolute URL to the web server that hosts the alternate content.

altContentPath. If -responseWith (ACS	NS) is specified, the path to the alternative content.

-
- olqDepth. Policy queue depth threshold value for the policy queue associated with this action. When the number of connections in the policy queue associated with this action increases to the specified number, subsequent requests are assigned to the LOWEST policy queue. Minimum value: 1 Maximum value: 4,294,967,294
- priqDepth. Policy queue depth threshold value for the specified priority queue. If the number of requests in the specified queue on the virtual server to which the policy associated with the current action is bound increases to the specified number, subsequent requests are assigned to the LOWEST priority queue. Minimum value: 1 Maximum value: 4,294,967,294
- maxConn. The maximum number of connections that can be open for requests that match the policy rule. Minimum value: 1 Maximum value: 4,294,967,294
- delay. The delay threshold, in microseconds, for requests that match the policy rule. If a matching request has been delayed for longer than the threshold, the Citrix ADC appliance performs the specified action. If NO ACTION is specified, then the appliance assigns requests to the LOWEST priority queue. Minimum value: 1 Maximum value: 599999,999
- dosTrigExpression. Adds an optional second-level check to trigger DoS actions.
- dosAction. Action to take when the appliance determines that it or a protected server is under DoS attack. Possible values: SimpleResponse, HICResponse.

These values specify HTTP challenge-response methods for validating the authenticity of incoming requests to mitigate an HTTP-DDoS attack.

In the HTTP challenge-response generation and validation process, AppQoE uses cookies to validate the client's response and verify that the client seems to be genuine. When sending a challenge, a Citrix ADC appliance generates two cookies:

Header cookie (_DOSQ). Contains client-specific information, so that the Citrix ADC appliance can verify the response.

Body cookie (_DOSH). Information used to validate the client machine. The client's browser (or the user, in the case of HIC) computes a value for this cookie. The Citrix ADC appliance compares that value with the expected value to verify the client.

The information that the appliance sends to the client for computing the _DOSH value is based on the DoS Action configuration.

1. SimpleResponse: In this case, a Citrix ADC appliance splits the value and generates a JavaScript code to combine the final value. A client machine capable of computing the original value is considered genuine.
2. HICResponse: in this case, a Citrix ADC appliance generates two single-digit numbers and generates images for those numbers. Then, using a backpatch framework, the appliance inserts those images as base64 strings.

Limitations

1. This is not a trivial CAPTCHA implementation, which is why that term not used.
2. The validation number is based on a Citrix ADC-generated number that does not change for 120s. This number should be dynamic or client specific.

To configure an AppQoE action by using the configuration utility

1. Navigate to **App-Expert** > **AppQoE** > **Actions**.
2. In the details pane, do one of the following:
   - To create a new action, click **Add**.
   - To modify an existing action, select the action, and then click **Edit**.
3. In the **Create AppQoE Action** or the **Configure AppQoE Action** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):
   - Name—name
   - Action type—respondWith
   - Priority—priority
   - Policy Queue Depth—polqDepth
   - Queue Depth—priqDepth

- DOS Action—dosAction

4. Click **Create** or **OK**.

AppQoE parameters

September 14, 2021

In the AppQoE parameters, you configure the session life of an AppQoE session, the file name of the file containing the customized response, and the number of client connections that can be placed in a queue.

To configure the AppQoE parameter settings by using the command line

At the command prompt, type the following commands:

- `set appqoe parameter [-sessionLife <secs>] [-avgwaitingclient <positive_integer>] [-MaxAltRespBandWidth <positive_integer>] [-dosAttackThresh <positive_integer>]`
- `show appqoe parameter`

Parameters for configuring the AppQoE parameters

- sessionLife

  Number of seconds to wait after displaying alternate content before the appliance displays the same content again. Default value: 300 Maximum Minimum value: 1 Maximum value: 4,294,967,294

- avgwaitingclient

  The average number of client requests that can be in the service waiting queue. Default value: 1000000 Maximum value: 4,294,967,294

- MaxAltRespBandWidth

  The maximum bandwidth to consume when sending alternate responses. If the maximum is reached, the appliance quits sending the alternate content til bandwidth consumption drops. Default value: 100 Minimum value: 1 Maximum value: 4,294,967,294

- dosAtckThrsh

  The denial-of-service attack threshold. The number of connections that must be waiting in queues before the appliance responds with DoS protection measures. Default value: 2000 Minimum value: 0 Maximum value: 4,294,967,294

To configure the AppQoE parameter settings by using the GUI

1. Navigate to **AppExpert** > **AppQoE**.
2. In the details pane, click **Configure AppQoE Parameters**.
3. In the **Configure AppQoE params** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Parameters" as follows (asterisk indicates a required parameter):
   - Session Life (secs)
   - sessionLife
   - Average waiting client—avgwaitingclient
   - Alternate Response Bandwidth Limit(Mbps) —MaxAltRespBandWidth
   - DOS Attack Threshold —dosAttackThresh
4. Click **OK**.

AppQoE policies

September 14, 2021

To implement AppQoE, you must configure at least one policy to tell your Citrix ADC how to distinguish the connections to be queued in a specific queue.

To configure an AppQoE policy by using the command line

At the command prompt, type the following command:

```
add appqoe policy <name> -rule <expression> -action <string>
```

Example:

The following example selects requests with a User-Agent header that contains "Android," and assigns them to the medium priority queue. These requests come from smartphones and tablets that run the Google Android operating system.

```
1 > add appqoe action appqoe-act-primd -priority MEDIUM
2 Done
3 > add appqoe policy appqoe-pol-primd -rule "HTTP.REQ.HEADER("User-Agent
     ").CONTAINS("Android")" -action appqoe-act-primd
4 Done
5 > sh appqoe policy appqoe-pol-primd
6         Name: appqoe-pol-primd
7         Rule: HTTP.REQ.HEADER("User-Agent").CONTAINS("Android")
8         Action: appqoe-act-primd
9         Hits: 0
```



```
10
11 Done
12 <!--NeedCopy-->
```

Parameters for configuring an AppQoE policy

- name. A name for the AppQoE policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the type of action.
- rule. A Citrix ADC expression that tells the appliance which connections it should handle.
- action. The AppQoE action to perform when a connection matches the policy.

To configure an AppQoE policy by using the configuration utility

1. Navigate to **App-Expert** > **AppQoE** > **Policies**.
2. In the details pane, do one of the following:
   - To create a policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Edit**.
3. If you are creating a policy, in the **Create AppQoE Policy** dialog, in the Name text box, type a name for your new policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the purpose and effect of this policy.

   If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.
4. In the **Action** drop-down list, choose the AppQoE action to perform when the policy matches a connection. Click the plus (+) to open the **Add AppQoE Action** dialog and add a new action.
5. In the **Rule** text box, either enter the policy expression directly, or click New to create a policy expression. If you click New, perform the following steps:

   a) In the **Create Expression** dialog box, click **Add**.
   1 In the **Add Expression** dialog box, select a common expression from the **Frequently Used Expressions** drop-down list, or use the **Construct Expression** drop-down lists to create the expression that defines which traffic to filter.
   If you choose to create your own expression, you start by selecting the first term from the

first drop-down list on the left side of the Construct Expression area. The choices in that list are:

- HTTP
- SYS
- CLIENT
- SERVER
- ANALYTICS
- TEXT

The default choice is HTTP. After you make a choice in the first drop-down list (or accept the default), you can choose the next term in your expression from the drop-down list to the right of it. The terms in that list and other lists that follow change depending on your previous choices. The lists offer only terms that are valid choices. Continue to select terms until you have finished the expression.

a) When you have created the expression that you want, click **OK**. The expression is added in the **Expression** text box.

6. Click **Create**. The expression appears in the **Rule** text box.

Entity template for load balancing virtual server

September 14, 2021

> **Warning**
>
> The entity template functionality is deprecated from Citrix ADC 13.0 build 82.x onwards and as an alternative Citrix recommends you to use the Style Books. For more information, see Style Books topic.

An entity template is a collection of information for creating a load balancing virtual server template on a Citrix ADC appliance. It provides a specification and a set of defaults to be configured for a load balancing virtual server. By using a template that defines a set of defaults, you can quickly configure multiple virtual servers that require a similar configuration while eliminating several configuration steps.

You can create an entity template by exporting the load balancing virtual server details to a template file. This can done only through the Citrix ADC GUI. You use the Citrix ADC GUI to export, import, and manage entity templates. You can share entity templates with other administrators and manage templates saved locally on your appliance or machine. You can also import entity templates from the appliance or your local computer.

Before creating a template, you should be familiar with the configuration of the load balancing virtual server.

Load balancing virtual server template

Load balancing entity templates are created in the same way that Citrix ADC application templates are created. When you export a load balancing virtual server to a template file, the following two files are automatically created:

- Load balancing virtual server template file. Contains XML elements that store the values of the parameters that are configured for the load balancing virtual server. The file also contains XML elements for storing information about bound policies.
- Deployment file. Contains XML elements that store deployment-specific information such as services, service groups, and configured variables.
  In the template and deployment files, each unit of configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, the load balancing method parameter, lbMethod, is encapsulated within the <lbmethod> and </lbmethod> tags.

> **Note:**
>
> After you export a load balancing virtual server, you can add elements, remove elements, and modify existing elements before importing the configuration information to a Citrix ADC appliance.

How a load balancing virtual server template works

When you create a template for a load balancing virtual server, you specify default values for the server. You specify what values must be read-only, what values must not be displayed, and what values users can configure. You also configure the pages that compose the template import wizard. All the information and settings you provide are stored in the template file.
When a user imports the template to a Citrix ADC appliance, the GUI guides the user through the various pages that you configured for the template. The GUI displays the read-only parameter values and prompts the user to specify values for the configurable parameters. After the user follows the instructions, the appliance creates the entity with the configured values.
You can create or modify an entity template for a load balancing virtual server from the Traffic Management node.
To export virtual server details to a template, you must specify the following options and settings for the template:

- The default value of a parameter.
- Whether the default values are visible to users.
- Whether the default values can be changed by users.

- The number of pages in the entity import wizard, including the page names, text, and available parameters.
- The entities that must be bound to the entity for which the template is being created.

For example, when you are creating a load balancing virtual server template, you can specify the policies that you want to bind to the virtual server that you create from the template. However, only binding information is included in the template. The bound entities are not included. If the entity template is imported to another Citrix ADC appliance, the bound entities must exist on the appliance at import time for the binding to succeed. If none of the bound entities exist on the target appliance, the entity (for which the template was configured) is created without any bindings. If only a subset of the bound entities exist on the target appliance, they are bound to the entity that is created from the template.
When you export a template for the load balancing virtual server, the configuration settings of the entity appear in the template. All bound entities are selected by default, but you can modify bindings as necessary. As in the case of a template that is not based on an existing entity, only binding information is included and not the entities. You can either save the template with the existing configuration settings or use the settings as a basis for creating a new configuration for a template.

Configure variables in load balancing virtual server template

Load balancing virtual server templates support the declaration of variables in the configured load balancing parameters and in bound policies and actions. The ability to declare variables enables you to replace preconfigured values with values that suit the environment into which you are importing the template.

As an example, consider the following expression configured for a policy that is bound to a load balancing virtual server for which you are creating a template. The expression evaluates the value of the accept-language header in an HTTP request.

```
HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")
```

If you want the value of the header to be configurable at import time, you can specify the string en-us as a variable.

After you create a variable, you can do the following:

- Assign more strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable

To configure variables in a load balancing virtual server template

Complete the following procedure to configure variables for a load balancing virtual server template by using the Citrix ADC GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**
2. In the details pane, right-click the virtual server that you want to export to a template file, and then click **Add**.
3. In the **Create Load balancing Virtual Server** page, set the virtual server parameters. For more information on configuring a load balancing virtual server, see How load balancing works
4. Once you have the set the parameters for the load balancing virtual server, click **Done**.



5. Click **Export as Template** link at the top to export the server details as a template file.
6. In the **Create Load Balancing Template** page, enter the template settings.
7. Click **Done**.



Modify a load balancing virtual server template

You can modify only the parameters, bindings, and pages configured for a template. The name and location of the template specified when the template was created cannot be changed. The Citrix ADC appliance does not provide you with the option of modifying a load balancing virtual server template.

To modify a load balancing virtual server by using Citrix ADC GUI

1. Navigate to **Traffic Management** > **Load balancing** > **Virtual Servers**.
2. In the **Load Balancing Virtual Server** page, modify the entity parameters.
3. Click Done.
4. Click **Export as a Template** link.
5. The modified changes are now available in the load balancing virtual server template file.
6. In the **Exported Load Balancing Template** page, click **Done**.

Manage load balancing virtual server templates

You can organize load balancing virtual server template files and deployment files by using the Citrix ADC GUI.

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the **Virtual Servers** page, select **Manage Template action**.
3. In the **Load Balancing Templates** page, click **Template File** tab.
4. In the **Template Files** tab page, you can upload or download a template from and to the appliance template folder.



5. Click **Close**.

To upload load balancing virtual server entity template by using Citrix ADC GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the **Virtual Servers** page, click **Select Action** and then select **Manage Template**.
3. In the Load Balancing Templates page, click **Template Files** tab.
4. In the **Template Files** tab page, click **Upload** to upload a template.

5. Click **Close**.



To download load balancing virtual server entity template by using Citrix ADC GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the **Virtual Servers** page, click **Select Action** and then select **Manage Template**.
3. In the **Load Balancing Templates** page, click **Template Files** tab.
4. In the Template Files tab page, select a template file and click Download.
5. Click Close.



Example of load balancing virtual server template and deployment template

Following is an example of a template file that was created from a load balancing virtual server called “Lbvip”:

```
1 COPY
2
```



```
<?xml version="1.0" encoding="UTF-8" ?>
  <template>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <entitytemplate>
      <lbvserver_list>
        <lbvserver>
          <name>Lbvip</name>
          <servicetype>HTTP</servicetype>
          <ipv46>0.0.0.0</ipv46>
          <ipmask>*</ipmask>
          <port>0</port>
          <range>1</range>
          <persistencetype>NONE</persistencetype>
          <timeout>2</timeout>
          <persistencebackup>NONE</persistencebackup>
          <backuppersistencetimeout>2</backuppersistencetimeout>
          <lbmethod>LEASTCONNECTION</lbmethod>
          <persistmask>255.255.255.255</persistmask>
          <v6persistmasklen>128</v6persistmasklen>
          <pq>OFF</pq>
          <sc>OFF</sc>
          <m>IP</m>
          <datalength>0</datalength>
          <dataoffset>0</dataoffset>
          <sessionless>DISABLED</sessionless>
          <state>ENABLED</state>
          <connfailover>DISABLED</connfailover>
          <clttimeout>180</clttimeout>
          <somethod>NONE</somethod>
          <sopersistence>DISABLED</sopersistence>
          <sopersistencetimeout>2</sopersistencetimeout>
          <redirectportrewrite>DISABLED</redirectportrewrite>
          <downstateflush>DISABLED</downstateflush>
          <gt2gb>DISABLED</gt2gb>
          <ipmapping>0.0.0.0</ipmapping>
          <disableprimaryondown>DISABLED</disableprimaryondown>
          <insertvserveripport>OFF</insertvserveripport>
          <authentication>OFF</authentication>
          <authn401>OFF</authn401>
          <push>DISABLED</push>
```



```
            <pushlabel>none</pushlabel>
            <l2conn>OFF</l2conn>
            <appflowlog>DISABLED</appflowlog>
            <icmpvsrresponse>PASSIVE</icmpvsrresponse>
            <lbvserver_cmppolicy_binding_list>
              <lbvserver_cmppolicy_binding>
                <name>Lbvip</name>
                <policyname>NOPOLICY-COMPRESSION</policyname>
                <priority>100</priority>
                <gotoprioritypressionexpression>END</gotoprioritypressionexpression>
                <bindpoint>REQUEST</bindpoint>
              </lbvserver_cmppolicy_binding>
            </lbvserver_cmppolicy_binding_list>
          </lbvserver>
        </lbvserver_list>
      </entitytemplate>
    </template>
<!--NeedCopy-->
```

Example of a deployment file

Following is the deployment file associated with the virtual server in the preceding example:
COPY

```
<?xml version="1.0" encoding="UTF-8" ?>
  <template_deployment>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <service_list>
      <service>
      <ip>1.2.3.4</ip>
      <port>80</port>
      <servicetype>HTTP</servicetype>
      </service>
    </service_list>
    <servicegroup_list>
      <servicegroup>
        <name>svcgrp</name>
        <servicetype>HTTP</servicetype>
        <servicegroup_servicegroupmember_binding_list>
```



```
            <servicegroup_servicegroupmember_binding>
              <ip>1.2.3.90</ip>
              <port>80</port>
            </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.8.0</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.8.1</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.9.0</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
        </servicegroup_servicegroupmember_binding_list>
      </servicegroup>
    </servicegroup_list>
</template_deployment>

<!--NeedCopy-->
```

HTTP callouts

September 14, 2021

For certain types of requests, or when certain criteria are met during policy evaluation, you might want to stall policy evaluation briefly, retrieve information from a server, and then perform a specific action that depends on the information that is retrieved. At other times, when you receive certain types of requests, you might want to update a database or the content hosted on a Web server. HTTP callouts enable you to perform all these tasks.

An HTTP callout is an HTTP or HTTPS request that the Citrix ADC appliance generates and sends to an external application when certain criteria are met during policy evaluation. The information that is retrieved from the server can be analyzed by default syntax policy expressions, and an appropriate action can be performed. You can configure HTTP callouts for HTTP content switching, TCP content switching, rewrite, responder, and for the token-based method of load balancing.

Before you configure an HTTP callout, you must set up an application on the server to which the callout will be sent. The application, which is called the *HTTP callout agent* , must be configured to respond to the HTTP callout request with the required information. The HTTP callout agent can also

be a Web server that serves the data for which the Citrix ADC appliance sends the callout. You must make sure that the format of the response to an HTTP callout does not change from one invocation to another.

After you set up the HTTP callout agent, you configure the HTTP callout on the Citrix ADC appliance. Finally, to invoke the callout, you include the callout in a default syntax policy in the appropriate Citrix ADC feature and then bind the policy to the bind point at which you want the policy to be evaluated.

After you have configured the HTTP callout, you must verify the configuration to make sure that the callout is working correctly.

How an HTTP callout works

September 14, 2021

When the Citrix ADC appliance receives a client request, the appliance evaluates the request against the policies bound to various bind points. During this evaluation, if the appliance encounters the HTTP callout expression, `SYS.HTTP_CALLOUT(<name>)`, it stalls policy evaluation briefly and sends a request to the HTTP callout agent by using the parameters configured for the specified HTTP callout. Upon receiving the response, the appliance inspects the specified portion of the response, and then either performs an action or evaluates the next policy, depending on whether the evaluation of the response from the HTTP callout agent evaluates to TRUE or FALSE, respectively. For example, if the HTTP callout is included in a responder policy, if the evaluation of the response evaluates to TRUE, the appliance performs the action associated with the responder policy.

If the HTTP callout configuration is incorrect or incomplete, or if the callout invokes itself recursively, the appliance raises an UNDEF condition, and updates the undefined hits counter.

The following figure illustrates the working of an HTTP callout that is invoked from a globally bound responder policy. The HTTP callout is configured to include the IP address of the client that is associated with an incoming request. When the Citrix ADC appliance receives a request from a client, the appliance generates the callout request and sends it to the callout server, which hosts a database of blacklisted IP addresses and an HTTP callout agent that checks whether the client's IP address is listed in the database. The HTTP callout agent receives the callout request, checks whether the client's IP address is listed, and sends a response that the Citrix ADC appliance evaluates. If the response indicates that the client's IP address is not blacklisted, the appliance forwards the response to the configured service. If the client's IP address is blacklisted, the appliance resets the client connection

Figure 1. HTTP Callout Entity Model



1: Client request
2: HTTP callout request to check whether the client is blacklisted
3: Response from HTTP callout agent
4: Request forwarded to service if 3 indicates a safe IP address
5: Connection RESET if 3 indicates a bad IP address

Notes on the format of HTTP requests and responses

September 14, 2021

The Citrix ADC appliance does not check for the validity of the HTTP callout request. Therefore, before you configure HTTP callouts, you must know the format of an HTTP request. You must also know the format of an HTTP response, because configuring an HTTP callout involves configuring expressions that evaluate the response from the HTTP callout agent.

This section includes the following sections:

- Format of an HTTP Request
- Format of an HTTP Response

Format of an HTTP Request

An HTTP request contains a series of lines that each end with a carriage return and a line feed, represented as either `<CR><LF>` or `\r\n`.

The first line of a request (the *message line*) contains the HTTP method and target. For example, a message line for a GET request contains the keyword GET and a string that represents the object that is to be fetched, as shown in the following example:


```
GET /mysite/mydirectory/index.html HTTP/1.1\r\n
<!--NeedCopy-->
```

The rest of the request contains HTTP headers, including a required Host header and, if applicable, a message body.

The request ends with a bank line (an extra <CR><LF> or \r\n).

Following is an example of a request:

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: */*\r\n
\r\n
<!--NeedCopy-->
```

Format of an HTTP Response

An HTTP response contains a status message, response HTTP headers, and the requested object or, if the requested object cannot be served, an error message.

Following is an example of a response:

```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag:  “04f97692cbd1:377” \r\n
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
<55-character response>
<!--NeedCopy-->
```

Configuring an HTTP callout

September 14, 2021

When configuring an HTTP callout, you specify the type of request (HTTP or HTTPS), destination, and format of the request. The expected format of the response, and, finally, the portion of the response that you want to analyze.

For the destination, you either specify the IP address and port of the HTTP callout agent. Or engage a load balancing, content switching, or cache redirection virtual server, to manage the HTTP callout requests.

In the first case, the HTTP callout requests are sent directly to the HTTP callout agent. In the second case, the HTTP callout requests are sent to the virtual IP address (VIP) of the specified virtual server. The virtual server processes the request in the same way as it processes a client request. For example, if you expect many callouts to be generated, you can configure instances of the HTTP callout agent on multiple servers, bind these instances (as services) to a load balancing virtual server, and then specify the load balancing virtual server in the HTTP callout configuration. The load balancing virtual server then balances the load on those configured instances as determined by the load balancing algorithm.

For the format of the HTTP callout request, you can specify the individual attributes of the HTTP callout request (an attribute-based HTTP callout), or you can specify the entire HTTP callout request as a advanced policy expression (an expression-based HTTP callout).

The following table describes the elements in an HTTP callout policy:

For more information, see policy-httpCallout

Parameter	Description
Name	Name of the callout, 127 character maximum
IP address and port (ip-address / *port*) or Virtual server name (vserver)	IPv4 or IPv6 address of the server to which the callout is sent, or a wildcard, and the port on the server to which the callout is sent, or a wildcard. Or, the name of a load balancing, content switching, or cache redirection virtual server with a service type of HTTP.
HTTP Method (httpMethod)	HTTP Method (httpMethod). Method used in the HTTP request that this callout sends. Valid values: GET or POST. Default: GET.
Host expression (hostExpr)	Host expression (hostExpr). Advanced text expression to configure the Host header. Maximum length: 255. The expression can be a literal value or it can be an advanced expression that derives the value. Examples: “10.101.10.11” , “http.req.header(”Host”)”

Parameter	Description
URL stem expression (urlStemExpr)	URL stem expression (urlStemExpr) An advanced string expression for generating the URL stem. Maximum length: 8191. The expression can be a literal string or an expression that derives the value. Examples: “”/mysite/index.html”” “http.req.url”
HTTP Headers (headers)	HTTP Headers (headers). Advanced text expression to insert HTTP headers and their values in the HTTP callout request. Specify a value for every header. You specify the header name as a string and the header value as an advanced expression. Specify the headers separated by space. Such as -headers cip(client.ip.src) hdr(http.req.header(“HDR”)). The number of headers can be 8
Expression-based request to send to the server (fullReqExpr)	Exact HTTP request that the Citrix ADC is to send as an advanced expression to 8191 characters. If you specify this parameter, you must omit the httpMethod, hostExpr, urlStemExpr, headers, and parameters arguments. The request expression is constrained by the feature where the callout is used. For example, an HTTP.RES expression cannot be used in a request-time policy bank or in a TCP content switching policy bank.
Expression-based request to send to the server (bodyExpr)	An advanced string expression for generating the body of the request. The expression can contain a literal string or an expression that derives the value (for example, client.ip.src). Mutually exclusive with -fullReqExpr.

Parameter	Description
Parameters	Advanced expression to insert query parameters in the HTTP request that the callout sends. Specify a value for every parameter that you configure. If the callout request uses the GET method, these parameters are inserted in the URL. If the callout request uses the POST method, these parameters are inserted in the POST body. You configure the query parameter name as a string and the value as an advanced expression. The parameter values are URL encoded. Specify the parameters separated by space like: -parameters name1(“name1”) name2(http.req.header(“hdr”)). The maximum 8 parameters can be configured.
Return type (returnType)	Type of data that the target application returns in the response to the callout. Valid values: TEXT: Treat the returned value as a text string. NUM: Treat the returned value as a number. BOOL: Treat the returned value as a Boolean value. Note: You cannot change the return type after it is set.
Expression to extract data from the response (resultExpr)	Advanced expression that extracts HTTP.RES objects from the response to the HTTP callout. The maximum length is 8191. The operations in this expression must match the return type. For example, if you configure a return type of text, the result expression must be a text-based expression. If the return type is num, the result expression (resultExpr) must return a numeric value similar to the following: “http.res.body(10000).length” Note: Sometimes, if you set a return type of TEXT and the result sent from the server exceeds 16 KB, the result expression can return NULL. For example, when the result is a concatenated string that exceeds 16 KB.

Parameter	Description
Scheme	The type of scheme for the callout server. Example: HTTP, https
cacheForSecs	Duration, in seconds, for which the callout response is cached. The cached responses are stored in an integrated caching content group named “calloutContentGroup”. If no duration is configured, the callout responses are not cached unless a normal caching configuration is used to cache them. This parameter takes precedence over any normal caching configuration that would otherwise apply to these responses.

Note: The appliance does not check for the validity of the request. You must make sure that the request is a valid request and doesn’t contain any confidential information. An incorrect or incomplete HTTP callout configuration results in a runtime UNDEF condition that is not associated with an action. The UNDEF condition merely updates the Undefined Hits counter, which enables you to troubleshoot an incorrectly configured HTTP callout. However, the appliance parses the HTTP callout request to enable you to configure certain Citrix ADC features for the callout. This can lead to an HTTP callout invoking itself. For information about callout recursion and how you can avoid it, see Avoiding HTTP Callout Recursion.

Finally, regardless of whether you use HTTP request attributes or an expression to define the format of the HTTP callout request, you must specify the format of the response from the HTTP callout agent and the portion of the response that you want to evaluate. The response type can be a Boolean value, a number, or text. Based on this return type only, you can use the further expression methods on the callout response. If the return type is a number, then you can use the number based expression on the callout response. The portion of the response that you want to evaluate is specified by an expression. For example, if you specify that the response contains text, you can use `HTTP.RES.BODY(<unit>)` to specify that the appliance must evaluate only the first <unit> bytes of the response from the callout agent.

At the command line, you first create an HTTP callout by using the
add command. When you add a callout, all parameters are set to a default value of NONE, except the HTTP method, which is set to a default value of GET. You then configure the callout’s parameters by using the set command. The set command is used to configure both types of callouts (attribute-based and expression-based). The difference lies in the parameters that are used for configuring the two types of callouts. So, the command-line instructions that follow include a set command for configur-

ing an attribute-based callout and a set command for configuring an expression-based callout. In the configuration utility, all of these configuration tasks are performed in a single dialog box.

Note: Before you put an HTTP callout into a policy, you can modify all configured parameters except the return type. Once an HTTP callout is in a policy, you cannot completely modify an expression that is configured in the callout. For example, you cannot change HTTP.REQ.HEADER("myval") to CLIENT.IP.SRC. You can modify the operators and arguments that are passed to the expression. For example, you can change `HTTP.REQ.HEADER("myVal1")` to `HTTP.REQ.HEADER("myVal2")`, or `HTTP.REQ.HEADER("myVal")` to `HTTP.REQ.HEADER("myVal").AFTER_STR(<string>)`. If the set command fails, create an HTTP callout.

HTTP callout configuration involves configuring advanced policy expressions. For more information about configuring advanced policy expressions, see Configuring advanced policy expression: getting started topic.

To configure an HTTP callout by using the command line interface

At the command prompt, do the following:

Create an HTTP callout.

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port<
    port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
    GET | POST )] [-hostExpr <expression>]  [-urlStemExpr <expression>]
    [-headers <name(value)> ...] [-parameters <name(value)> ...] [-
    bodyExpr <expression>] [-fullReqExpr <expression>] [-scheme ( http |
     https )] [-resultExpr <expression>] [-cacheForSecs <secs>] [-
    comment <string>]

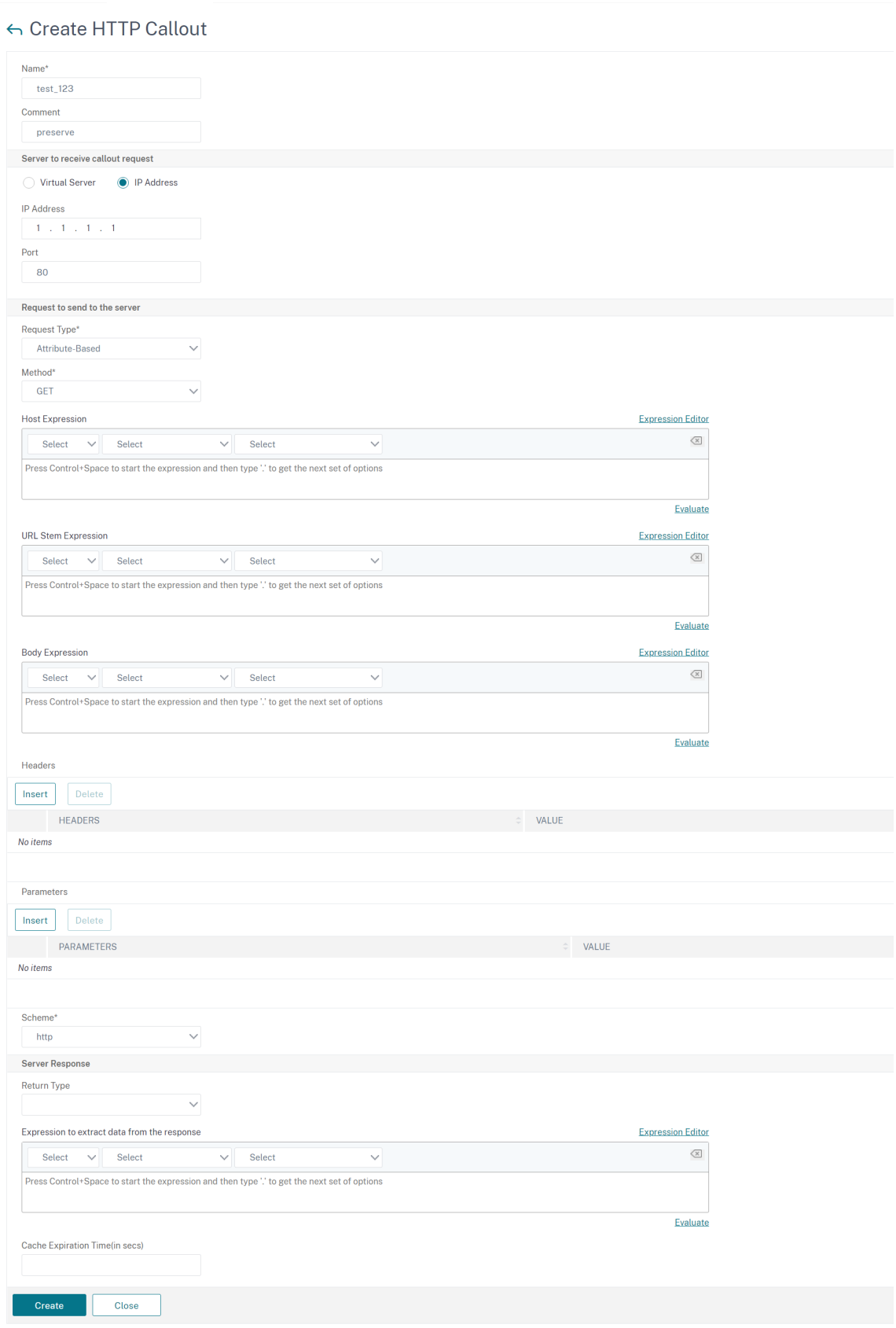
<!--NeedCopy-->
```

Example:

```
add policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'-urlStemExpr "http
    .req.url" -parameters Name("My Name") -headers Name("MyHeader")-
    resultExpr "http.res.body(10000).length"

<!--NeedCopy-->
```

Modify the HTTP callout configuration.

```
set policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-port
     <port|*>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <
```



```
    string>] [-headers <name(value)> ...] [-parameters <name(value)>
    ...] [-resultExpr <string>]

<!--NeedCopy-->
```

Example:

```
> set policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'-urlStemExpr "http
    .req.url" -parameters Name("My Name") -headers Name("MyHeader") -
    resultExpr "http.res.body(10000).length"
<!--NeedCopy-->
```

Configure HTTP callout using fullReqExpr parameter.

```
set policy httpCallout <name> [-vServer <string>] [-returnType <
    returnType>] [-fullReqExpr <string>] [-resultExpr <string>]
<!--NeedCopy-->
```

Example:

```
> set policy httpCallout mycallout1 -vserver lbv1 -returnType num
    fullReqExpr q{
 "GET " + http.req.url + "HTTP/" + http.req.version.major + "." + http.
    req.version.minor.sub(1) + "r\nHost:10.101.10.10\r\nAccept: */*\r\n
    \r\n" }


<!--NeedCopy-->
```

Verify the configurations of the HTTP callout.

```
show policy httpCallout `<name>`

sh policy httpCallout mycallout1
> Name: mycallout1
>Vserver: lbv1 (UP)
Effective Vserver state: UP
Return type: TEXT
Scheme: HTTP
Full REQ expr: "GET " + http.req.url + "HTTP/" + http.req.version.major
    + "." + http.req.version.minor.sub(1)+ "r\nHost:10.101.10.10\r\
    nAccept: */*\r\n\r\n"
Result expr: http.res.body(100)
Hits: 0
```



```
Undef Hits: 0
Done
>

<!--NeedCopy-->
```

To configure an HTTP callout by using the configuration utility

1. Navigate to **AppExpert** > **HTTP Callouts**.
2. In the details pane, click **Add**.
3. In the **Create HTTP Callout** dialog box, configure the parameters of the HTTP callout. For a description of the parameter, hover the mouse cursor over the check box.
4. Click **Create** and then click **Close**.



Verifying the configuration

September 14, 2021

For an HTTP callout to work correctly, all the HTTP callout parameters and the entities associated with the callout must be configured correctly. While the Citrix ADC appliance does not check the validity of the HTTP callout parameters, it indicates the state of the bound entities, namely the server or virtual server to which the HTTP callout is sent. The following table lists the icons and describes the conditions under which the icons are displayed.

Icon	Indicates that
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is UP.
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is OUT OF SERVICE.
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is DOWN.

Table 1. Icons That Indicate the States of Entities Bound to an HTTP Callout

For an HTTP callout to function correctly, the icon must be green at all times. If the icon is not green, check the state of the callout server or virtual server to which the HTTP callout is sent. If the HTTP callout is not working as expected even though the icon is green, check the parameters configured for the callout.

You can also verify the configuration by sending test requests that match the policy from which the HTTP callout is invoked, checking the hits counter for the policy and the HTTP callout, and verifying the responses that the Citrix ADC appliance sends to the client.

Note: An HTTP callout can sometimes invoke itself recursively a second time. If this happens, the hits counter is incremented by two counts for each callout that is generated by the appliance. For the hits counter to display the correct value, you must configure the HTTP callout in such a way that it does not invoke itself a second time. For more information about how you can avoid HTTP callout recursion,

see Avoiding HTTP Callout Recursion.

To view the hits counter for an HTTP callout

1. Navigate to **AppExpert** > **HTTP Callouts**.
2. In the details pane, click the HTTP callout for which you want to view the hits counter, and then view the hits in the **Details** area.

Invoking an HTTP Callout

September 14, 2021

After you configure an HTTP callout, you invoke the callout by including the `SYS.HTTP_CALLOUT(<name>)` expression in a default syntax policy rule. In this expression, `<name>` is the name of the HTTP callout that you want to invoke.

You can use default syntax expression operators with the callout expression to process the response and then perform an appropriate action. The return type of the response from the HTTP callout agent determines the set of operators that you can use on the response. If the part of the response that you want to analyze is text, you can use a text operator to analyze the response. For example, you can use the CONTAINS(<string>) operator to check whether the specified portion of the response contains a particular string, as in the following example:

```
1 SYS.HTTP_CALLOUT(mycallout).contains("Good IP address")
2 <!--NeedCopy-->
```

If you use the preceding expression in a responder policy, you can configure an appropriate responder action.

Similarly, if the part of the response that you want to evaluate is a number, you can use a numeric operator such as GT(int). If the response contains a Boolean value, you can use a Boolean operator.

Note: An HTTP callout can invoke itself recursively. HTTP callout recursion can be avoided by combining the HTTP callout expression with a default syntax expression that prevents recursion. For information about how you can avoid HTTP callout recursion, see Avoiding HTTP Callout Recursion.

You can also cascade HTTP callouts by configuring policies that each invoke a callout after evaluating previously generated callouts. In this scenario, after one policy invokes a callout, when the Citrix ADC appliance is parsing the callout before sending the callout to the callout server, a second set of policies can evaluate the callout and invoke additional callouts, which can in turn be evaluated by a third set of policies, and so on. Such an implementation is described in the following example.

First, you could configure an HTTP callout called myCallout1, and then configure a responder policy, Pol1, to invoke myCallout1. Then, you could configure a second HTTP callout, myCallout2, and a responder policy, Pol2. You configure Pol2 to evaluate myCallout1 and invoke myCallout2. You bind both responder policies globally.

To avoid HTTP callout recursion, myCallout1 is configured with a unique custom HTTP header called "Request1." Pol1 is configured to avoid HTTP callout recursion by using the default syntax expression,

```
HTTP.REQ.HEADER("Request1").EQ("Callout Request").NOT.
<!--NeedCopy-->
```

Pol2 uses the same default syntax expression, but excludes the .NOT operator so that the policy evaluates myCallout1 when the Citrix ADC appliance is parsing it. Note that myCallout2 identifies its own unique header called "Request2," and Pol2 includes a default syntax expression to prevent myCallout2 from invoking itself recursively.

Example:

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -
    returnType TEXT -hostExpr
 ""10.102.3.95"" -urlStemExpr "\"/cgi-bin/check_clnt_from_database.pl\"
     " -headers Request1
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.
    RES.BODY(100)"

Done

> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout
     Request\").NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS(\"IP Matched\")" RESET

Done

> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

```



```
> set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -
    returnType TEXT -hostExpr
"\"10.102.3.96\"" -urlStemExpr "\"/cgi-bin/
    check_clnt_location_from_database.pl\"" -headers Request2
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.
    RES.BODY(200)"

 Done

> add responder policy Pol2 "HTTP.REQ.HEADER(\"Request2\").EQ(\"Callout
     Request\").NOT &&
 HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\") && SYS.
    HTTP_CALLOUT(myCallout2).CONTAINS
(\"APAC\")" RESET

 Done

> bind responder global Pol2 110 END -type OVERRIDE

Done
<!--NeedCopy-->
```

Avoiding HTTP callout recursion

September 14, 2021

Even though the Citrix ADC appliance does not check for the validity of the HTTP callout request, it parses the request once before it sends the request to the HTTP callout agent. This parsing allows the appliance to treat the callout request as any other incoming request, which in turn allows you to configure several useful Citrix ADC features (such as integrated caching) to work on the callout request.

However, during this parsing, the HTTP callout request can select the same policy and therefore invoke itself recursively. The appliance detects the recursive invocation and raises an undefined (UNDEF) condition. However, the recursive invocation results in the policy and HTTP callout select counters being incremented by two counts each instead of one count each.

To prevent a callout from invoking itself, you must identify at least one unique characteristic of the HTTP callout request, and then exclude all requests with this characteristic from being processed by the policy rule that invokes the callout. You can do so by including another default syntax expression in the policy rule. The expression must precede the `SYS.HTTP_CALLOUT(<name>)` expression so that it is evaluated before the callout expression is evaluated. For example:


```
<Expression that prevents callout recursion> OR SYS.HTTP_CALLOUT(<name
   >)
<!--NeedCopy-->
```

When you configure a policy rule in this way, when the appliance generates the request and parses it, the compound rule evaluates to FALSE, the callout is not generated a second time, and the select counters are incremented correctly.

One way by which you can assign a unique characteristic to an HTTP callout request is to include a unique custom HTTP header when you configure the callout. Following is an example of an HTTP callout called “myCallout.” The callout generates an HTTP request that checks whether a client’s IP address is present in a database of blacklisted IP addresses. The callout includes a custom header called “Request,” which is set to the value “Callout Request.” A globally bound responder policy, “Pol1,” invokes the HTTP callout but excludes all requests whose Request header is set to this value, thus preventing a second invocation of myCallout. The expression that prevents a second invocation is HTTP.REQ.HEADER(“Request”).EQ(”Callout Request”).NOT.

Example:

```
> add policy httpCallout myCallout
 Done

> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -
   returnType TEXT -hostExpr "\"10.102.3.95\"" -urlStemExpr "\"/cgi-bin
   /check_clnt_from_database.pl\"" -headers Request("Callout Request")
   -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"
Done

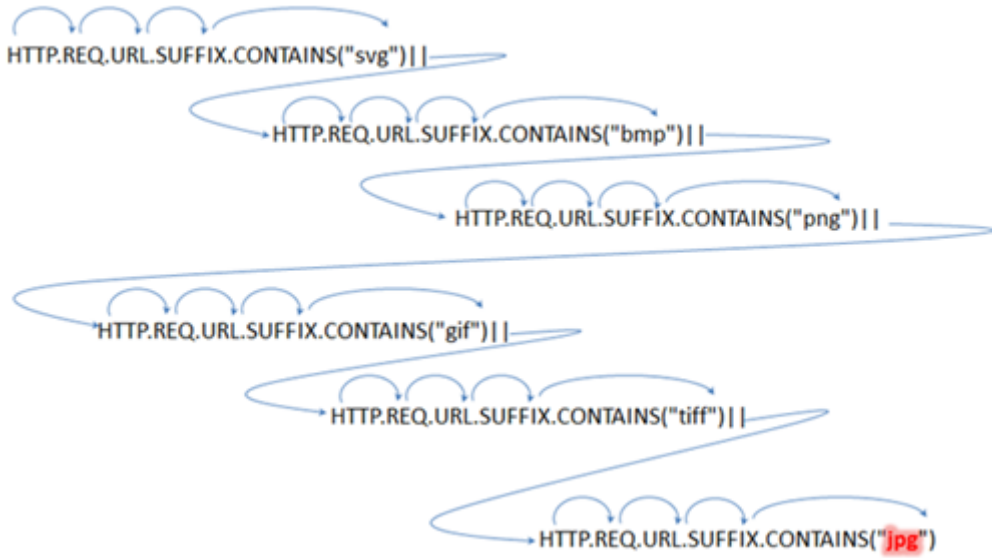
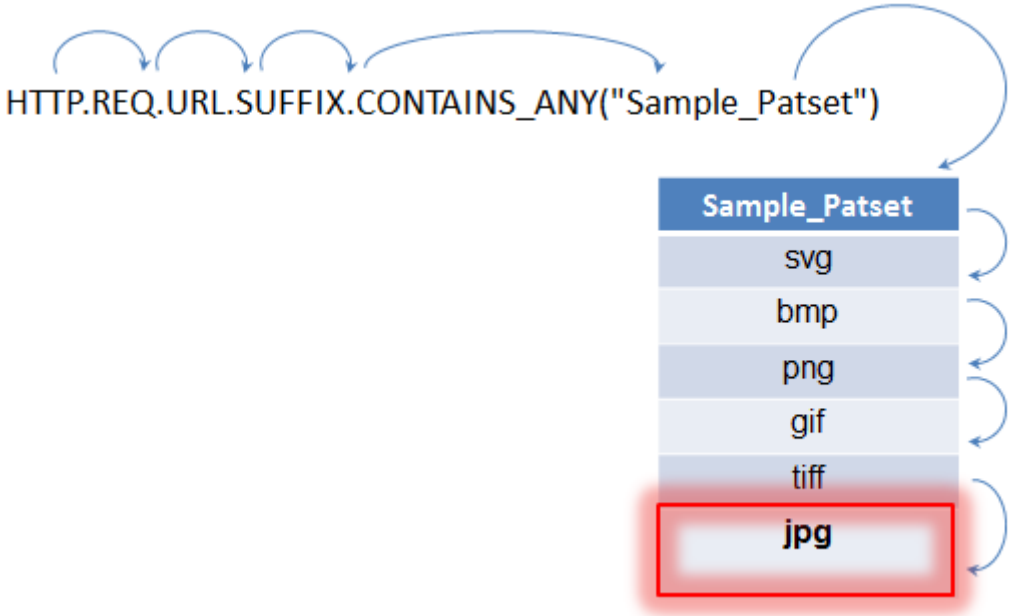
> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout
   Request\").NOT && SYS.HTTP_CALLOUT(myCallout).CONTAINS(\"IP Matched
   \")" RESET
Done

> bind responder global Pol1 100 END -type OVERRIDE
Done
<!--NeedCopy-->
```

> **Note:**
> You can also configure an expression to check whether the request URL includes the stem expression configured for the HTTP callout. To implement the solution, ensure the HTTP callout agent can respond only to HTTP callouts and not to other requests directed through the appliance. If the HTTP callout agent is an application or Web server that serves other client requests, such an expression prevents the appliance from processing those client requests. Instead, use a unique

> custom header as described earlier.

Caching HTTP callout responses

September 14, 2021

For improved performance while using callouts, you can use the integrated caching feature to cache callout responses. The responses are stored in an integrated caching content group named callout-ContentGroup for a specified time duration.

Note: To cache callout responses, make sure that the integrated caching feature is enabled.

To set the cache duration by using the command line interface

At the command prompt, type:

```
set policy httpCallout <name> -cacheForSecs <secs>
```

Example:

```
1 > set httpcallout httpcallout1 -cacheForSecs 120
2 <!--NeedCopy-->
```

To set the cache duration by using the configuration utility

1. Navigate to **AppExpert > HTTP Callouts**.
2. In the details pane, select the HTTP callout for which you want to set the cache duration and click **Open**.
3. In the **Configure HTTP Callout** dialog box, specify the **Cache Expiration Time**.
4. Verify that you have entered the correct time duration, and then click **OK**.

Use Case: Filtering clients by using an IP blacklist

September 14, 2021

HTTP callouts can be used to block requests from clients that are blacklisted by the administrator. The list of clients can be a publicly known blacklist, a blacklist that you maintain for you organization, or a combination of both.

The Citrix ADC appliance checks the IP address of the client against the pre-configured blacklist and blocks the transaction if the IP address has been blacklisted. If the IP address is not in the list, the appliance processes the transaction.

To implement this configuration, you must perform the following tasks:

1. Enable responder on the Citrix ADC appliance.
2. Create an HTTP callout on the Citrix ADC appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response to the HTTP callout, and then bind the policy globally.
4. Create an HTTP callout agent on the remote server.

Enabling responder

You must enable responder before you can use it.

To enable responder by using the GUI

1. Make sure that you have installed the responder license.
2. In the configuration utility, expand AppExpert, and right-click **Responder**, and then click **Enable Responder** feature.

Creating an HTTP callout on the Citrix ADC appliance

Create an HTTP callout, HTTP_Callout, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout pdf.

Configuring a responder policy and binding it globally

After you configure the HTTP callout, verify the callout configuration, and then configure a responder policy to invoke the callout. While you can create a responder policy in the
Policies sub-node and then bind it globally by using the
Responder Policy Manager, this demonstration uses the
Responder Policy Manager to create the responder policy and bind the policy globally.

To create a responder policy and bind it globally by usin

1. Navigate to **AppExpert** > **Responder**.
2. In the details pane, under **Policy Manager**, click **Policy Manager**.
3. In the **Responder Policy Manager** dialog box, click **Override Global**.

4. Click **Insert Policy**, and then, under **Policy Name**, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
   a) In Name, type **PolicyResponder1**.
   b) In **Action**, select **RESET**.
   c) In **Undefined-Result Action**, select **Global undefined-result** action.
   d) In **Expression**, type the following default syntax expression:

```
"HTTP.REQ.HEADER("Request").EQ("Callout Request").NOT && SYS.
   HTTP_CALLOUT(HTTP_Callout).CONTAINS("IP Matched")"
<!--NeedCopy-->
```

   e) Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP callout agent on the remote server

You must now create an HTTP callout agent on the remote callout server that will receive callout requests from the Citrix ADC appliance and respond appropriately. The HTTP callout agent is a script that is different for each deployment and must be written with the server specifications in mind, such as the type of database and the scripting language supported.

Following is a sample callout agent that verifies whether the given IP address is part of an IP blacklist. The agent has been written in the Perl scripting language and uses a MYSQL database.

The following CGI script checks for a given IP address on the callout server.

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
        use DBI();
        use CGI qw(:standard);
#Take the Client IP address from the request query
        my $ip_to_check = param('cip');
# Where a MYSQL database is running
        my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
# Database username to connect with
        my $db_user_name =  'dbuser' ;
# Database password to connect with
        my $db_password = 'dbpassword';
        my ($id, $password);
# Connecting to the database
        my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
        my $sth = $dbh->prepare(qq{
```



```
  select * from bad_clnt  }
 );
        $sth->execute();
        while (my ($ip_in_database) = $sth->fetchrow_array()) {

        chomp($ip_in_database);
# Check for IP match
        if ($ip_in_database eq $ip_to_check) {

              print "\n IP Matched\n";
                                                    $sth->finish();
                                               exit;
                 }

        }

       print "\n IP Failed\n";
       $sth->finish();
       exit;
<!--NeedCopy-->
```

Use Case: ESI support for fetching and updating content dynamically

September 14, 2021

Edge Side Includes (ESI) is a markup language for edge-level dynamic Web content assembly. It helps in accelerating dynamic Web-based applications by defining a simple markup language to describe cacheable and non-cacheable Web page components that can be aggregated, assembled, and delivered at the network edge. By using HTTP callouts on the Citrix ADC appliance, you can read through the ESI constructs and aggregate or assemble content dynamically.

To implement this configuration, you must perform the following tasks:

1. Enable rewrite on the Citrix ADC appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a rewrite action to replace the ESI content with the callout response body.
4. Configure a rewrite policy to specify the conditions under which the action is performed, and then bind the rewrite policy globally.

Enabling rewrite

Rewrite must be enabled before it is used on the Citrix ADC appliance. The following procedure describes the steps to enable the rewrite feature.

To enable rewrite by using the GUI

1. Make sure that you have installed the rewrite license.
2. In the configuration utility, expand AppExpert, and right-click Rewrite, and then click Enable Rewrite feature.

Creating an HTTP Callout on the Citrix ADC Appliance

For more information about creating an HTTP callout, see Configuring an HTTP Callout.
For more information about the parameter values, see Parameters and Values for HTTP-Callout-2 pdf.

Configuring the Rewrite Action

Create a rewrite action, Action-Rewrite-1, to replace the ESI content with the callout response body. Use the parameter settings shown in the following table.

Table 2. Parameters and Values for Action-Rewrite-1

Parameter	Value
Name	Action-Rewrite-1
Type	Replace
Expression to choose target text reference	“HTTP.RES.BODY(500).AFTER_STR (\” <example>\”).BEFORE_STR (\”</example>\”)”
String expression for replacement text	“SYS.HTTP_CALLOUT(HTTP-Callout-2)”

To configure the rewrite action by using the configuration utility

1. Navigate to **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Rewrite Action** dialog box, in Name, type **Action-Rewrite-1**.
4. In Type, select **REPLACE**.
5. In **Expression** to choose target text reference, type the following default syntax expression:


```
"HTTP.RES.BODY(500).AFTER_STR("<example>").BEFORE_STR("<example>")
    "
<!--NeedCopy-->
```

6. In the String expression for replacement text, type the following string expression:

```
"SYS.HTTP_CALLOUT(HTTP-Callout-2)"
<!--NeedCopy-->
```

7. Click **Create**, and then click **Close**.

Creating the Rewrite Policy and Binding it Globally

Create a rewrite policy, Policy-Rewrite-1, with the parameter settings shown in the following table. You can create a rewrite policy in the
Policies subnode and then bind it globally by using the
Rewrite Policy Manager. Alternatively, you can use the
Rewrite Policy Manager to perform both these tasks simultaneously. This demonstration uses the Rewrite Policy Manager to perform both tasks.

Table 3. Parameters and Values for Policy-Rewrite-1

Parameter	Value
Name	Policy-Rewrite-1
Action	Action_Rewrite-1
Undefined Result Action	-Global undefined-result action-
Expression	“HTTP.REQ.HEADER(”Name”).CONTAINS (”Callout”).NOT”

To configure a rewrite policy and bind it globally by using the configuration utility

1. Navigate to **AppExpert** > **Rewrite**.
2. In the details pane, under **Policy Manager**, click **Rewrite Policy Manager**.
3. In the **Rewrite Policy Manager** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Rewrite Policy** dialog box, do the following:
   1.In Name, type Policy-Rewrite-1.
   a) In Action, select Action-Rewrite-1.
   b) In Undefined-Result Action, select Global undefined-result action.

c) In Expression, type the following default syntax expression:

```
"HTTP.REQ.HEADER("Name").CONTAINS("Callout").NOT"
<!--NeedCopy-->
```

a) Click **Create**, and then click **Close**.

6. Click **Apply Changes**, and then click **Close**.

Use Case: Access control and authentication

September 14, 2021

In high security zones, it is mandatory to externally authenticate the user before a resource is accessed by clients. On the Citrix ADC appliance, you can use HTTP callouts to externally authenticate the user by evaluating the credentials supplied. In this example, the assumption is that the client is sending the user name and password through HTTP headers in the request. However, the same information could be fetched from the URL or the HTTP body.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the Citrix ADC appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling Responder

The responder feature must be enabled before it is used on the Citrix ADC appliance.

To enable responder by using the configuration utility

1. Make sure that the responder license is installed.
2. In the configuration utility, expand AppExpert, and right-click Responder, and then click **Enable Responder feature**.

Creating an HTTP callout on the Citrix ADC appliance

Create an HTTP callout, HTTP-Callout-3, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

Table 1. Parameters and Values for HTTP-Callout-3

Parameter	Value	Name
Name	Policy-Responder-3	

Parameter

Value

Name

HTTP-Callout-3

Server to receive callout request:

IP Address

10.103.9.95

Port

80

Request to send to the server:

Method

GET

Host Expression

10.102.3.95

URL Stem Expression

“/cgi-bin/authenticate.pl”

Headers:

Name

Request

Value-expression

Callout Request

Parameters:

Name

Username

Value-expression

HTTP.REQ.HEADER(“Username”).VALUE(0)

Name

Password

Value-expression

HTTP.REQ.HEADER(“Password”).VALUE(0)

Server Response:

Return Type

TEXT

Expression to extract data from the response

HTTP.RES.BODY(100)

Creating a Responder Policy to Analyze the Response

Create a responder policy, Policy-Responder-3, that will check the response from the callout server and RESET the connection if the source IP address has been blacklisted. Create the policy with the parameters settings shown in the following table. While you can create a responder policy in the Policies subnode and then bind it globally by using the
Responder Policy Manager, this demonstration uses the
Responder Policy Manager to create the responder policy and bind the policy globally.

Table 2. Parameters and Values for Policy-Responder-3

Parameter	Value
Name	Policy-Responder-3
Action	RESET
Undefined-Result-Action	-Global undefined-result action-
Expression	“HTTP.REQ.HEADER(\“Request\”).EQ(\“Callout Request\”).NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\“Authentication Failed\”)”

To create a responder policy and bind it globally by using the configuration utility

1. Navigate to **AppExpert** > **Responder**.
2. In the details pane, under **Policy Manager**, click **Responder Policy Manager**.
3. In the **Responder Policy Manger** dialog box, click **Override Global**.

4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
   a) In Name, type Policy-Responder-3.
   b) In Action, select **RESET**.
   c) In Undefined-Result Action , select Global undefined-result action.
   d) In the Expression text box, type:

```
"HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.
   HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\"Authentication Failed
   \")"
<!--NeedCopy-->
```

   a) Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout Agent on the Remote Server

You now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the Citrix ADC appliance and responds appropriately. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

Following is sample callout agent pseudo-code that verifies whether the supplied user name and password are valid. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To verify the supplied user name and password by using pseudo-code

1. Accept the user name and password supplied in the request and format them appropriately.
2. Connect to the database that contains all the valid user names and passwords.
3. Check the supplied credentials against your database.
4. Format the response as required by the HTTP callout.
5. Send the response to the Citrix ADC appliance.

Use Case: OWA-based spam filtering

September 14, 2021

Spam filtering is the ability to dynamically block emails that are not from a known or trusted source or that have inappropriate content. Spam filtering requires an associated business logic that indicates that a particular kind of message is spam. When the Citrix ADC appliance processes Outlook Web Access (OWA) messages based on the HTTP protocol, HTTP callouts can be used to filter spam.

You can use HTTP callouts to extract any portion of the incoming message and check with an external callout server that has been configured with rules that are meant for determining whether a message is legitimate or spam. In case of spam email, for security reasons, the Citrix ADC appliance does not notify the sender that the email is marked as spam.

The following example conducts a very basic check for various listed keywords in the email subject. These checks can be more complex in a production environment.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the Citrix ADC appliance.
2. Create an HTTP callout on the Citrix ADC appliance and configure it with details about the external server and other required parameters.
3. Create a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling responder

The responder feature must be enabled before it can be used on the Citrix ADC appliance.

To enable responder by using the GUI

1. Make sure that the responder license is installed.
2. In the configuration utility, expand AppExpert, and right-click **Responder**, and then click **Enable Responder** feature.

Creating an HTTP callout on the Citrix ADC appliance

Create an HTTP callout, HTTP-Callout-4, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

For more information, see Parameters and Values for HTTP-Callout-4 pdf.

Creating a responder action

Create a responder action, Action-Responder-4. Create the action with the parameter settings shown in the following table.

Parameter	Value
Name	Action-Responder-4
Type	Respond with
Target	””HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n””

Table 2. Parameters and Values for Action-Responder-4

To create a responder action by using the configuration utility

1. Navigate to **AppExpert** > **Responder** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, in Name, type **Action-Responder-4**.
4. In Type, click **Respond with**.
5. In Target, type:

```
"\"HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By:
   ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\
   nCache-Control: no-cache\r\n\r\n\""
<!--NeedCopy-->
```

6. Click **Create**, and then click **Close**.

Creating a Responder Policy to Invoke the HTTP Callout

Create a responder policy, Policy-Responder-4, that will check the request body and, if the body contains the word “
subject,” invoke the HTTP callout to verify the email. Create the policy with the parameter settings shown in the following table. While you can create a responder policy in the
Policies subnode and then bind it globally by using the
Responder Policy Manager, this demonstration uses the
Responder Policy Manager to create the responder policy and bind it globally.

Parameter	Value
Name	Policy-Responder-4
Action	Action-Responder-4
Undefined-Result-Action	-Global undefined-result action-
Expression	“HTTP.REQ.BODY(1000).CONTAINS(”urn:schemas:httpmail:su && SYS.HTTP_CALLOUT(HTTP-Callout-4)”

To create a responder policy by using the configuration utility

1. Navigate to **AppExpert** > **Responder**.
2. In the details pane, under **Policy Manager**, click **Responder policy** manager.
3. In the **Responder Policy Manger** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
   a) In Name, type **Policy-Responder-4**.
   b) In Action, click **Action-Responder-4**.
   c) In Undefined-Result Action, click **Global undefined-result** action.
   d) In the **Expression** text box, type:

```
"HTTP.REQ.BODY(1000).CONTAINS(\"urn:schemas:httpmail:subject
   \") && SYS.HTTP_CALLOUT(HTTP-Callout-4)"
<!--NeedCopy-->
```

   e) Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP callout agent on the remote server

You will now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the Citrix ADC appliance and responds accordingly. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

The following pseudo-code provides instructions for creating a callout agent that checks a list of words that are generally understood to indicate spam mails. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To identify spam email by using pseudo-code

1. Accept the email subjecta provided by the Citrix ADC appliance.
2. Connect to the database that contains all the terms against which the email subject is checked.
3. Check the words in the email subject against the spam word list.
4. Format the response as required by the HTTP callout.
5. Send the response to the Citrix ADC appliance.

Use Case: Dynamiccontent switching

September 14, 2021

This use case provides dynamic content switching by using an HTTP callout to get the name of the load balancing virtual server to which the request is forwarded.

1. Add a content switching virtual server.

```
add cs vserver cs_vserver1 HTTP 10.102.29.196 80
<!--NeedCopy-->
```

2. Create an HTTP callout.

```
add policy httpCallout http_callout1
<!--NeedCopy-->
```

3. Configure the HTTP callout to respond with the name of the load balancing virtual server from a request that contains the client IP address in the HTTP header “X-CLIENT-IP”.

```
> set policy httpCallout http_callout1 -IPAddress 10.217.14.23 -
    port 80 -returnType TEXT -hostExpr ""www.get-lbvip.com"" -
    urlStemExpr "\"/index.html\"" -headers X-CLIENT-IP(CLIENT.IP.
    SRC) -resultExpr "HTTP.RES.BODY(1000).AFTER_STR(\"<lbvip>\").
    BEFORE_STR("<lbvip")"
<!--NeedCopy-->
```

4. Configure the content switching action to retrieve the callout response.

```
add cs action cs_action1 -targetVserverExpr 'SYS.HTTP_CALLOUT(
    http_callout1)'
<!--NeedCopy-->
```


> Note:
>
> You must bind a load balancing virtual server to the content switching virtual server to account for:

- The non-availability of the load balancing virtual server that the callout resolves to.
- A UNDEF condition that results from the execution of the callout.

```
> bind cs vserver cs_vserver1 -lbvserver default_lbvip
<!--NeedCopy-->
```

5. Configure the content switching policy.

```
add cs policy cs_policy1 -rule true -action cs_action1
<!--NeedCopy-->
```

6. Binding the content switching policy to the content switching virtual server.

```
bind cs vserver cs_vserver1 -policyName cs_policy1 -priority 10
<!--NeedCopy-->
```

Pattern sets and data sets

September 14, 2021

Policy expressions for string matching operations on a large set of string patterns tend to become long and complex. Resources consumed by the evaluation of such complex expressions are significant in terms of processing cycles, memory, and configuration size. You can create simpler, less resource-intensive expressions by using pattern matching.

Depending on the type of patterns that you want to match, you can use one of the following features to implement pattern matching:

- A pattern set is an array of indexed patterns used for string matching during default syntax policy evaluation. Example of a pattern set: imagetypes {svg, bmp, png, gif, tiff, jpg}.
- A data set is a specialized form of pattern set. It is an array of patterns of types number (integer), IPv4 address, or IPv6 address.

In many cases, you can use either pattern sets or data sets. However, in cases where you want specific matches for numerical data or IPv4 and IPv6 addresses, you must use data sets.

> **Note**:

> Pattern sets and data sets can be used only in default syntax policies.

To use pattern sets or data sets, first create the pattern set or data set and bind patterns to it. Then, when you configure a policy for comparing a string in a packet, use an appropriate operator and pass the name of the pattern set or data set as an argument.

How string matching works with pattern sets and data sets

September 14, 2021

A pattern set or data set contains a set of patterns, and each pattern is assigned a unique index. When a policy is applied to a packet, an expression identifies a string to be evaluated, and the operator compares the string to the patterns defined in the pattern set or data set until a match is found or all patterns have been compared. Then, depending on its function, the operator returns either a boolean value that indicates whether or not a matching pattern was found or the index of the pattern that matches the string.

Note: This topic explains the working of a pattern set. Data sets work the same way. The only difference between pattern sets and data sets is the type of patterns defined in the set.

Consider the following use case to understand how patterns can be used for string matching.

You want to determine whether the URL suffix (target text) contains any of the image file extensions. Without using pattern sets, you would have to define a complex expression, as follows:

```
1 HTTP.REQ.URL.SUFFIX.CONTAINS("svg") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
     bmp") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("png") ¦¦
2 HTTP.REQ.URL.SUFFIX.CONTAINS("gif") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
     tiff") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("jpg")
3 <!--NeedCopy-->
```

If the URL has a suffix of “jpg,” with the above compound expression, the Citrix ADC appliance has to iterate through the entire compound expression sequentially, from one sub expression to the next, to determine that the request refers to a jpg image. The following figure shows the steps in the process.



When a compound expression includes hundreds of sub expressions, the above process is resource intensive. A better alternative is an expression that invokes a pattern set, as shown in the following figure.



During policy evaluation as shown above, the operator (CONTAINS_ANY) compares the string identified in the request with the patterns defined in the pattern set until a match is found. With the Sample_Patset expression, the multiple iterations through six sub expressions are reduced to just one.

By eliminating the need to configure compound expressions that perform string matching with multiple OR operations, pattern sets or data sets simplify configuration and accelerate processing of requests and responses.

Configuring a Pattern Set

September 14, 2021

To configure a pattern set, you must specify the strings that are to serve as patterns. You can manually assign a unique index value to each of these patterns, or you can allow the index values to be assigned automatically.

Note: Pattern sets are case sensitive (unless you specify the expression to ignore case). Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."

Points to remember about index values:

- You cannot bind the same index value to more than one pattern.
- An automatically assigned index value is one number larger than the highest index value of the existing patterns within the pattern set. For example, if the highest index value of existing patterns in a pattern set is 104, the next automatically assigned index value is 105.
- If you do not specify an index for the first pattern, index value 1 is automatically assigned to that pattern.
- Index values are not regenerated automatically if one or more patterns are deleted or modified. For example, if the set contains five patterns, with indexes from 1 through 5, and if the pattern with an index of 3 is deleted, the other index values in the pattern set are not automatically regenerated to produce values from 1 through 4.
- The maximum index value that can be assigned to a pattern is 4294967290. If that value is already assigned to a pattern in the set, you must manually assign index values to any newly added patterns. An unused index value that is lower than a currently used value cannot be assigned automatically.

To configure a pattern set by using the command line interface

At the command prompt, do the following:

1. Create a pattern set.

   ```
   add policy patset <name>
   ```

Example:

```
add policy patset samplepatset
```

1. Bind patterns to the pattern set.

   ```
   bind policy patset <name> <string> [-index <positive_integer>][-charset
    ( ASCII | UTF_8 )] [-comment <string>]
   ```

Example:


```
bind policy patset samplepatset product1 -index 1 -comment short description
 about the pattern bound to the pattern set
```

Note: Repeat this step for all the patterns you want to bind to the pattern set.

1. Verify the configuration.

```
show policy patset <name>
```

To configure a pattern set by using the configuration utility

1. Navigate to **AppExpert** > **Pattern Sets**.
2. In the details pane, click **Add** to open the **Create Pattern Set** dialog box.
3. Specify a name for the pattern set in the Name text box.
4. Under Specify Pattern, type the first pattern and, optionally, specify values for the following parameters:
   - Treat back slash as escape character—Select this check box to specify that any backslash characters that you might include in the pattern are to be treated as escape characters.
   - Index—A user assigned index value, from 1 through 4294967290.
5. Verify that you have entered the correct characters, and then click **Add**.
6. Repeat steps 4 and 5 to add more patterns, and then click **Create**.

Configure file-based pattern sets

The Citrix ADC appliance supports file-based pattern sets.

To configure file-based pattern sets by using CLI

At the command prompt, type the following commands:

- Import a new pattern set file into the Citrix ADC appliance.

```
import policy patsetfile <src> <name> -delimiter <char> -charset
   <ASCII | UTF_8>
<!--NeedCopy-->
```

Example:

```
import policy patsetfile local:test.csv clientids_list -
   delimiter ,
<!--NeedCopy-->
```

You can import a file from a local device, HTTP server, or FTP server. To add the file from your local device, the file must be available in `/var/tmp` location.

- Update an existing pattern set file on the Citrix ADC appliance.

```
update policy -patsetfile <patset filename>
<!--NeedCopy-->
```

Example:

```
update policy -patsetfile clientids_list
<!--NeedCopy-->
```

- Add a pattern set file to the packet engine.

```
add policy -patsetfile <patset filename>
<!--NeedCopy-->
```

Example:

```
add policy -patsetfile clientids_list
<!--NeedCopy-->
```

- Bind patterns to the pattern set.

```
add policy patset <patset name> -patsetfile <patset filename>
<!--NeedCopy-->
```

Example:

```
add policy patset clientid_patset -patsetfile clientids_list
<!--NeedCopy-->
```

- Verify the configuration.

```
show policy patsetfile clientids_list

Name: clientids_list
Patset Name: clientid_patset
Number of Imported Patterns: 8
Number of Bound Patterns: 8
(All the patterns bound successfully)

Done
<!--NeedCopy-->
```

To configure file-based pattern sets by using GUI

1. Navigate to **AppExpert-> Pattern Set Files**.

2. In the **Imported** pane, click **Import**.
3. In the **Configure Policy Patset File** page, select the file you want to import, and click **OK**.
4. Select the imported file, and click **Add**.
5. In the **Create Policy Patset File** page, enter the details, and click **Create** to add a policy pattern set.

Configuring a data set

September 14, 2021

To configure a data set, you must specify the strings that server as a pattern, assign a type (number, IPv4 address, or IPv6 address) and configure the dataset range. You can manually assign a unique index value to the pattern, or you can allow the index values to be assigned automatically. Dataset is not related to HTTP or any 7 layer protocol. It works only on text or string. There are different types of dataset such as NUM, ULONG, IPv4, IPv6, MAC, DOUBLE. You can select a type and define the dataset range based on the specified type.

> **Note:**
>
> Policy data sets are case sensitive (unless you specify the expression to ignore case). Therefore, the MAC address ff:ff:ff:ff:ff:ff for example, is not the same as the MAC address FF:FF:FF:FF:FF:FF.

The rules applied for index values of data sets are similar to pattern sets. For information about index values, see Configuring a Pattern Set.

To configure a data set

You must complete the following steps to configure a data set:

1. Add a policy dataset
2. Bind pattern to a policy dataset
3. Add a policy expression
4. Verify the policy configuration

Add a policy dataset

At the command prompt, do the following:

```
add policy dataset <name> <type>
```

Example:

```
add policy dataset ds1 ipv4 -comment numbers
```


Bind a pattern to the data set

At the command prompt, type:

```
bind policy dataset <name> <value> [-index <positive_integer>] [-endRange <
string>] [-comment <string>]
```

Example:

```
bind policy dataset ds1 1.1.1.1 -endRange 1.1.1.10 -comment short description
 about the pattern bound to the data set
```

> **Note:**
>
> You must repeat this step for all the patterns you want to bind to the data set. You can bind only up to 5000 patterns to a dataset.
>
> And, a dataset range must not overlap with other ranges bound to a dataset and cannot include single values bound to the dataset. If you bind a dataset with an overlapping range results in an error.

Example:

```
1 add policy dataset ip_set ipv4
2  Done
3 bind policy dataset ip_set 2.2.2.25
4  Done
5 bind policy dataset ip_set 2.2.2.20 -endRange 2.2.2.30
6 ERROR: The range overlaps an existing range or includes a value bound
     to the dataset.
7 <!--NeedCopy-->
```

A value is considered to be in the dataset if it is either equal to a single value bound to the data set or is between the lower-value and upper-value (lower-value <= value && value <- upper-value), for a range bound to the data set.

Use policy expression in a policy data set

At the command prompt, type:

```
add policy expression exp1 http.req.body(100).contains_any("ds1")
```

Where,
The expression checks whether there is any pattern (or pattern within the range) bound to the dataset ds1 is present in the first 100 bytes of the HTTP request body.

Verify dataset configuration

At the command prompt, type:

```
show policy dataset ds1
> show policy dataset ds1
```

Example:

```
1         Dataset:        ds1
2          Type:   IPV4
3 1)       Bound Dataset Range from: 1.1.1.1       through: 1.1.1.10
            Index:  1
4 <!--NeedCopy-->
```

To configure a data set by using the configuration utility

Follow the steps given below to configure a policy dataset:

1. Navigate to **AppExpert > Data Sets**.
2. In the details pane, under Data Sets, click **Add**.
3. In the **Configure Data Set** page, set the following parameters.
   a) Name. Name of the policy data set.
   b) Type. Type of value to bind to the dataset.

   Configuring data set
4. Click **Insert** to bind the dataset value of specific type.
   a) Value. Value of the specified type associated with the dataset.
   b) Index. The index value of the dataset.
   c) End range. The dataset entry. This is a range <value> to <end_range>.
   d) Comments. A short description about the data set.

   dataset binding
5. Click **Insert** and **Close**.
6. Enter comments.
7. Click **Create** and **Close**.

Using pattern sets and data sets

September 14, 2021

Default syntax policy expressions that take pattern sets or data sets as an argument can be used to perform string matching operations.

The usage is as follows:

```
<text>.<operator>("<name>")
<!--NeedCopy-->
```

where,

- <text> is the expression that identifies a string in a packet. Example: HTTP.REQ.HEADER(“Host”).
- <operator> is one of the operators described in the Pattern Set Types table pdf.

For sample usage, see Sample Usage.

Sample usage

September 14, 2021

To understand the usage of pattern sets in expressions, consider the example of a pattern set named “imagetypes.”

Patterns	Index value
svg	1
bmp	2
png	3
gif	4
tiff	5
jpg	6

Table 1. Pattern set “imagetypes”

Example 1: Determine whether the suffix of an HTTP request is one of the file extensions defined in the “imagetypes” pattern set.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_ANY(“imagetypes”)
- **Sample URL.** http://www.example.com/homepageicon.jpg
- **Result.** TRUE

Example 2: Determine whether the suffix of an HTTP request is one of the file extensions defined in

the “imagetypes” pattern set, and return the index of that pattern.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”)
- **Sample URL.** `http://www.example.com/mylogo.png`
- **Result.** 4 (The index value of the pattern “gif”.)

Example 3: Use the index value of a pattern to determine whether the URL suffix is within a specified index-value range.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(3) && HTTP.REQ.URL.SUFFIX.EQUALS
- **Sample URL.** `http://www.example.com/mylogo.png`
- **Result.** TRUE (The index value of gif file types is 4.)

Example 4: Implement one set of policies for file extensions bmp, jpg, and png, and a different set of policies for gif, tiff, and svg files.

An expression that returns the index of a matched pattern can be used to define traffic subsets for a web application. The following two expressions could be used in content switching policies for a content switching virtual server:

- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).LE(3)
- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(4)

Variables

September 14, 2021

Variables are named objects that store information in the form of tokens. These tokens are used within and across different transactions on the Citrix ADC Appliance for internal computation and policy processing.

The Citrix ADC appliance supports creation of variables of the following types:

- **Singleton variables.** Can have a single value of one of the following types: ulong and text (max-size). The ulong type is an unsigned 64-bit integer, the text type is a sequence of bytes, and max-size is the maximum number of bytes in the sequence.
- **Map variables.** Maps hold values associated with keys: each key-value pair is called a map entry. The key for each entry is unique within the map. Maps are specified as follows:

  map (key_type, value_type, max-values).

  where,

  - *key_type* is the data type of the key. It is of type text (max-size).

- *value_type* is the data type of the values of the map. It can be of type ulong or text (max-size).
- *max-values* is the maximum number of entries that the map can contain. It is of type ulong.

Values for these variables are set using assignments which must be invoked on policy actions.

> **Note**: Variables are not yet supported in a high-availability setup or in a cluster.

Variables Scope

A map variable or a singleton variable can have a global scope. Alternatively, the scope of a singleton variable can be limited to a single transaction.

- **Global Scope Variable** - A variable with global scope (the default) has only one instance, and that instance has the same value(s) across all cores of a Citrix ADC appliance and across all nodes of a cluster or HA configuration. Global variable values exist until they are explicitly deleted, until they expire, or until a standalone appliance is restarted or all nodes of a cluster or HA configuration are restarted.
- **Transaction Scope Variable** - A variable with transaction scope has a separate instance, with its own value, for each transaction processed by the Citrix ADC appliance. When the transaction processing is complete, the transaction variable value is deleted.

> **Note**: Transaction scope variables are available in Citrix ADC release 10.5.e or later.

Configuring and Using Variables

September 14, 2021

You must first create a variable and then assign a value or specify the operation that must be performed on the variable. After performing these operations, you can use the assignment as a policy action.

Note: Once configured, a variable's settings cannot be modified or reset. If the variable needs to be changed, the variable and all references to the variable (expressions and assignments) must be deleted. The variable can then be re-added with new settings, and the references (expressions and assignments) can be re-added.

To configure variables by using the command line interface

1. Create a variable.


```
add ns variable <name> -type <string> [-scope global] [-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )] [-ifNoValue ( undef | init )] [-init <string>] [-expires <positive_integer>] [-comment <string>]
<!--NeedCopy-->
```

Note: Refer to the man page “man add ns variable” for description of the command parameters.

Example 1: Create a ulong variable named “my_counter” and initialize it to 1.

```
add ns variable my_counter  - type ulong -init 1
<!--NeedCopy-->
```

Example 2: Create a map named “user_privilege_map”. The map will contain keys of maximum length 15 characters and text values of maximum length 10 characters, with a maximum of 10000 entries.

```
add ns variable user_privilege_map -type map(text(15),text(10),10000)
<!--NeedCopy-->
```

Note: If the map contains 10000 unexpired entries, assignments for new keys reuse one of the least recently used entries. By default, an expression trying to get a value for a non-existent key will initialize an empty text value.

Assign the value or specify the operation to be performed on the variable. This is done by creating an assignment.

```
add ns assignment <name> -variable <expression> [-set <expression> | -add <expression> | -sub <expression> | -append <expression> | -clear ] [-comment <string>]
<!--NeedCopy-->
```

Note: A variable is referenced by using the variable selector ($). Therefore,
$variable1 is used to refer to text or ulong variables. Similarly,
$variable2[key-expression] is used to refer to map variables.

Example 1: Define an assignment named “inc_my_counter” that automatically adds 1 to the “my_counter” variable.

```
add ns assignment inc_my_counter -variable $my_counter -add 1
<!--NeedCopy-->
```

Example 2: Define an assignment named “set_user_privilege” that adds to the “user_privilege_map” variable an entry for the client’s IP address with the value returned by the “get_user_privilege” HTTP callout.


```
add ns assignment set_user_privilege -variable $user_privilege_map[
    client.ip.src.typecast_text_t] -set sys.http.callout(
    get_user_privilege)
<!--NeedCopy-->
```

Note: If an entry for that key already exists, the value will be replaced. Otherwise a new entry for the key and value will be added. Based on the previous declaration for user_privilege_map, if the map already has 10000 entries, one of the least recently used entries will be reused for the new key and value.

1. Invoke the variable assignment in a policy.

   There are two functions that can operate on map variables.

   - **$name.valueExists(key-expression).** Returns true if there is a value in the map selected by the key-expression. Otherwise returns false. This function will update the expiration and LRU information if the map entry exists, but will not create a new map entry if the value does not exist.
   - **$name.valueCount.** Returns the number of values currently held by the variable. This is the number of entries in a map. For a singleton variable, this is 0 if the variable is uninitialized or 1 otherwise.

   Example: Invoke the assignment named “set_user_privilege” with a compression policy.

```
add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src.typecast_text_t).not -resAction
    set_user_privilege
<!--NeedCopy-->
```

Use Case to Insert HTTP header in the Response Side

The following example shows an example of a singleton variable.

Add a singleton variable of type text. This variable can hold maximum 100 Bytes data.

```
add ns variable http_req_data -type text(100) -scope transaction
<!--NeedCopy-->
```

Add an assignment action, which will be used to store the HTTP request data into the variable.

```
add ns assignment set_http_req_data -variable $http_req_data -set http.
    req.body(100)
<!--NeedCopy-->
```


Add a rewrite action to insert HTTP header, whose value will be fetched from the variable.

```
add rewrite action act_ins_header insert_http_header user_name
   $http_req_data.after_str("user_name").before_str("password")
<!--NeedCopy-->
```

Add a rewrite policy which will evaluate in the request time, and take assignment action to store data. When we hit this policy, we'll take assignment action and store the data into the ns variable (http_req_data)

```
add rewrite policy pol_set_variable true set_http_req_data

bind rewrite global pol_set_variable 10 -type req_dEFAULT
<!--NeedCopy-->
```

Add a rewrite policy which will evaluate in the response time, and add an HTTP header in the response.

```
add rewrite policy pol_ins_header true act_ins_header

bind rewrite global pol_ins_header 10 -type res_dEFAULT
<!--NeedCopy-->
```

Assignment action

In a Citrix ADC appliance, an assignment action bound to the policy is triggered when the policy rule evaluates to true. The action updates the value in the variable which can be used in subsequent policy rule evaluations. This way, the same variable can be updated and used for subsequent policy evaluations in the same feature. Previously, the appliance executed assignment actions only after evaluating all of the policies in the feature when the policies of the associated assignment actions evaluated to true. Therefore, the variable value set by the assignment action cannot be used in the subsequent policy rule evaluations within the feature.

This functionality can be understood better with a use case that controls access list for clients on a Citrix ADC appliance. The access decision is provided by a separate web service, with the request `GET /client-access?<client-IP-address>` which returns a response with "BLOCK" or "ALLOW" in the body. The HTTP callout is configured to include the IP address of the client that is associated with an incoming request. When the Citrix ADC appliance receives a request from a client, the appliance generates the callout request and sends it to the callout server, which hosts a database of blacklisted IP addresses and an HTTP callout agent that checks whether the client's IP address is listed in the database. The HTTP callout agent receives the callout request, checks whether the client's IP address is listed, and sends a response. The response is a status code, 200, 302 along with "BLOCK" or "ALLOW" in the body. Based on the status code, the appliance performs the policy evaluation. If the policy

evaluation is true, the assignment action is triggered immediately and action sets the value to the variable. The appliance uses and sets this variable value for subsequent policy evaluation in the same module.

Use case for configuring assignment action

Follow the steps below to configure assignment action and use variable for subsequent policies:

1. The access decision is provided by a separate web service, with the request which returns a response with BLOCK or ALLOW in the body.

   ```
   GET /url-service>/url-allowed?<URL path>
   ```

2. Set up a map variable to hold the access decisions for URLs.

   ```
   add ns variable url_list_map -type 'map(text(1000),text(10),10000)'
   ```

3. Set up an HTTP callout to send the access request to the web service.

   ```
   add policy httpCallout url_list_callout -vserver url_vs -returnType
   TEXT -urlStemExpr '"/url-allowed?" + HTTP.REQ.URL.PATH'-resultExpr '
   HTTP.RES.BODY(10)'
   ```

4. Set up an assignment action to invoke the callout to get the access decision and assign it to the map entry for the URL.

   ```
   add ns assignment client_access_assn -variable '$client_access_map[
   CLIENT.IP.SRC.TYPECAST_TEXT_T]'-set SYS.HTTP_CALLOUT(client_access_callout
   )
   ```

5. Set up a responder action to send a 403 response if a URL request is blocked.

   ```
   add responder action url_list_block_act respondwith '"HTTP/1.1 403
   Forbidden\\r\\n\\r\\n"'
   ```

6. Set up a responder policy to set the map entry for the URL if it is not already set. With the immediate action enhancement, the map entry value is set when this policy is evaluated. Prior to the enhancement, the assignment was not done until all responder policies had been evaluated decision is provided by a separate web service.

   ```
   add responder policy url_list_assn_pol '!$url_list_map.VALUEEXISTS(HTTP
   .REQ.URL.PATH)'url_list_assn
   ```

7. Set up a responder policy to block access to a URL if its map entry value is BLOCK. With the immediate action enhancement, the map entry set by the preceding policy is available for use in this policy. Prior to the enhancement, the map entry would still be unset at this point.

   ```
   add responder policy client_access_block_pol '$client_access_map[CLIENT
   .IP.SRC.TYPECAST_TEXT_T] == "BLOCK"'client_access_block_act
   ```


8. Bind the responder policies to the virtual server. **Note**: We cannot globally bind the policies because we don't want to execute them for the HTTP callout on a separate virtual server.

```
bind lb vserver vs -policyName client_access_assn_pol -priority 10 -
gotoPriorityExpression NEXT -type REQUEST
bind lb vserver vs -policyName client_access_block_pol -priority 20 -
gotoPriorityExpression END -type REQUEST
```

To configure variables by using the configuration utility

1. Navigate to **AppExpert > NS Variables**, to create a variable.
2. Navigate to **AppExpert > NS Assignments**, to assign value(s) to the variable.
3. Navigate to the appropriate feature area where you want to configure the assignment as an action.

Use Case: Caching User Privileges

September 14, 2021

In this use case, user privileges ("GOLD", "SILVER", and so on) must be retrieved from an external web service.

To achieve this use case, perform the following operations

Create an HTTP callout to fetch the user privileges from the external web service.

```
1 add policy httpcallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <
     port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
     GET | POST )] [-hostExpr <string>] [-urlStemExpr <string>] [-headers
      <name(value)> ...] [-parameters <name(value)> ...] [-bodyExpr <
     string>][-fullReqExpr <string>] [-scheme ( http | https )] [-
     resultExpr <string>] [-cacheForSecs <secs>] [-comment <string>]
2
3 add policy httpcallout get_user_privilege -ipaddress 10.217.193.84 -
     port 80 -returnType text -httpMethod GET -hostExpr '"/
     get_user_privilege"' -resultExpr 'http.res.body(5)'
4 <!--NeedCopy-->
```

Store the privileges in a variable.


```
add ns variable <name> -type <string> [-scope ( global | transaction )
    ][-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )][-
    ifNoValue ( undef | init )] [-init <string>] [-expires <
    positive_integer>] [-comment <string>]

add ns variable user_privilege_map -type map(text(15),text(10),10000) -
    expires 1200

add ns assignment set_user_privilege -variable $user_privilege_map[
    client.ip.src] -set sys.http_callout(get_user_privilege)
<!--NeedCopy-->
```

Create a policy to check if there is already a cached entry for the client's IP address; if not, it calls the HTTP callout to set a map entry for the client.

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src).not -resAction set_user_privilege>
<!--NeedCopy-->
```

Create a policy that compresses if the cached privilege entry for the client is "GOLD".

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy compress_if_gold_privilege_pol -rule '
    $user_privilege_map[client.ip.src].eq("GOLD")' -resAction compress
<!--NeedCopy-->
```

Bind the compression policies globally.

```
bind cmp global <policyName> [-priority <positive_integer>] [-state (
    ENABLED | DISABLED )] [-gotoPriorityExpression <expression>] [-type
    <type>] [-invoke (<labelType> <labelName>) ]

bind cmp global set_user_privilege_pol -priority 10 NEXT

bind cmp global compress_if_gold_privilege_pol -priority 20 END
<!--NeedCopy-->
```

Use Case: Limiting the Number of Sessions

September 14, 2021

In this use case, the requirement is to limit the number of active backend sessions. In the deployment, each session login has login in the URL and each session logout has logout in the URL. On successful login, the backend sets a sessionid cookie with a unique 10 character value.

To achieve this use case, perform the following operations:

1. Create a map variable that can store each active session. The key of the map is the sessionid. The expiry time for the variable is set to 600 seconds (10 minutes).

```
1 > add ns variable session_map -type map(text(10),ulong,100) -
     expires 600
2 <!--NeedCopy-->
```

2. Create the following assignments for the map variable:

   - Create an entry for the sessionid and set that value to 1 (this value is not actually used).

```
1   > add ns assignment add_session -variable '$session_map[http.
       req.cookie.value("sessionid")]' -set 1
2   <!--NeedCopy-->
```

   - Deallocate the entry for a session ID, which implicitly decrements the value count for session_map.

```
1   > add ns assignment delete_session -variable '$session_map[
       http.req.cookie.value("sessionid")]' -clear
2   <!--NeedCopy-->
```

3. Create responder policies for the following:

   - To check if a map entry exists for that sessionid in the HTTP request. The add_session assignment is executed if the map entry does not exist.

```
1   > add responder policy add_session_pol 'http.req.url.contains
       ("twbkwbis.P_SabanciLogin") || $session_map.valueExists(
       http.req.cookie.value("netsuis"))' add_session
2   <!--NeedCopy-->
```

     Note: The
     valueExists() function in the
     add_session_pol policy counts as a reference to the session's map entry, so each request resets the expiration timeout for its session. If no requests for a session are received after 10 minutes, the session's entry will be deallocated.

   - To check when the session is logged out. The delete_session assignment is executed.


```
add responder policy delete_session_pol "http.req.url.
   contains(\"Logout\")" delete_session
<!--NeedCopy-->
```

- To check for login requests and if the number of active sessions exceed 100. If these conditions are satisfied, in order to limit the number of sessions, the user is redirected to a page that indicates that the server is busy.

```
add responder action redirect_too_busy redirect "/too_busy.
   html"
add responder policy check_login_pol "http.req.url.contains
   (\"twbkwbis.P_SabanciLogin\") && $session_map.valueCount >
    100" redirect_too_busy
<!--NeedCopy-->
```

4. Bind the responder policies globally.

```
bind responder global add_session_pol 30 next
bind responder global delete_session_pol 10
bind responder global check_login_pol 20
<!--NeedCopy-->
```

Policies and expressions

September 14, 2021

The following topics provide the conceptual and reference information that you require for configuring advanced policies on the Citrix® Citrix ADC® appliance.

To know about all the advanced policy expressions supported on the Citrix ADC appliance, see Policy Expressions

Introduction to Policies and Expressions	Describes the purpose of expressions, policies, and actions, and how different Citrix ADC applications make use of them.
Configuring Advanced Policies	Describes the structure of advanced policies and how to configure them individually and as policy banks.
Configuring Advanced Expressions: Getting Started	Describes expression syntax and semantics, and briefly introduces how to configure expressions and policies.

Advanced Expressions: Evaluating Text	Describes expressions that you configure when you want to operate on text (for example, the body of an HTTP POST request or the contents of a user certificate).
Advanced Expressions: Working with Dates, Times, and Numbers	Describes expressions that you configure when you want to operate on any type of numeric data (for example, the length of a URL, a client's IP address, or the date and time that an HTTP request was sent).

Advanced Expressions: Parsing HTTP, TCP, and UDP Data	Describes expressions for parsing IP and IPv6 addresses, MAC addresses, and data that is specific to HTTP and TCP traffic.				
Advanced Expressions: Parsing SSL Certificates	Describes how to configure expressions for SSL traffic and client certificates, for example, how to retrieve the expiration date of a certificate or the certificate issuer.				
Advanced Expressions: IP and MAC Addresses, Throughput, VLAN IDs	Describes expressions that you can use to work with any other client- or server-related data not discussed in other chapters.	Typecasting Data	Describes expressions for transforming data of one type to another.	Regular Expressions	Describes how to pass regular expressions as arguments to operators in advanced expressions.

Configuring Classic Policies and Expressions	Provides details on how to configure the simpler policies and expressions known as classic policies and classic expressions.	Expressions Reference	A reference for classic and advanced expression arguments.	Summary Examples of Advanced Expressions and Policies	Examples of classic and advanced expressions and policies, in both quick reference and tutorial format, that you can customize for your own use.
Tutorial Examples of Advanced Policies for Rewrite	Examples of advanced policies for use in the Rewrite feature.				
Tutorial Examples of Classic Policies	Examples of classic policies for Citrix ADC features such as application firewall and SSL.				

Migration of Apache mod_rewrite Rules to Advanced Policies	Examples of functions that were written using the Apache HTTP Server mod_rewrite engine, with examples of these functions after translation into Rewrite and Responder policies on the Citrix ADC.

Introduction to policies and expressions

September 14, 2021

For many Citrix ADC features, policies control how a feature evaluates data. A policy uses a logical expression, called as rule, to evaluate data, and applies one or more actions based on evaluation. Alternatively, a policy can apply a profile, which defines a complex action.

Some Citrix ADC features use default syntax policies, which provide greater capabilities than older classic policies. If you have migrated to a newer release of the Citrix ADC software and have configured classic policies for features that use default syntax policies, you have to manually migrate policies to advanced policy infrastructure.

Classic and advanced policies

September 16, 2021

> **Warning:**
>
> Classic policy expressions are deprecated from Citrix ADC 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies

Classic policies evaluate basic characteristics of traffic and other data. For example, classic policies can identify whether an HTTP request or response contains a particular type of header or URL.

Advanced policies can perform the same type of evaluations as classic policies. In addition, advanced policy infrastructure (PI) enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header).

In addition to assigning a policy an action or profile, you bind the policy to a particular point in the processing associated with the Citrix ADC features. The bind point is one factor that determines when the policy will be evaluated.

Benefits of using advanced policies

Default syntax policies use a powerful expression language that is built on a class-object model, and they offer several options that enhance your ability to configure the behavior of various Citrix ADC features. With advanced policy infrastructure (PI), you can do the following:

- Perform fine-grained analyses of network traffic from layers 2 through 7.
- Evaluate any part of the header or body of an HTTP or HTTPS request or response.
- Bind policies to the multiple bind points that the advanced policy infrastructure (PI) supports at the default, override, and virtual server levels.
- Use goto expressions to transfer control to other policies and bind points, as determined by the result of expression evaluation.
- Use special tools such as pattern sets, policy labels, rate limit identifiers, and HTTP callouts, which enable you to configure policies effectively for complex use cases.

Also, the configuration utility extends robust graphical user interface support for advanced policy infrastructure (PI) and expressions and enables users who have limited knowledge of networking protocols to configure policies quickly and easily. The configuration utility also includes a policy evaluation feature for advanced policies You can use this feature to evaluate a advanced policy and test its behavior before you commit it, thus reducing the risk of configuration errors.

Basic components of an advanced policy

Following are a few characteristics of an Advanced policy:

- Name. Each policy has a unique name.
- Rule. The rule is a logical expression that enables the Citrix ADC feature to evaluate a piece of traffic or another object. For example, a rule can enable the Citrix ADC to determine whether an HTTP request originated from a particular IP address, or whether a Cache-Control header in an HTTP request has the value “No-Cache.”

Advanced policies can use all of the expressions that are available in a classic policy, with the exception of classic expressions for the SSL VPN client. In addition, Advanced policies enable you to configure more complex expressions.

- Bindings. To ensure that the Citrix ADC can invoke a policy when it is needed, you associate the policy, or bind it, to one or more bind points.

You can bind a policy globally or to a virtual server. For more information, see About policy bindings.

- An associated action. An action is a separate entity from a policy. Policy evaluation ultimately results in the Citrix ADC performing an action.

For example, a policy in the integrated cache can identify HTTP requests for .png or .jpeg files. An action that you associate with this policy determines that the responses to these types of requests are served from the cache.

For some features, you configure actions as part of a more complex set of instructions known as a profile.

How different Citrix ADC features use policies

The Citrix ADC supports various features that rely on policies for operation. The following table summarizes how the Citrix ADC features use policies.

Feature Name	Policy Type	How You Use Policies in the Feature
System	Classic	For the Authentication function, policies contain authentication schemes for different authentication methods. For example, you can configure LDAP and certificate-based authentication schemes. You also configure policies in the Auditing function.
DNS	Advanced	To determine how to perform DNS resolution for requests.
SSL	Classic and Advanced	To determine when to apply an encryption function and add certificate information to clear text. To provide end-to-end security, after a message is decrypted, the SSL feature re-encrypts clear text and uses SSL to communicate with Web servers.
Compression	Classic and Advanced	To determine what type of traffic is compressed.
Integrated Caching	Advanced	To determine whether HTTP responses are cacheable.
Responder	Advanced	To configure the behavior of the Responder function.
Protection Features	Classic	To configure the behavior of the Filter, SureConnect, and Priority Queuing functions.

Feature Name	Policy Type	How You Use Policies in the Feature
Content Switching	Classic and Advanced	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type, and associated cache server.
AAA - Traffic Management	Classic. Exceptions: Traffic policies support only advanced policy infrastructure (PI)s aand authorization policies support both classic and advanced policy infrastructure (PI).	To check for client-side security before users log in and establish a session. Traffic policies, which determine whether single sign-on (SSO) is required, use only the default syntax. Authorization policies authorize users and groups that access intranet resources through the appliance.
Cache Redirection	Classic	To determine whether responses are served from a cache or from an origin server.

Feature Name	Policy Type	How You Use Policies in the Feature
Rewrite	Advanced	To identify HTTP data that you want to modify before serving. The policies provide rules for modifying the data. For example, you can modify HTTP data to redirect a request to a new home page, or a new server, or a selected server based on the address of the incoming request, or you can modify the data to mask server information in a response for security purposes. The URL Transformer function identifies URLs in HTTP transactions and text files for the purpose of evaluating whether a URL should be transformed.
Application Firewall	Classic and Advanced	To identify characteristics of traffic and data that should or should not be admitted through the firewall.
Citrix Gateway, Clientless Access function	Advanced	To define rewrite rules for general Web access using the Citrix Gateway.
Citrix Gateway	Classic	To determine how the Citrix Gateway performs authentication, authorization, auditing, and other functions.

About actions and profiles

Policies do not themselves take action on data. Policies provide read-only logic for evaluating traffic. To enable a feature to perform an operation based on a policy evaluation, you configure actions or profiles and associate them with policies.

Note: Actions and profiles are specific to particular features. For information about assigning actions and profiles to features, see the documentation for the individual features.

About actions

Actions are steps that the Citrix ADC takes, depending on the evaluation of the expression in the policy. For example, if an expression in a policy matches a particular source IP address in a request, the action that is associated with this policy determines whether the connection is permitted.

The types of actions that the Citrix ADC can take are feature specific. For example, in Rewrite, actions can replace text in a request, change the destination URL for a request, and so on. In Integrated Caching, actions determine whether HTTP responses are served from the cache or an origin server.

In some Citrix ADC features actions are predefined, and in others they are configurable. In some cases, (for example, Rewrite), you configure the actions using the same types of expressions that you use to configure the associated policy rule.

About profiles

Some Citrix ADC features enable you to associate profiles, or both actions and profiles, with a policy. A profile is a collection of settings that enable the feature to perform a complex function. For example, in the application firewall, a profile for XML data can perform multiple screening operations, such as examining the data for illegal XML syntax or evidence of SQL injection.

Use of actions and profiles in particular features

The following table summarizes the use of actions and profiles in different Citrix ADC features. The table is not exhaustive. For more information about specific uses of actions and profiles for a feature, see the documentation for the feature.

Feature	Use of an Action	Use of a Profile
Application firewall	Synonymous with a profile	All application firewall features use profiles to define complex behaviors, including pattern-based learning. You add these profiles to policies.
Citrix Gateway	The following features of the Citrix Gateway use actions: Pre-Authentication. Uses Allow and Deny actions. You add these actions to a profile., Authorization. Uses Allow and Deny actions. You add these actions to a policy. TCP Compression. Uses various actions. You add these actions to a policy.	The following features use a profile: Pre-Authentication, Session, Traffic, and Clientless Access. After configuring the profiles, you add them to policies.
Rewrite	You configure URL rewrite actions and add them to a policy.	Not used.
Integrated Caching	You configure caching and invalidation actions within a policy	Not used.
AAA - Traffic Management	You select an authentication type, set an authorization action of ALLOW or DENY, or set auditing to SYSLOG or NSLOG.	You can configure session profiles with a default timeout and authorization action.
Protection Features	You configure actions within policies for the following functions: Filter, Compression, Responder, and SureConnect.	Not used.
SSL	You configure actions within SSL policies	Not used.

Feature	Use of an Action	Use of a Profile
System	The action is implied. For the Authentication function, it is either Allow or Deny. For Auditing, it is Auditing On or Auditing Off.	Not used.
DNS	The action is implied. It is either Drop Packets or the location of a DNS server.	Not used.
SSL Offload	The action is implied. It is based on a policy that you associate with an SSL virtual server or a service.	Not used.
Compression	Determine the type of compression to apply to the data	Not used.
Content Switching	The action is implied. If a request matches the policy, the request is directed to the virtual server associated with the policy.	Not used.
Cache Redirection	The action is implied. If a request matches the policy, the request is directed to the origin server.	Not used.

About policy bindings

A policy is associated with, or bound to, an entity that enables the policy to be invoked. For example, you can bind a policy to request-time evaluation that applies to all virtual servers. A collection of policies that are bound to a particular bind point constitutes a policy bank.

Following is an overview of different types of bind points for a policy:

- Request time global. A policy can be available to all components in a feature at request time.
- Response time global. A policy can be available to all components in a feature at response time.
- Request time, virtual server-specific.

A policy can be bound to request-time processing for a particular virtual server. For example, you

can bind a request-time policy to a cache redirection virtual server to ensure that particular requests are forwarded to a load balancing virtual server for the cache, and other requests are sent to a load balancing virtual server for the origin.

- Response time, virtual server-specific. A policy can also be bound to response-time processing for a particular virtual server.
- User-defined policy label. For advanced policy infrastructure (PI), you can configure custom groupings of policies (policy banks) by defining a policy label and collecting a set of related policies under the policy label.
- Other bind points. The availability of additional bind points depends on type of policy (classic or advanced policies), and specifics of the relevant Citrix ADC feature. For example, classic policies that you configure for the Citrix Gateway have user and group bind points.

For additional information about advanced policy bindings, see Bind policies that use the advanced policies and Configure a policy bank for a virtual server. For additional information about classic policy bindings, see Configure a classic policy.

About evaluation order of policies

For classic policies, policy groups and policies within a group are evaluated in a particular order, depending on the following:

- The bind point for the policy, for example, whether the policy is bound to request-time processing for a virtual server or global response-time processing. For example, at request time, the Citrix ADC evaluates all request-time classic policies before evaluating any virtual server-specific policies.
- The priority level for the policy. For each point in the evaluation process, a priority level that is assigned to a policy determines the order of evaluation relative to other policies that share the same bind point. For example, when the Citrix ADC evaluates a bank of request-time, virtual server-specific policies, it starts with the policy that is assigned to the lowest priority value. In classic policies, priority levels must be unique across all bind points.

For Advanced policies, as with classic policies, the Citrix ADC selects a grouping, or bank, of policies at a particular point in overall processing. Following is the order of evaluation of the basic groupings, or banks, of Advanced policies:

1. Request-time global override
2. Request-time, virtual server-specific (one bind point per virtual server)
3. Request-time global default
4. Response-time global override
5. Response-time virtual server-specific
6. Response-time global default

However, within any of the preceding banks of policies, the order of evaluation is more flexible than in classic policies. Within a policy bank, you can point to the next policy to be evaluated regardless of the priority level, and you can invoke policy banks that belong to other bind points and user-defined policy banks.

Order of evaluation based on traffic flow

As traffic flows through the Citrix ADC and is processed by various features, each feature performs policy evaluation. Whenever a policy matches the traffic, the Citrix ADC stores the action and continues processing until the data is about to leave the Citrix ADC. At that point, the Citrix ADC typically applies all matching actions. Integrated Caching, which only applies a final Cache or NoCache action, is an exception.

Some policies affect the outcome of other policies. Following are examples:

- If a response is served from the integrated cache, some other Citrix ADC features do not process the response or the request that initiated it.
- If the Content Filtering feature prevents a response from being served, no subsequent features evaluate the response.

If the application firewall rejects an incoming request, no other features can process it.

Classic and advanced policy expressions

September 14, 2021

One of the most fundamental components of a policy is its rule. A policy rule is a logical expression that enables the policy to analyze traffic. Most of the policy's functionality is derived from its expression.

An expression matches characteristics of traffic or other data with one or more parameters and values. For example, an expression can enable the Citrix ADC to accomplish the following:

- Determine whether a request contains a certificate.
- Determine the IP address of a client that sent a TCP request.
- Identify the data that an HTTP request contains (for example, a popular spreadsheet or word processing application).
- Calculate the length of an HTTP request.

About classic expressions

Classic expressions enable you to evaluate basic characteristics of data. They have a structured syntax that performs string matching and other operations.

Following are a few simple examples of classic expressions:

- An HTTP response contains a particular type of Cache Control header.

res.http.header Cache-Control contains public

- An HTTP response contains image data.

res.http.header Content-Type contains image

- An SSL request contains a certificate.

req.ssl.client.cert exists

About advanced policy expressions

Any feature that uses default syntax policies also uses Advanced expressions. For information about which features use Advanced policies, see the table Citrix ADC Feature, Policy Type, and Policy Usage.

Advanced policy expressions have a few other uses. In addition to configuring Advanced expressions in policy rules, you configure Advanced
expressions in the following situations:

- Integrated Caching:

  You use Advanced policy expressions to configure a selector for a content group in the integrated cache.

- Load Balancing:

  You use Advanced policy expressions to configure token extraction for a load balancing virtual server that uses the TOKEN method for load balancing.

- Rewrite:

  You use Advanced policy expressions to configure rewrite actions.

- Rate-based policies:

  You use Advanced policy expressions to configure limit selectors when configuring a policy to control the rate of traffic to various servers.

Following are a few simple examples of Advanced policy expressions:

- An HTTP request URL contains no more than 500 characters.

```
http.req.url.length \<= 500
```


- An HTTP request contains a cookie that has fewer than 500 characters.

```
http.req.cookie.length \< 500
```

- An HTTP request URL contains a particular text string.

```
http.req.url.contains(".html")
```

Converting policy expressions using the NSPEPI tool

September 17, 2021

> **Note:**
>
> You can download the NSPEPI and preconfig check tool from the public GitHub. For more information, see Github NEPEPI page and GitHub preconfig page for detailed instructions to download the tools. We recommend customers to use the tools available in GitHub for the most complete and up-to-date version.

Classic policy-based features and functionalities are deprecated from NetScaler 12.0 build 56.20 onwards. As an alternative, Citrix recommends you to use the Advanced policy infrastructure. As part of this effort, when you upgrade to Citrix ADC 12.1 build 56.20 or later, you must replace the Classic policy-based features and functionalities to its corresponding non-deprecated features and functionalities. Also, you must convert Classic policies and expressions to Advanced policies and expressions. Also, all new Citrix ADC features support only Advanced policy infrastructure.

The `nspepi` tool can perform the following:

1. Convert Classic policy expressions to Advanced policy expressions.
2. Convert certain Classic policies and their entity bindings to Advanced policies and bindings.
3. Convert a few more deprecated features to their corresponding non-deprecated features.
4. Convert classic filter commands to advanced filter commands.

> **Note:**
>
> After the `nspepi` tool successfully converts the ns.conf config file, the tool displays the converted file as a new file with a prefix, “new_”. If the converted config file has errors or warnings, you must manually fix them as part of the conversion process. Once converted, you must test the file in the test environment and then use it to replace the actual ns.conf config file. After testing, you must reboot the appliance for the newly converted or fixed ns.conf config file.

Features that only support Classic policies or expressions are deprecated and they can be replaced by the corresponding non-deprecated features.

> **Note:**
>
> Information pertaining to the older version of the `nspepi` tool is available in a PDF format. For more information, see Classic policy conversion using nspepi tool prior to 12.1-51.16 PDF.

Conversion warnings and error files

Before you use the tool for your conversion, there are few warnings to keep in mind:

1. All warnings and errors are output to the console. There is a warning file created where the configuration files are stored.
2. The warnings and error file has the same name as the input file but with a prefix “warn_” added to the file name. During expression conversion (when using -e), the warnings show up in the current directory with a name “warn_expr”.

> **Note:**
>
> This file is in a standard log file format, with date/time stamp and log level. Previous instances of the file are kept with suffixes like “.1”, “.2”, and so forth as the tool is run multiple times. At most 10 instances will be kept.

Converted file format

When converting a configuration file (using “-f”), the converted file is put into the same directory as where the input configuration file exists with the same name but a prefix “new_”.

Commands or features handled by the nspepi conversion tool

Following are the commands handled during the auto conversion process.

- The following Classic policies and their expressions are converted to Advanced policies and expressions. The conversion includes entity bindings and global bindings.

1. add appfw policy
2. add cmp policy
3. add cr policy
4. add cs policy
5. add tm sessionPolicy
6. add filter action
7. add filter policy
8. filter policy binding to load balancing, content switching, cache redirection, and global.

> **Note:**
>
> However, for “add tm sessionPolicy”, you cannot bind to global override in Advanced policies.

- The rule parameter configured in “add lb virtual server” is converted from Classic expression to Advanced expression.
- The SPDY parameter configured in the “add ns httpProfile” or the “set ns httpProfile” command is changed to “-http2 ENABLED”.
- Named expressions (“add policy expression” commands). Each Classic named policy expression is converted to its corresponding Advanced named expression with “nspepi_adv_” set as the prefix. In addition, usage of named expressions for the converted Classic expressions is changed to the corresponding Advanced named expressions. In addition, every named expression has two named expressions, where one is Classic and the other one is Advanced (as shown below).
- Tunnel TrafficPolicy conversion is supported.
- Handling builtin-in classic policy bindings in CMP, CR, and Tunnel.
- Patclass feature is converted to Pat set feature.
- “-pattern” parameter in the “add rewrite action” command is converted to use “-search” parameter.
- SYS.EVAL_CLASSIC_EXPR is converted to the equivalent non-deprecated advanced expression. These expressions can be seen in any command where advanced expressions are allowed.
- Q and S prefixes of advanced expressions are converted to equivalent non-deprecated advanced expressions. These expressions can be seen in any command where advanced expressions are allowed.

For example:

```
add policy expression classic_expr ns_true
Converts to:
add policy expression classic_expr ns_true
add policy expression nspepi_adv_classic_expr TRUE
<!--NeedCopy-->
```

- The policyType parameter configured in the “set cmp parameter” command is removed. By default, the policy type is “Advanced”.

Convert classic filter commands to advanced filter commands

The `nspepi` tool can convert commands based on classic filter actions such as add, bind and so forth to advanced filter commands.

However, the nepepi tool does not support the following filter commands.

1. add filter action <`action Name`> FORWARD <`service name`>

2. add filter action <action name> ADD prebody
3. add filter action <action name> ADD postbody

> **Note:**
>
> 1. If there are existing rewrite or responder features in ns.conf and their policies are bound globally with the GOTO expression as END or USER_INVOCATION_RESULT and the bind type is REQ_X or RES_X then the tool converts bind filter commands partially and comments out. Warning is displayed to put manual effort.
> 2. If there are existing rewrite or responder features and their policies are bound to virtual servers(for example, load balancing, content switching or cache redirect) of type HTTPS with GOTO - END or USER_INVOCATION_RESULT, the tool converts bind filter commands partially and then comments out. Warning is displayed to put manual effort.

Example

Following is a sample input:

```
add lb vserver v1 http 1.1.1.1 80 -persistenceType NONE -cltTimeout
    9000
add cs vserver csv1 HTTP 1.1.1.2 80 -cltTimeout 180 -persistenceType
    NONE
add cr vserver crv1 HTTP 1.1.1.3 80 -cacheType FORWARD
add service svc1 1.1.1.4 http 80
add filter action fact_add add 'header:value'
add filter action fact_variable add 'H1:%%HTTP.TRANSID%%'
add filter action fact_prebody add prebody
add filter action fact_error_act1 ERRORCODE 200 "<HTML>Good URL</HTML>"
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_add_res -rule ns_true -resAction fact_add
add filter policy fpol_error_res -rule ns_true -resAction
    fact_error_act1
add filter policy fpol_error_req -rule ns_true -reqAction
    fact_error_act1
add filter policy fpol_add_req -rule ns_true -reqAction fact_add
add filter policy fpol_variable_req -rule ns_true -reqAction
    fact_variable
add filter policy fpol_variable_res -rule ns_true -resAction
    fact_variable
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
```



```
    fact_forward_act1
bind lb vserver v1 -policyName fpol_add_res
bind lb vserver v1 -policyName fpol_add_req
bind lb vserver v1 -policyName fpol_error_res
bind lb vserver v1 -policyName fpol_error_req
bind lb vserver v1 -policyName fpol_variable_res
bind lb vserver v1 -policyName fpol_variable_req
bind lb vserver v1 -policyName fpol_forward_req
bind cs vserver csv1 -policyName fpol_add_req
bind cs vserver csv1 -policyName fpol_add_res
bind cs vserver csv1 -policyName fpol_error_res
bind cs vserver csv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_add_req
bind cr vserver crv1 -policyName fpol_add_res
bind cr vserver crv1 -policyName fpol_error_res
bind cr vserver crv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_forward_req
bind filter global fpol_add_req
bind filter global fpol_add_res
bind filter global fpol_error_req
bind filter global fpol_error_res
bind filter global fpol_variable_req
bind filter global fpol_variable_res
bind filter global fpol_variable_res -state DISABLED
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
After conversion, warning/error messages will be displayed for manual
    effort.
Warning files:
cat warn_<input file name>:
2019-11-07 17:13:34,724: ERROR - Conversion of [add filter action
    fact_prebody add prebody] not supported in this tool.
2019-11-07 17:13:34,739: ERROR - Conversion of [add filter action
    fact_forward_act1 FORWARD svc1] not supported in this tool.
2019-11-07 17:13:38,042: ERROR - Conversion of [add filter policy
    fpol_prebody_req -rule ns_true -reqAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:38,497: ERROR - Conversion of [add filter policy
    fpol_prebody_res -rule ns_true -resAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:39,035: ERROR - Conversion of [add filter policy
    fpol_forward_req -rule ns_true -reqAction fact_forward_act1] not
    supported in this tool.
2019-11-07 17:13:39,060: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
```



```
       fixed ordering of the types of bindings without interleaving, except
        that global bindings are allowed before all other bindings and
       after all bindings. If you have global bindings in the middle of non
       -global bindings or any other interleaving then you will need to
       reorder all your bindings for that feature and direction. Refer to
       nspepi documentation. If command is required please take a backup
       because comments will not be saved in ns.conf after triggering 'save
        ns config': bind filter global fpol_variable_res -state DISABLED
53
54
55 <!--NeedCopy-->
```

Following is a sample output. All converted commands are commented.

```
1  cat new_<input file name>
2  add rewrite action fact_add insert_http_header header "\"value\""
3  add filter action fact_prebody add prebody
4  add filter action fact_forward_act1 FORWARD svc1
5  add filter policy fpol_prebody_req -rule ns_true -reqAction
       fact_prebody
6  add filter policy fpol_prebody_res -rule ns_true -resAction
       fact_prebody
7  add filter policy fpol_forward_req -rule ns_true -reqAction
       fact_forward_act1
8  bind lb vserver v1 -policyName fpol_forward_req
9  bind cr vserver crv1 -policyName fpol_forward_req
10 #bind filter global fpol_variable_res -state DISABLED
11 bind filter global fpol_prebody_req
12 bind filter global fpol_forward_req
13 add rewrite action nspepi_adv_fact_variable insert_http_header H1 HTTP.
       RES.TXID
14 add rewrite action fact_variable insert_http_header H1 HTTP.REQ.TXID
15 add responder action fact_error_act1 respondwith "HTTP.REQ.VERSION.
       APPEND(\" 200 OK\\r
16 nConnection: close\\r
17 nContent-Length: 21\\r\\n\\r
18 n<HTML>Good URL</HTML>\")"
19 add rewrite action nspepi_adv_fact_error_act1 replace_http_res "HTTP.
       REQ.VERSION.APPEND(\" 200 OK\\r
20 nConnection: close\\r
21 nContent-Length: 21\\r\\n\\r
22 n<HTML>Good URL</HTML>\")"
23 add rewrite policy fpol_add_res TRUE fact_add
24 add rewrite policy fpol_error_res TRUE nspepi_adv_fact_error_act1
25 add responder policy fpol_error_req TRUE fact_error_act1
```



```
add rewrite policy fpol_add_req TRUE fact_add
add rewrite policy fpol_variable_req TRUE fact_variable
add rewrite policy fpol_variable_res TRUE nspepi_adv_fact_variable
set cmp parameter -policyType ADVANCED
bind rewrite global fpol_add_req 100 NEXT -type REQ_DEFAULT
bind rewrite global fpol_variable_req 200 NEXT -type REQ_DEFAULT
bind rewrite global fpol_add_res 100 NEXT -type RES_DEFAULT
bind rewrite global fpol_error_res 200 NEXT -type RES_DEFAULT
bind rewrite global fpol_variable_res 300 NEXT -type RES_DEFAULT
bind responder global fpol_error_req 100 END -type REQ_DEFAULT
bind lb vserver v1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_res -type RESPONSE -priority
    200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_res -type RESPONSE -
    priority 300 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_add_req -type REQUEST -priority 100
     -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_req -type REQUEST -
    priority 200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_req -type REQUEST -priority
    100 -gotoPriorityExpression END
bind cs vserver csv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END
bind cr vserver crv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END

<!--NeedCopy-->
```


Convert classic filter commands to advanced feature commands if existing rewrite or responder policy bindings have goto expression END or USE_INNVOCATION

In this conversion, if a rewrite policy bound to one or more virtual servers and if the server has END or USE_INVOCATION_RESULT, the tool comments out the commands.

Example

Following is a sample input command:

```
COPY
add filter policy fpol1 -rule ns_true -resAction reset
add filter policy fpol2 -rule ns_true -reqAction reset
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind lb vserver v1_http -policyName fpol1
bind cs vserver csv1_http -policyName fpol1
bind lb vserver v2_http -policyName fpol2
bind cs vserver csv2_http -policyName fpol2
bind cr vserver crv2_http -policyName fpol2
bind filter global fpol1 -priority 100
bind filter global fpol2 -priority 100
<!--NeedCopy-->
```

Following is a sample output command:

```
COPY
```



```
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
add rewrite policy fpol1 TRUE RESET
add responder policy fpol2 TRUE RESET
#bind lb vserver v1_http -policyName fpol1 -type RESPONSE
#bind cs vserver csv1_http -policyName fpol1 -type RESPONSE
#bind rewrite global fpol1 100 -type RES_DEFAULT
#bind lb vserver v2_http -policyName fpol2 -type REQUEST
#bind cs vserver csv2_http -policyName fpol2 -type REQUEST
#bind cr vserver crv2_http -policyName fpol2 -type REQUEST
#bind responder global fpol2 100 -type REQ_DEFAULT
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
   -invoke policylabel pl
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
   REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
   gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
   gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
   gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
   gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
   gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST-

<!--NeedCopy-->
```

Commands or features not handled by the nspepi conversion tool

Following are some commands that are not handled as part of the auto conversion process.

- Some bindings cannot be converted if there are certain interleaving of priorities between global and non-global bind points, between users and groups, and also among bindings to different entities. These have the affected configuration commented out and an error produced. Such configurations must be converted manually.
- Both Classic and Advanced policies can be bound to cmp global. There are many cases where the functionality changes once Classic policies are converted to Advanced policies. We have converted commands that can be solved by commenting out some policies. Still there are some

commands that cannot be converted. In such cases an error will be produced and conversion has to be done manually.
- Not all uses of Classic built-in named expressions are converted to equivalent Advanced named expressions.
- Client security expressions are not handled.
- The "-precedence" option for content switching and cache redirection virtual servers is not handled.
- Sure Connect (SC)
- Priority Queuing (PQ)
- HTTP Denial of Service (HDOS)
- HTML Injection
- Authentication
- Authorization
- VPN
- Syslog
- Nslog
- File based Classic expressions are not handled.

> **Note:**
>
> For some features like Patclass/filter, the command syntax is changed. If there are cmd policies, then cmd policies might need to be changed depending on customer requirement.

Known Issues

The following errors can be produced by the `nspepi` tool:

- If there is an issue when converting an expression.
- If a named policy expression uses the -clientSecurityMessage parameter because this parameter is not supported in the Advanced policy expression.

> **Note:**
>
> All classic policy bindings with -state option disabled are commented out. The -state option is not available for Advanced policy bindings.

Running the nspepi tool

The following is a command line example for running the `nspepi` tool. This tool is run from the command line of the shell (you need to type the "shell" command to the NetScaler"CLI" to get to that). Either "-f" or"-e" must be specified to perform a conversion. Use of "-d" is intended for Citrix personnel to analyze for support purposes.


```
usage: nspepi [-h] (-e <classic policy expression> | -f <path to ns
    config file>)[-d] [-v] [-V

Convert classic policy expressions to advanced policy expressions and
    deprecated commands to non-deprecated
commands.

optional arguments:
-h, --help show this help message and exit
-e <classic policy expression>, --expression <classic policy expression
    >
convert classic policy expression to advanced policy
expression (maximum length of 8191 allowed)
-f <path to ns config file>, --infile <path to ns config file>
convert netscaler config file
-d, --debug log debug output
-v, --verbose show verbose output
-V, --version show program's version number and exit
<!--NeedCopy-->
```

Usage Examples:

1. `nspepi -e "req.tcp.destport == 80"`
2. `nspepi -f ns.conf`

Following are few examples of running the `nspepi` tool by using the CLI

Example output for –e parameter:

```
root@ns# nspepi -e "req.http.header foo == \"bar\""
"HTTP.REQ.HEADER(\"foo\").EQ(\"bar\")"
<!--NeedCopy-->
```

Example output for -f parameter:

```
root@ns# cat sample.conf
add c**Input**r vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout
    180 -originUSIP OFF
add cr policy cr_pol1 -rule ns_true
bind cr vserver cr_vs -policyName cr_pol1
<!--NeedCopy-->
```

Running nspepi with -f parameter:

```
nspepi -f sample.conf
<!--NeedCopy-->
```


Converted config is available in a new file `new_sample.conf`.
Check the `warn_sample.conf` file for any warnings or errors that might have been generated.

Example output of -f parameter along with -v parameter

```
nspepi -f sample.conf -v
INFO - add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180
    -originUSIP OFF
INFO - add cr policy cr_pol1 -rule TRUE -action ORIGIN
INFO - bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
   gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

Converted config is available in a new file `new_sample.conf`.
Check the `warn_sample.conf` file for any warnings or errors that might have been generated.

Converted Config file:

```
root@ns# cat new_sample.conf
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
   originUSIP OFF
add cr policy cr_pol1 -rule TRUE -action ORIGIN
set cmp parameter -policyType ADVANCED
bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
   gotoPriorityExpression END -type REQUEST

<!--NeedCopy-->
```

Example output of a sample configuration with no errors or warnings:

```
nspepi -f sample_2.conf
<!--NeedCopy-->
```

Converted config is available in a new file `new_sample_2.conf`.
Check the `warn_sample_2.conf` file for any warnings or errors that might have been generated.

Example output of a sample configuration with warnings:

```
root@ns# cat sample_2.conf
add policy expression security_expr "req.tcp.destport == 80" -
   clientSecurityMessage "Not allowed"
set cmp parameter -policyType CLASSIC
add cmp policy cmp_pol1 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol2 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
bind cmp global cmp_pol1
```



```
8  bind cmp global cmp_pol2 -state DISABLED
9  bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
       RES_DEFAULT
10 bind lb vserver lb_vs -policyName cmp_pol2
11 root@ns#
12 <!--NeedCopy-->
```

Example of running nspepi with -f parameter:

```
1  root@ns# nspepi -f sample_2.conf
2  ERROR - Error in converting expression security_expr : conversion of
       clientSecurityMessage based expression is not supported.
3  WARNING - Following bind command is commented out because state is
       disabled. Advanced expressions only have a fixed ordering of the
       types of bindings without interleaving, except that global bindings
       are allowed before all other bindings and after all bindings. If you
        have global bindings in the middle of non-global bindings or any
       other interleaving then you will need to reorder all your bindings
       for that feature and direction. Refer to nspepi documentation. If
       command is required please take a backup because comments will not
       be saved in ns.conf after triggering 'save ns config': bind cmp
       global cmp_pol2 -state DISABLED
4  Warning - Bindings of advanced CMP policies to cmp global are commented
        out, because initial global cmp parameter is classic but advanced
       policies are bound. Now global cmp parameter policy type is set to
       advanced. If commands are required please take a backup because
       comments will not be saved in ns.conf after triggering 'save ns
       config'. Advanced expressions only have a fixed ordering of the
       types of bindings without interleaving, except that global bindings
       are allowed before all other bindings and after all bindings. If you
        have global bindings in the middle of non-global bindings or any
       other interleaving then you will need to reorder all your bindings
       for that feature and direction. Refer to nspepi documentation.
5  root@ns#
6  <!--NeedCopy-->
```

Converted file:

```
1  root@ns# cat new_sample_2.conf
2  add policy expression security_expr "req.tcp.destport == 80" -
       clientSecurityMessage "Not allowed"
3  set cmp parameter -policyType ADVANCED
4  add cmp policy cmp_pol1 -rule TRUE -resAction COMPRESS
5  add cmp policy cmp_pol2 -rule TRUE -resAction COMPRESS
6  add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
```



```
#bind cmp global cmp_pol2 -state DISABLED
#bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
    RES_DEFAULT
bind cmp global cmp_pol1 -priority 100 -gotoPriorityExpression END -
   type RES_DEFAULT
bind lb vserver lb_vs -policyName cmp_pol2 -priority 100 -
   gotoPriorityExpression END -type RESPONSE
root@ns#
<!--NeedCopy-->
```

Warning file:

```
root@ns# cat warn_sample_2.conf
2019-02-28 06:20:10,590: ERROR - Error in converting expression
   security_expr : conversion of clientSecurityMessage based expression
    is not supported.
2019-02-28 06:20:12,187: WARNING - Following bind command is commented
   out because state is disabled. Advanced expressions only have a
   fixed ordering of the types of bindings without interleaving, except
    that global bindings are allowed before all other bindings and
   after all bindings. If you have global bindings in the middle of non
   -global bindings or any other interleaving then you will need to
   reorder all your bindings for that feature and direction. Refer to
   nspepi documentation. If command is required please take a backup
   because comments will not be saved in ns.conf after triggering 'save
    ns config': bind cmp global cmp_pol2 -state DISABLED
2019-02-28 06:20:12,191: WARNING - Bindings of advanced CMP policies to
    cmp global are commented out, because initial global cmp parameter
   is classic but advanced policies are bound. Now global cmp parameter
    policy type is set to advanced. If commands are required please
   take a backup because comments will not be saved in ns.conf after
   triggering 'save ns config'. Advanced expressions only have a fixed
   ordering of the types of bindings without interleaving, except that
   global bindings are allowed before all other bindings and after all
   bindings. If you have global bindings in the middle of non-global
   bindings or any other interleaving then you will need to reorder all
    your bindings for that feature and direction. Refer to nspepi
   documentation.
root@ns#
<!--NeedCopy-->
```


Binding Priorities

Advanced policies do not allow arbitrary interleaving by priority between global and non-global and between different binding types. If you rely on such interleaving of Classic policy priorities, you need to adjust the priorities to conform to the Advanced policy rules and to get the behavior you desire. Priorities in Advanced policies are local to a bind point. A bind point is a unique combination of protocol, feature, direction, and entity (entities are specific virtual servers, users, groups, services, and either global override or global default). Policy priorities are not followed across bind points.

For a given protocol, feature, and direction the order of evaluation of Advanced policies is given below:

- Global override.
- (Current) authentication, authorization, and auditing user.
- Authentication, authorization, and auditing groups (that the user is a member of) in order of weight - ordering is undefined if two or more groups have the same weight.
- LB virtual server that either the request was received on or that Content Switching selected.
- Content switching virtual server, cache redirection virtual server that the request was received on.
- Service selected by load balancing.
- Global default.

For authorization policy evaluation, the order is:

- Systems override.
- Load balancing virtual server that either the request was received on or that CS selected.
- Content switching virtual server that the request was received on.
- System default.

Within each bind point, the policies are evaluated in order of priority from lowest numbered to highest numbered. Policies are only evaluated for the protocol used and the direction that the message was received from.

Classic policy bindings that require manual reprioritization

Here are some types of Classic policy bindings that require manual reprioritization to accomplish your needs. All these are for a given feature and the direction.

- Classic priorities that increase in priority number opposite to the direction of the above entity type lists. For example a content switching virtual server binding lower than a load balancing virtual server binding.
- Classic priorities that interleave authentication, authorization, and auditing groups. One part of one group is before some other group and yet another part is after part of that other group.
- Classic priorities that increase in number other than the order of weights of authentication, authorization, and auditing groups.

- Classic global priorities that are less than some non-global priority and the same global priorities are greater than some other non-global priority (in other words, any segment of priorities that are a non-global, followed by one or more globals, followed by a non-global).

Classic Policy Deprecation FAQs

September 16, 2021

- **What are the classic policies deprecated from Citrix ADC 12.0 release onwards?**

  All the features and functionalities mentioned in the Deprecated policies table are deprecated from Citrix ADC release 12.0 build 56.20. Citrix recommends you to see the following tables (in PDF format) for deprecated feature and policy details.

  - Table 1 for deprecated policies and its alternative.
  - Table 2 for deprecated Citrix ADC functionalities and its alternative with configuration details.

- **How can I convert classic policy based feature and functionalities to Advanced policy?**

  You can use the Citrix ADC proprietary `nspepi` tool to convert commands, expressions, and configurations. `nspepi` tool helps to convert all the classic expressions in the Citrix ADC configuration to the Advanced policy expressions. For more information about the `nspepi` tool, see Converting policy expressions using NSPEPI tool.

- **From which release are classic policy based features and functionalities deprecated?**

  Citrix ADC 12.0 build 56.20 and later.

- **What steps to follow when I upgrade my appliance to a build that does not support the classic policy based features?**

  Citrix recommends using advanced policies before you upgrade your appliance to releases later than Citrix ADC release 13.0. For more information, see Advanced Policies.

- **How long will the deprecated features be supported on a Citrix ADC appliance?**

  Citrix will not support the classic policy and its usage in releases later than Citrix ADC release 13.0.

- **Do I have to reboot my appliance after converting the configuration file?**

  Yes, you have to reboot the Citrix ADC instance after successful conversion of the `ns.config` file.

Before you proceed

September 14, 2021

Before configuring expressions and policies, be sure you understand the relevant Citrix ADC feature and the structure of your data, as follows:

- Read the documentation on the relevant feature.
- Look at the data stream for the type of data that you want to configure.

You may want to run a trace on the type of traffic or content that you want to configure. This will give you an idea of the parameters and values, and operations on these parameters and values, that you need to specify in an expression.

Note: The Citrix ADC supports either classic or Advanced policy within a feature. You cannot have both types in the same feature. Over the past few releases, some Citrix ADC features have migrated from using classic policies and expressions to Advanced policy and expressions. If a feature of interest to you has changed to the Advanced policy format, you may have to manually migrate the older information. Following are guidelines for deciding if you need to migrate your policies:

- If you configured classic policies in a version of the Integrated Caching feature prior to release 9.0 and then upgrade to version 9.0 or later, there is no impact. All legacy policies are migrated to the Advanced policy format.
- For other features, you need to manually migrate classic policies and expressions to the Advanced syntax if the feature has migrated to the Advanced policy.

Configure advanced policy infrastructure

September 14, 2021

You can create advanced policies for various Citrix ADC features, including DNS, Rewrite, Responder, and Integrated Caching, and the clientless access function in the Citrix Gateway. Policies control the behavior of these features.

When you create a policy, you assign it a name, a rule (an expression), feature-specific attributes, and an action that is taken when data matches the policy. After creating the policy, you determine when it is invoked by binding it globally or to either request-time or response-time processing for a virtual server.

Policies that share the same bind point are known as a *policy bank*. For example, all policies that are bound to a virtual server constitute the policy bank for the virtual server. When binding the policy, you assign it a priority level to specify when it is invoked relative to other policies in the bank. In addition

to assigning a priority level, you can configure an arbitrary evaluation order for policies in a bank by specifying Goto expressions.

In addition to policy banks that are associated with a built-in bind point or a virtual server, you can configure *policy labels*. A policy label is a policy bank that is identified by an arbitrary name. You invoke a policy label, and the policies in it, from a global or virtual-server-specific policy bank. A policy label or a virtual-server policy bank can be invoked from multiple policy banks.

For some features, you can use the policy manager to configure and bind policies.

Rules for names in identifiers used in policies

September 14, 2021

The names of identifiers in the named expression, HTTP callout, pattern set, and rate limiting features must begin with an ASCII alphabet or an underscore (_). The remaining characters can be ASCII alphanumeric characters or underscores (_).

The names of these identifiers must not begin with the following reserved words:

- The words ALT, TRUE, or FALSE or the Q or S one-character identifier.
- The special-syntax indicator RE (for regular expressions) or XP (for XPath expressions).
- Expression prefixes, which currently are the following:
  - CLIENT
  - EXTEND
  - HTTP
  - SERVER
  - SYS
  - TARGET
  - TEXT
  - URL
  - MYSQL
  - MSSQL

Additionally, the names of these identifiers cannot be the same as the names of enumeration constants used in the policy infrastructure. For example, the name of an identifier cannot be IGNORECASE, YEAR, or LATIN2_CZECH_CS (a MySQL character set).

Note: The Citrix ADC appliance performs a case-insensitive comparison of identifiers with these words and enumeration constants. For example, names of the identifiers cannot begin with TRUE, True, or true.

Create or modify a policy

September 14, 2021

All policies have some common elements. Creating a policy consists, at minimum, of naming the policy and configuring a rule. The policy configuration tools for the various features have areas of overlap, but also differences. For the details of configuring a policy for a particular feature, including associating an action with the policy, see the documentation for the feature.

To create a policy, begin by determining the purpose of the policy. For example, you may want to define a policy that identifies HTTP requests for image files, or client requests that contain an SSL certificate. In addition to knowing the type of information that you want the policy to work with, you need to know the format of the data that the policy is analyzing.

Next, determine whether the policy is globally applicable, or if it pertains to a particular virtual server. Also consider the effect that the order in which your policies are evaluated (which will be determined by how you bind the policies) will have on the policy that you are about to configure.

Create a policy by using the CLI

At the command prompt, type the following commands to create a policy and verify the configuration:

```
1 -  add responder|dns|cs|rewrite|cache policy <policyName> -rule <
     expression> [<feature-specific information>]
2
3 -  show rewrite policy <name>
4 <!--NeedCopy-->
```

Example 1:

```
1  add rewrite policy "pol_remove-ae" true "act_remove-ae"
2  Done
3  > show rewrite policy pol_remove-ae
4              Name: pol_remove-ae
5              Rule: true
6              RewriteAction: act_remove-ae
7              UndefAction: Use Global
8              Hits: 0
9              Undef Hits: 0
10             Bound to: GLOBAL RES_OVERRIDE
11             Priority: 90
12             GotoPriorityExpression: END
13 Done
14 <!--NeedCopy-->
```


Example 2:

```
add cache policy BranchReportsCachePolicy -rule q{
 http.req.url.query.value("actionoverride").contains("branchReport s")
    }
  -action cache
Done
show cache policy BranchReportsCachePolicy
            Name: BranchReportsCachePolicy
            Rule: http.req.url.query.value("actionoverride").contains("
               branchReports")
            CacheAction: CACHE
            Stored in group: DEFAULT
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
Done
<!--NeedCopy-->
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, seeConfigure advanced policy expressions: Get started.

Create or modify a policy by using the GUI

1. In the navigation pane, expand the name of the feature for which you want to configure a policy, and then click **Policies**. For example, you can select **Content Switching, Integrated Caching, DNS, Rewrite, or Responder**.
2. In the details pane, click **Add**, or select an existing policy and click **Open**. A policy configuration dialog box appears.
3. Specify values for the following parameters. (An asterisk indicates a required parameter. For a term in parentheses, see the corresponding parameter in “Parameters for creating or modifying a policy.”)
4. Click **Create**, and then click **Close**.
5. Click **Save**. A policy is added.
   Note: After you create a policy, you can view the policy’s details by clicking the policy entry in the configuration pane. Details that are highlighted and underlined are links to the corresponding entity (for example, a named expression).

Policy configuration examples

September 14, 2021

These examples show how policies and their associated actions are entered at the command line interface. In the configuration utility, the expressions would appear in the Expression window of the feature-configuration dialog box for the integrated caching or rewrite feature.

Following is an example of creating a caching policy. Note that actions for caching policies are built in, so you do not need to configure them separately from the policy.

```
1   add cache policy BranchReportsCachePolicy -rule q{
2  http.req.url.query.value("actionoverride").contains("branchReports") }
3   -action cache
4 <!--NeedCopy-->
```

Following is an example of a rewrite policy and action:

```
1  add rewrite action myAction1 INSERT_HTTP_HEADER "myHeader" "
      valueForMyHeader"
2  add rewrite policy myPolicy1 "http.req.url.contains(\"myURLstring\")"
      myAction1
3 <!--NeedCopy-->
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, see Configure advanced policy expressions: Get started.

Configure and bind policies with the policy manager

September 14, 2021

> **Warning:**
>
> Classic policy expressions are no longer supported from Citrix ADC 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies.

Some applications provide a specialized Policy Manager in the Citrix ADC configuration utility to simplify configuring policy banks. It also lets you find and delete policies and actions that are not being used.

The Policy Manager is currently available for the Rewrite, Integrated Caching, Responder, and Compression features.

The following are keyboard equivalents for the procedures in this section:

- For editing a cell in the Policy Manager, you can tab to the cell and click F2 or press the SPACE bar on the keyboard.
- To select an entry in a drop-down menu, you can tab to the entry, press the space bar to view the drop-down menu, use the UP and DOWN ARROW keys to navigate to the entry that you want, and press the space bar again to select the entry.
- To cancel a selection in a drop-down menu, press the Escape key.
- To insert a policy, tab to the row above the insertion point and press Control + Insert, or click Insert Policy.
- To remove a policy, tab to the row that contains the policy and press Delete.

Note: Note that when you delete the policy, the Citrix ADC searches the Goto Expression values of other policies in the bank. If any of these Goto Expression values match the priority level of the deleted policy, they are removed.

Configure policy bindings by using the policy manager

1. In the navigation pane, click the feature for which you want to configure policies. The choices are Responder, Integrated Caching, Rewrite or Compression.
2. In the details pane, click **Policy Manager**.
3. If you are configuring classic policy bindings for compression, in the Compression Policy Manager dialog box, click **Switch to Classic Syntax**. The dialog box switches to the classic syntax view and displays the Switch to Advanced Policy button. At any time before you complete configuring policy bindings, if you want to configure bindings for policies that use the Advanced Policy click the Switch to Advanced Policy button.
4. For features other than Responder, to specify the bind point, click Request or Response, and then click one of the request-time or response-time bind points. The options are Override Global, LB Virtual Server, CS Virtual Server, Default Global, or Policy Label. If you are configuring the Responder, the Request and Response flow types are not available.
5. To bind a policy to this bind point, click Insert Policy, and select a previously configured policy, a NOPOLICY label, or the New policy option. Depending on the option that you select, you have the following choices:
   - **New policy:** Create the policy as described in "Create or modify a policy," and then configure the priority level, GoTo expression, and policy invocation as described in the table, "Format of each entry in a policy bank."
   - **Existing policy**, **NOPOLICY**, or `NOPOLICY\<feature name\>`: Configure the priority level, GoTo expression, and policy invocation as described in the table, "Format of each

entry in a policy bank." The **NOPOLICY** or NOPOLICY\<feature name\> options are available only for policies that use Advanced Policies.

6. Repeat the preceding steps to add entries to this policy bank.
7. To modify the priority level for an entry, you can do any of the following:
   - Double-click the Priority field for an entry and edit the value.
   - Click and drag a policy to another row in the table.
   - Click Regenerate Priorities.

   In all three cases, priority levels of all other policies are modified as needed to accommodate the new value. Goto Expressions with integer values are also updated automatically. For example, if you change a priority value of 10 to 100, all policies with a Goto Expression value of 10 are updated to the value 100.
8. To change the policy, action, or policy bank invocation for an row in the table, click the down arrow to the right of the entry and do one of the following:
   - To change the policy, select another policy name or select New Policy and follow the steps in Create or modify a policy.
   - To change the Goto Expression, select Next, End, USE_INVOCATION_RESULT, or select more and enter an expression whose result returns the priority level of another entry in this policy bank.
   - To modify an invocation, select an existing policy bank, or click New Policy Label and follow the steps in Bind a policy to a policy label.
9. To unbind a policy or a policy label invocation from this bank, click any field in the row that contains the policy or policy label, and then click Unbind Policy.
10. When you are done, click Apply Changes. A message in the status bar indicates that the policy is bound successfully.

Remove unused policies by using the policy manager

1. In the navigation pane, click the feature for which you want to configure the policy bank. The choices are Responder, Integrated Caching, or Rewrite.
2. In the details pane, click <Feature Name> policy manager.
3. In the **Feature Name** > **Policy Manage**r dialog box, click **Cleanup Configuration**.
4. In the **Cleanup Configuration** dialog box, select the items that you want to delete, and then click **Remove**.
5. In the Remove dialog box, click **Yes**.
6. Click **Close**. A message in the status bar indicates that the policy is removed successfully.

Unbind a policy

September 14, 2021

If you want to re-assign a policy or delete it, you must first remove its binding.

Unbind an integrated caching, rewrite, or compression advanced policy globally by using the CLI

At the command prompt, type the following commands to unbind an integrated caching, rewrite, or compression Advanced policy globally and verify the configuration:

```
1 -  unbind cache|rewrite|cmp global <policyName> [-type req_override|
     req_default|res_override|res_default] [-priority <positiveInteger>]
2
3 -  show cache|rewrite|cmp global
4 <!--NeedCopy-->
```

Example:

```
1  > unbind cache global_nonPostReq
2  Done
3  > show cache global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 1
6
7      2)      Global bindpoint: RES_DEFAULT
8              Number of bound policies: 1
9
10 Done
11 <!--NeedCopy-->
```

The priority is required only for the “dummy” policy named NOPOLICY.

Unbind a responder policy globally by using the CLI

At the command prompt, type the following commands to unbind a responder policy globally and verify the configuration:

```
1 -  unbind responder global <policyName> [-type override|default] [-
     priority <positiveInteger>]
2
3 -  show responder global
```



```
<!--NeedCopy-->
```

Example:

```
> unbind responder global pol404Error
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1
Done
<!--NeedCopy-->
```

The priority is required only for the “dummy” policy named NOPOLICY.

Unbind a DNS policy globally by using the CLI

At the command prompt, type the following commands to unbind a DNS policy globally and verify the configuration:

```
-  unbind responder global <policyName>

-  unbind responder global
<!--NeedCopy-->
```

Example:

```
unbind dns global dfgdfg
Done
show dns global
    Policy name : dfgdfggfhg
            Priority : 100
            Goto expression : END
Done
<!--NeedCopy-->
```

Unbind an advanced policy from a virtual server by using the CLI

At the command prompt, type the following commands to unbind an Advanced policy from a virtual server and verify the configuration:

```
-  unbind cs vserver <name> -policyName <policyName> [-priority <
    positiveInteger>] [-type REQUEST|RESPONSE]

-  show lb vserver <name>
```



```
<!--NeedCopy-->
```

Example:

```
unbind cs vserver vs-cont-switch  -policyName pol1
Done
> show cs vserver vs-cont-switch
            vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
            State: UP
            Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
            Time since last state change: 0 days, 02:47:55.750
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            State Update: DISABLED
            Default:        Content Precedence: RULE
            Vserver IP and Port insertion: OFF
            Case Sensitivity: ON
            Push: DISABLED  Push VServer:
            Push Label Rule: none
Done
<!--NeedCopy-->
```

The priority is required only for the “dummy” policy named NOPOLICY.

Unbind an integrated caching, responder, rewrite, or compression Advanced policy globally by using the GUI

1. In the navigation pane, click the feature with the policy that you want to unbind (for example, Integrated Caching).
2. In the details pane, click <Feature Name> policy manager.
3. In the **Policy Manager** dialog box, select the bind point with the policy that you want to unbind, for example, Advanced Global.
4. Click the policy name that you want to unbind, and then click Unbind Policy.
5. Click **Apply Changes**.
6. Click **Close**. A message in the status bar indicates that the policy is unbound successfully.

Unbind a DNS policy globally by using the GUI

1. Navigate to **Traffic Management** > **DNS** > **Policies**.
2. In the details pane, click **Global Bindings**.

3. In the **Global Bindings** dialog box, select policy and click **unbind policy**.
4. Click **OK**. A message in the status bar indicates that the policy is unbinded successfully.

Unbind an advanced policy from a load balancing or content switching virtual server by using the GUI

1. Navigate to **Traffic Management,** and expand Load Balancing or Content Switching, and then click **Virtual Servers**.
2. In the details pane, double-click the virtual server from which you want to unbind the policy.
3. On the **Policies** tab, in the **Active** column, clear the check box next to the policy that you want to unbind.
4. Click **OK**. A message in the status bar indicates that the policy is unbinded successfully.

Create policy labels

September 14, 2021

In addition to the built-in bind points where you set up policy banks, you can also configure user-defined policy labels and associate policies with them.

Within a policy label, you bind policies and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. The Citrix ADC also permits you to define an arbitrary evaluation order as follows:

- You can use “goto” expressions to point to the next entry in the bank to be evaluated after the current one.
- You can use an entry in a policy bank to invoke another bank.

Each feature determines the type of policy that you can bind to a policy label, the type of load balancing virtual server that you can bind the label to, and the type of content switching virtual server from which the label can be invoked. For example, a TCP policy label can only be bound to a TCP load balancing virtual server. You cannot bind HTTP policies to a policy label of this type. And you can invoke a TCP policy label only from a TCP content switching virtual server.

After configuring a new policy label, you can invoke it from one or more banks for the built-in bind points.

Create a caching policy label by using the CLI

At the command prompt, type the following commands to create a Caching policy label and verify the configuration:


```
-  add cache policylabel <labelName> -evaluates req|res

-  show cache policylabel<labelName>
<!--NeedCopy-->
```

Example:

```
> add cache policylabel lbl-cache-pol -evaluates req
Done

> show cache policylabel lbl-cache-pol
            Label Name: lbl-cache-pol
            Evaluates: REQ
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

Create a content switching policy label by using the CLI

At the command prompt, type the following commands to create a Content Switching policy label and verify the configuration:

```
-  add cs policylabel <labelName> http|tcp|rtsp|ssl

-  show cs policylabel <labelName>
<!--NeedCopy-->
```

Example:

```
> add cs policylabel lbl-cs-pol http
Done
> show cs policylabel lbl-cs-pol
            Label Name: lbl-cs-pol
            Label Type: HTTP
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```


Create a rewrite policy label by using the CLI

At the command prompt, type the following commands to create a Rewrite policy label and verify the configuration:

```
- add rewrite policylabel <labelName> http_req|http_res|url|text|
   clientless_vpn_req|clientless_vpn_res

- show rewrite policylabel <labelName>
<!--NeedCopy-->
```

Example:

```
> add rewrite policylabel lbl-rewrt-pol http_req
Done

> show rewrite policylabel lbl-rewrt-pol
            Label Name: lbl-rewrt-pol
            Transform Name: http_req
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

Create a responder policy label by using the CLI

At the command prompt, type the following commands to create a Responder policy label and verify the configuration:

```
- add responder policylabel <labelName>

- show responder policylabel <labelName>
<!--NeedCopy-->
```

Example:

```
> add responder policylabel lbl-respndr-pol
Done

> show responder policylabel lbl-respndr-pol
            Label Name: lbl-respndr-pol
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```


Note: Invoke this policy label from a policy bank. For more information, see the "Binding a Policy to a Policy Label" section.

Create a policy label by using the GUI

1. In the navigation pane, expand the feature for which you want to create a policy label, and then click **Policy Labels**. The choices are Integrated Caching, Rewrite, Content Switching, or Responder.
2. In the details pane, click **Add**.
3. In the Name box, enter a unique name for this policy label.
4. Enter feature-specific information for the policy label. For example, for Integrated Caching, in the Evaluates drop-down menu, you would select REQ if you want this policy label to contain request-time policies, or select RES if you want this policy label to contain response-time policies. For Rewrite, you would select a Transform name.
5. Click **Create**.
6. Configure one of the built-in policy banks to invoke this policy label. For more information, see the "Binding a Policy to a Policy Label" section. A message in the status bar indicates that the policy label is created successfully.

Bind a policy to a policy label

As with policy banks that are bound to the built-in bind points, each entry in a policy label is a policy that is bound to the policy label. As with policies that are bound globally or to a vserver, each policy that is bound to the policy label can also invoke a policy bank or a policy label that is evaluated after the current entry has been processed. The following table summarizes the entries in a policy label.

- **Name**. The name of a policy, or, to invoke another policy bank without evaluating a policy, the "dummy" policy name NOPOLICY.

  You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.

- **Priority**. An integer. This setting can work with the Goto expression.

- **Goto Expression**. Determines the next policy to evaluate in this bank. You can provide one of the following values:

  - **NEXT**. Go to the policy with the next higher priority.
  - **END**. Stop evaluation.
  - **USE_INVOCATION_RESULT**. Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT.
  - **Positive number:** The priority number of the next policy to be evaluated.

  - **Numeric expression**. An expression that produces the priority number of the next policy to be evaluated.

  The Goto can only proceed forward in a policy bank.

  If you omit the Goto expression, it is the same as specifying END.

- **Invocation Type**. Designates a policy bank type. The value can be one of the following:
  - **Request Vserver**. Invokes request-time policies that are associated with a virtual server.
  - **Response Vserver**. Invokes response-time policies that are associated with a virtual server.
  - **Policy label**. Invokes another policy bank, as identified by the policy label for the bank.
- **Invocation Name**. The name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

Configure a policy label or virtual server policy bank

September 14, 2021

After you have created policies, and created policy banks by binding the policies, you can perform additional configuration of policies within a label or policy bank. For example, before you configure invocation of an external policy bank, you might want to wait until you have configured that policy bank.

This topic includes the following sections:

- Configure a policy label
- Configure a policy bank for a virtual server

Configure a policy label

A policy label consists of a set of policies and invocations of other policy labels and virtual server-specific policy banks. An Invoke parameter enables you to invoke a policy label or a virtual server-specific policy bank from any other policy bank. A special-purpose NoPolicy entry enables you to invoke an external bank without processing an expression (a rule). The NoPolicy entry is a “dummy” policy that does not contain a rule.

For configuring policy labels from the Citrix ADC command line, note the following elaborations of the command syntax:

- gotoPriorityExpression is configured as described in Table 2. Format of Each Entry in a Policy Bank of the section “Entries in a Policy Bank” in Bind policies using advanced policy.

- The type argument is required. This is unlike binding a conventional policy, where this argument is optional.
- You can invoke the bank of policies that are bound to a virtual server by using the same method as you use for invoking a policy label.

Configure a policy label by using the CLI

At the command prompt, type the following commands to configure a policy label and verify the configuration:

```
1 -  bind cache|rewrite|responder policylabel <policylabelName> -
     policyName <policyName> -priority <priority> [-
     gotoPriorityExpression <gotoprioritыExpression>] [-invoke reqvserver
     |resvserver|policylabel <policyLabelName>|<vserverName>]
2
3 -  show cache|rewrite|responder policylabel <policylabelName>
4 <!--NeedCopy-->
```

Example:

```
1 bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -
     priority 100
2 Done
3 show cache policylabel _reqBuiltinDefaults
4             Label Name: _reqBuiltinDefaults
5             Evaluates: REQ
6             Number of bound policies: 3
7             Number of times invoked: 0
8     1)      Policy Name: _nonGetReq
9             Priority: 100
10            GotoPriorityExpression: END
11    2)      Policy Name: _advancedConditionalReq
12            Priority: 200
13            GotoPriorityExpression: END
14
15    3)      Policy Name: _personalizedReq
16            Priority: 300
17            GotoPriorityExpression: END
18 Done
19 <!--NeedCopy-->
```


Invoke a policy label from a rewrite policy bank with a NOPOLICY entry by using the CLI

At the command prompt, type the following commands to invoke a policy label from a Rewrite policy bank with a NOPOLICY entry and verify the configuration:

```
1 -  bind rewrite global <policyName> <priority> <gotoPriorityExpression>
      -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke
     reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>
2
3 -  show rewrite global
4 <!--NeedCopy-->
```

Example:

```
1 > bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke
     policylabel lbl-rewrt-pol
2 Done
3 > show rewrite global
4      1)      Global bindpoint: REQ_DEFAULT
5              Number of bound policies: 1
6
7      2)      Global bindpoint: REQ_OVERRIDE
8              Number of bound policies: 1
9 Done
10 <!--NeedCopy-->
```

Invoke a policy label from an Integrated Caching policy bank by using the CLI

At the command prompt, type the following commands to invoke a policy label from an Integrated Caching policy bank and verify the configuration:

```
1 -  bind cache global NOPOLICY -priority <priority> -
     gotoPriorityExpression <gotopriorityExpression> -type REQ_OVERRIDE|
     REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|
     policylabel <policyLabelName>|<vserverName>
2
3 -  show cache global
4 <!--NeedCopy-->
```

Example:

```
1 bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -
     type REQ_DEFAULT -invoke policylabel lbl-cache-pol
2 Done
3 > show cache global
```



```
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Invoke a policy label from a Responder policy bank by using the CLI

At the command prompt, type the following commands to invoke a policy label from a Responder policy bank and verify the configuration:

```
-  bind responder global NOPOLICY <priority> <gotopriorityExpression> -
   type OVERRIDE|DEFAULT -invoke vserver|policylabel <policyLabelName
   >|<vserverName>

-  show responder global
<!--NeedCopy-->
```

Example:

```
> bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke
   policylabel lbl-respndr-pol
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

Done
<!--NeedCopy-->
```

Configure a policy label by using the GUI

1. In the navigation pane, expand the feature for which you want to configure a policy label, and then click Policy Labels. The choices are Integrated Caching, Rewrite, or Responder.
2. In the details pane, double-click the label that you want to configure.
3. If you are adding a new policy to this policy label, click Insert Policy, and in the Policy Name field, select New Policy. For more information about adding a policy, see Create or modify a policy. Note that if you are invoking a policy bank, and do not want a rule to be evaluated prior to the invocation, click Insert Policy, and in the Policy Name field select NOPOLICY.

4. For each entry in this policy label, configure the following:

   - **Policy Name:**

     This is already determined by the Policy Name, new policy, or NOPOLICY entry that you inserted in this bank.

   - **Priority:**

     A numeric value that determines either an absolute order of evaluation within the bank, or is used in conjunction with a Goto expression.

   - **Expression:**

     The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see Configure advanced policy expressions: Get started.

   - **Action:**

     The action to be taken if this policy evaluates to TRUE.

   - **Goto Expression:**

     Optional. Used to augment the Priority level to determine the next policy or policy bank to evaluate. For more information on possible values for a Goto expression, see Table 2. Format of Each Entry in a Policy Bank of the section “Entries in a Policy Bank” in Bind policies using advanced policy.

   - **Invoke:**

     Optional. Invokes another policy bank.

5. Click **OK**. A message in the status bar indicates that the policy label is configured successfully.

Configure a policy bank for a virtual server

You can configure a bank of policies for a virtual server. The policy bank can contain individual policies, and each entry in the policy bank can optionally invoke a policy label or a bank of policies that you configured for another virtual server. If you invoke a policy label or policy bank, you can do so without triggering an expression (a rule) by selecting a NOPOLICY “dummy” entry instead of a policy name.

Add policies to a virtual server policy bank by using the CLI

At the command prompt, type the following commands to add policies to a virtual server policy bank and verify the configuration:


```
- bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <
    policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression
    <expression>] [-type REQUEST|RESPONSE]

- show lb|cs vserver <virtualServerName>
<!--NeedCopy-->
```

Example:

```
add lb vserver vs-cont-sw TCP
Done
show lb vserver vs-cont-sw
            vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
            State: DOWN
            Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
            Time since last state change: 0 days, 00:02:14.420
            Effective State: DOWN
            Client Idle Timeout: 9000 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Connection Failover: DISABLED
Done
<!--NeedCopy-->
```

Invoke a policy label from a virtual server policy bank with a NOPOLICY entry by using the CLI

At the command prompt, type the following commands to invoke a policy label from a virtual server policy bank with a NOPOLICY entry and verify the configuration:

```
- bind lb|cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
    NOPOLICY-CACHE|NOPOLICY-RESPONDER -priority <integer> -type REQUEST|
    RESPONSE -gotoPriorityExpression <gotoprioritiyExpression> -invoke
    reqVserver|resVserver|policyLabel <vserverName>|<labelName>

- show lb vserver
<!--NeedCopy-->
```

Example:


```
> bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200
    -type REQUEST -gotoPriorityExpression NEXT -invoke policyLabel lbl-
   rewrt-pol
Done
<!--NeedCopy-->
```

Configure a virtual server policy bank by using the GUI

1. In the left navigation pane, expand **** **Traffic Management** > **Load Balancing, Traffic Management** > **Content Switching, Traffic Management** > **SSL Offload, Security** > **AAA - Application Traffic**, or **Citrix Gateway**, as appropriate, and then click **Virtual Servers.**
2. In the details pane, select the virtual server that you want to configure, and then click **Open**.
3. In the **Configure Virtual Server** dialog box click the **Policies** tab.
4. To create a new policy in this bank, click the icon for the type of policy or policy label that you want to add to the virtual server's bank of policies, click **Insert Policy**. Note that if you want to invoke a policy label without evaluating a policy rule, select the NOPOLICY "dummy" policy.
5. To configure an existing entry in this policy bank, enter the following:
   - **Priority:**

     A numeric value that determines either an absolute order of evaluation within the bank or is used in conjunction with a Goto expression.
   - **Expression:**

     The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see Configuring Advanced Policy Expressions: Getting Started.
   - **Action:**

     The action to be taken if this policy evaluates to TRUE.
   - **Goto Expression:**

     Optional. Determines the next policy or policy bank evaluate. For more information on possible values for a Goto expression, see the section "Entries in a Policy Bank" in Bind policies using advanced policy.
   - **Invoke:**

     Optional. To invoke another policy bank, select the name of the policy label or virtual server policy bank that you want to invoke.
6. Click **OK**. A message in the status bar indicates that the policy is configured successfully.

Invoke or remove a policy label or virtual server policy bank

September 14, 2021

Unlike a policy, which can only be bound once, you can use a policy label or a virtual server's policy bank any number of times by invoking it. Invocation can be performed from two places:

- From the binding for a named policy in a policy bank.
- From the binding for a NOPOLICY "dummy" entry in a policy bank.

Typically, the policy label must be of the same type as the policy from which it is invoked. For example, you would invoke a responder policy label from a responder policy.

Note: When binding or unbinding a global NOPOLICY entry in a policy bank at the command line, you specify a priority to distinguish one NOPOLICY entry from another.

Invoke a rewrite or integrated caching policy label by using the CLI

At the command prompt, type the one of the following commands to invoke a rewrite or integrated caching policy label and verify the configuration:

```
1 -  bind cache global <policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
2
3 -  bind rewrite global<policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
4
5 -  show cache global|show rewrite global
6 <!--NeedCopy-->
```

Example:

```
1 > bind cache global _nonPostReq2 -priority 100 -type req_override -
     invoke
2     policylabel lbl-cache-pol
3 Done
4 > show cache global
5     1)      Global bindpoint: REQ_DEFAULT
6             Number of bound policies: 2
7
8     2)      Global bindpoint: RES_DEFAULT
```



```
            Number of bound policies: 1

     3)     Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Invoke a responder policy label by using the CLI

At the command prompt, type the following commands to invoke a responder policy label and verify the configuration:

```
- bind responder global <policy_Name> <priority_as_positive_integer>
   [<gotoPriorityExpression>] -type REQ_OVERRIDE|REQ_DEFAULT|OVERRIDE|
   DEFAULT -invoke vserver|policylabel <label_name>

- show responder global
<!--NeedCopy-->
```

Example:

```
> bind responder global pol404Error1 300 -invoke policylabel lbl-
   respndr-pol
Done
> show responder global
     1)     Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

Done
<!--NeedCopy-->
```

Invoke a virtual server policy bank by using the CLI

At the command prompt, type the following commands to invoke a Virtual Server Policy Bank and verify the configuration:

```
- bind lb vserver <vserver_name> -policyName <policy_Name> -priority <
   positive_integer> [-gotoPriorityExpression <expression>] -type
   REQUEST|RESPONSE -invoke reqvserver|resvserver|policylabel <
   policy_Label_Name>

- bind lb vserver <vserver_name>
```



```
<!--NeedCopy-->
```

Example:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
Done

> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
            Time since last state change: 28 days, 06:37:49.250
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
        100   Hits: 0

    2)      Policy : pol-ssl Priority:0
    3)      Policy : ns_cmp_msapp Priority:100
    4)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

Remove a rewrite or integrated caching policy label by using the CLI

At the command prompt, type one of the following commands to remove a rewrite or integrated caching policy label and verify the configuration:

```
-  unbind rewrite global <policyName> -priority <positiveInteger> -type
    REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

```



```
-  unbind cache global <policyName> -priority <positiveInteger> -type
   REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  show rewrite global|show cache global
<!--NeedCopy-->
```

Example:

```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
> show rewrite global
Done
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Remove a responder policy label by using the CLI

At the command prompt, type the following commands to remove a responder policy label and verify the configuration:

```
-  unbind responder global <policyName> -priority <positiveInteger> -
   type OVERRIDE|DEFAULT

-  show responder global
<!--NeedCopy-->
```

Example:

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Remove a virtual server policy label by using the CLI

At the command prompt, type one of the following commands to remove a Virtual Server policy label and verify the configuration:


```
-  unbind lb vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>

-  unbind cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>

-  show lb vserver|show cs vserver
<!--NeedCopy-->
```

Example:

```
> unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
Done
> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
            Time since last state change: 28 days, 06:47:54.600
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
        100   Hits: 0

    1)      Policy : pol-ssl Priority:0
    2)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```


Invoke a policy label or virtual server policy bank by using the GUI

1. Bind a policy, as described in Bind a policy globally, Bind a policy to a virtual server, or Bind a policy to a policy label. Alternatively, you can enter a NOPOLICY "dummy" entry instead of a policy name. You do this if you do not want to evaluate a policy before evaluating the policy bank.
2. In the Invoke field, select the name of the policy label or virtual server policy bank that you want to evaluate if traffic matches the bound policy. A message in the status bar indicates that the policy label or virtual server policy bank is invoked successfully.

Remove a policy label invocation by using the GUI

1. Open the policy and clear the Invoke field. Unbinding the policy also removes the invocation of the label. A message in the status bar indicates that the policy label is removed successfully.

Configuring advanced policy expression: getting started

September 14, 2021

Advanced policies evaluate data based on information that you supply in Advanced policy expressions. An Advanced policy expression analyzes data elements (for example, HTTP headers, source IP addresses, the Citrix ADC system time, and POST body data). In addition to configuring an Advanced policy expression in a policy, in some Citrix ADC features, you configure Advanced policy expression outside of the context of a policy.

To create an Advanced policy expression, you select a prefix that identifies a piece of data that you want to analyze, and then you specify an operation to perform on the data. For example, an operation can match a piece of data with a text string that you specify, or it can transform a text string into an HTTP header. Other operations match a returned string with a set of strings or a string pattern. You configure compound expressions by specifying Boolean and arithmetic operators, and by using parentheses to control the order of evaluation.

Advanced policy expression can also contain classic expressions. You can assign a name to a frequently used expression to avoid having to build the expression repeatedly.

Policies and a few other entities include rules that the Citrix ADC uses to evaluate a packet in the traffic flowing through it, to extract data from the Citrix ADC system itself, to send a request (a "callout") to an external application, or to analyze another piece of data. A rule takes the form of a logical expression that is compared against traffic and ultimately returns values of TRUE or FALSE.

The elements of the rule can themselves return TRUE or FALSE, string, or numeric values.

Before configuring an Advanced policy expression, you need to understand the characteristics of the data that the policy or other entity is to evaluate. For example, when working with the Integrated Caching feature, a policy determines what data can be stored in the cache. With Integrated Caching, you need to know the URLs, headers, and other data in the HTTP requests and responses that the Citrix ADC receives. With this knowledge, you can configure policies that match the actual data and enable the Citrix ADC to manage caching for HTTP traffic. This information helps you determine the type of expression that you need to configure in the policy.

Basic elements of an advanced policy expression

September 14, 2021

An Advanced policy expression consists of, at a minimum, a prefix (or a single element used in place of a prefix). Most expressions also specify an operation to be performed on the data that the prefix identifies. You format an expression of up to 1,499 characters as follows:

```
<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]
```

where

- <prefix>

  is an anchor point for starting an expression.

  The prefix is a period-delimited key that identifies a unit of data. For example, the following prefix examines HTTP requests for the presence of a header named Content-Type:

  http.req.header(“Content-Type”)

  Prefixes can also be used on their own to return the value of the object that the prefix identifies.

- <operation>

  identifies an evaluation that is to be performed on the data identified by the prefix.

  For example, consider the following expression:

  http.req.header(“Content-Type”).eq(“text/html”)

  In this expression, the following is the operator component:

  eq(“text/html”)

  This operator causes the Citrix ADC to evaluate any HTTP requests that contain a Content-Type header, and in particular, to determine if the value of this header is equal to the string “text/html.” For more information, see “Operations.”

- <compound-operator>

  is a Boolean or arithmetic operator that forms a compound expression from multiple prefix or prefix.operation elements.

  For example, consider the following expression:

  http.req.header(“Content-Type”).eq(“text/html”) && http.req.url.contains(“.html”)

Prefixes

An expression prefix represents a discrete piece of data. For example, an expression prefix can represent an HTTP URL, an HTTP Cookie header, or a string in the body of an HTTP POST request. An expression prefix can identify and return a wide variety of data types, including the following:

- A client IP address in a TCP/IP packet
- Citrix ADC system time
- An external callout over HTTP
- A TCP or UDP record type

In most cases, an expression prefix begins with one of the following keywords:

- CLIENT:
  - Identifies a characteristic of the client that is either sending a request or receiving a response, as in the following examples:
  - The prefix client.ip.dst designates the destination IP address in the request or response.
  - The prefix client.ip.src designates the source IP address.
- HTTP:
  - Identifies an element in an HTTP request or a response, as in the following examples:
  - The prefix http.req.body(integer) designates the body of the HTTP request as a multiline text object, up to the character position designated in integer.
  - The prefix http.req.header(“header_name”) designates an HTTP header, as specified in header_name.
  - The prefix http.req.url designates an HTTP URL in URL-encoded format.
- SERVER:

  Identifies an element in the server that is either processing a request or sending a response.
- SYS:

  Identifies a characteristic of the Citrix ADC that is processing the traffic.

  Note: Note that DNS policies support only SYS, CLIENT, and SERVER objects.

  In addition, in the Citrix Gateway, the Clientless VPN function can use the following types of prefixes:

- TEXT:

  Identifies any text element in a request or a response.

- TARGET:

  Identifies the target of a connection.

- URL:

  Identifies an element in the URL portion of an HTTP request or response.

As a general rule of thumb, any expression prefix can be a self-contained expression. For example, the following prefix is a complete expression that returns the contents of the HTTP header specified in the string argument (enclosed in quotation marks):

```
http.res.header.("myheader")
```

Or you can combine prefixes with simple operations to determine TRUE and FALSE values. For example, the following returns a value of TRUE or FALSE:

```
http.res.header.("myheader").exists
```

You can also use complex operations on individual prefixes and multiple prefixes within an expression, as in the following example:

```
http.req.url.length + http.req.cookie.length <= 500
```

Which expression prefixes you can specify depends on the Citrix ADC feature. The following table describes the expression prefixes that are of interest on a per-feature basis

Feature	Types of Expression Prefix Used in the Feature
DNS	SYS, CLIENT, SERVER
Responder in Protection Features	HTTP, SYS, CLIENT
Content Switching	HTTP, SYS, CLIENT
Rewrite	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN
Integrated Caching	HTTP, SYS, CLIENT, SERVER
Citrix Gateway, Clientless Access	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN

Table 1. Permitted Types of Expression Prefixes in Various Citrix ADC Features

Note: For details on the permitted expression prefixes in a feature, see the documentation for that feature.

Single-element expressions

The simplest type of Advanced policy expression contains a single element. This element can be one of the following:

- true. An Advanced policy expression can consist simply of the value true. This type of expression always returns a value of TRUE. It is useful for chaining policy actions and triggering Goto expressions.
- false. An Advanced policy expression can consist simply of the value false. This type of expression always returns a value of FALSE.
- A prefix for a compound expression. For example, the prefix HTTP.REQ.HOSTNAME is a complete expression that returns a host name and HTTP.REQ.URL is a complete expression that returns a URL. The prefix could also be used in conjunction with operations and additional prefixes to form a compound expression.

Operations

In most expressions, you also specify an operation on the data that the prefix identifies. For example, suppose that you specify the following prefix:

```
http.req.url
```

This prefix extracts URLs in HTTP requests. This expression prefix does not require any operators to be used in an expression. However, when you configure an expression that processes HTTP request URLs, you can specify operations that analyze particular characteristics of the URL. Following are a few possibilities:

- Search for a particular host name in the URL.
- Search for a particular path in the URL.
- Evaluate the length of the URL.
- Search for a string in the URL that indicates a time stamp and convert it to GMT.

The following is an example of a prefix that identifies an HTTP header named Server and an operation that searches for the string IIS in the header value:

```
http.res.header("Server").contains("IIS")
```

Following is an example of a prefix that identifies host names and an operation that searches for the string “www.mycompany.com” as the value of the name:

```
http.req.hostname.eq("www.mycompany.com")
```

Basic operations on expression prefixes

The following table describes a few of the basic operations that can be performed on expression prefixes.

Operation	Determines Whether or Not
CONTAINS(<string>)	The object matches <string>. Following is an example: http.req.header(“Cache-Control”).contains(“no-cache”)
EXISTS	A particular item is present in an object. Following is an example: http.res.header(“MyHdr”).exists
EQ(<text>)	A particular non-numeric value is present in an object. Following is an example: http.req.method.eq(post)
EQ(<integer>)	A particular numeric value is present in an object. Following is an example: client.ip.dst.eq(10.100.10.100)
LT(<integer>)	An object’s value is less than a particular value. Following is an example: http.req.content_length.lt(5000)
GT(<integer>)	An object’s value is greater than a particular value. Following is an example: http.req.content_length.gt(5)

The following table summarizes a few of the available types of operations.

Operation Type	Description
Text operations	Match individual strings and sets of strings with any portion of a target. The target can be an entire string, the start of a string, or any portion of text in between the start and the end of the string. For example, you can extract the string “XYZ” from “XYZSomeText”. Or, you can compare an HTTP header value with an array of different strings. You can also transform text into another type of data. Following are examples: Transform a string into an integer value, create a list from the query strings in a URL, and transform a string into a time value.

Operation Type	Description
Numeric operations	Numeric operations include applying arithmetic operators, evaluating content length, the number of items in a list, dates, times, and IP addresses.

Compound advanced policy expressions

September 14, 2021

You can configure an Advanced policy expression with boolean or arithmetic operators and atomic operations. The following compound expression has a boolean AND:

```
http.req.hostname.eq("mycompany.com")&& http.req.method.eq(post)
```

The following expression adds the value of two targets, and compares the result to a third value:

```
http.req.url.length + http.req.cookie.length \<= 500
```

A compound expression can have any number of logical and arithmetic operators.

The following expression evaluates the length of an HTTP request. This expression is based on the URL and cookie.
This expression evaluates the text in the header. Also, does a Boolean AND on these two results:

```
http.req.url.length + http.req.cookie.length \<= 500 && http.req.header.
contains("some text")
```

You can use parentheses to control the order of evaluation in a compound expression.

Booleans in compound expressions

You configure compound expressions with the following operators:

- &&.

  This operator is a logical AND. For the expression to evaluate to TRUE, all components must evaluate to TRUE.

  Example:

  http.req.url.hostname.eq(“myHost”) && http.req.header(“myHeader”).exists

- ||.

This operator is a logical OR. If any component of the expression evaluates to TRUE, the entire expression is TRUE.

- !.

  P Does a logical NOT on the expression.

Sometimes, the Citrix ADC configuration utility offers AND, NOT, and OR operators in the **Add Expression** dialog box. However, these compound expressions are of limited use. Citrix recommends that you use the operators &&, ||, and! To configure compound expressions that use Boolean logic.

Parentheses in compound expressions

You can use parentheses to control the order of evaluation of an expression. The following is an example:

```
http.req.url.contains("myCompany.com")|| (http.req.url.hostname.eq("myHost"
)&& http.req.header("myHeader").exists)
```

The following is another example:

```
(http.req.header("Content-Type").exists && http.req.header("Content-Type").
eq("text/html"))|| (http.req.header("Transfer-Encoding").exists || http.req
.header("Content-Length").exists)
```

Compound operations for strings

The following table describes operators that you can use to configure compound operations on string data.

Operations that produce a string value	Description
str + str	Concatenates the value of the expression on the left of the operator with the value on the right. Example: http.req.hostname + http.req.url.protocol
str + num	Concatenates the value of the expression on the left of the operator with a numeric value on the right. Example: http.req.hostname + http.req.url.content_length

Operations that produce a string value	Description
num + str	Concatenates the numeric value of the expression on the left side of the operator with a string value on the right. Example: http.req.url.content_length + http.req.url.hostname
str + ip	Concatenates the string value of the expression on the left side of the operator with an IP address value on the right. Example: http.req.hostname + 10.00.000.00
IP + str	Concatenates the IP address value of the expression on the left of the operator with a string value on the right. Example: client.ip.dst + http.req.url.hostname
str1 ALT str2	Uses string2 if the evaluation of string1 results in an undef exception or the result is a null string. Otherwise uses string1 and never evaluates string2. Example: http.req.hostname alt client.ip.src

Operations on strings that produce a result of TRUE or FALSE	Description
str == str	Evaluates whether the strings on either side of the operator are the same. Following is an example: http.req.header(“myheader”) == http.res.header(“myheader”)
str <= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or precedes it alphabetically.
str >= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or follows it alphabetically.
str < str	Evaluates whether the string on the left side of the operator precedes the string on the right alphabetically.

Operations on strings that produce a result of TRUE or FALSE	Description
str > str	Evaluates whether the string on the left side of the operator follows the string on the right alphabetically.
str !!= str	Evaluates whether the strings on either side of the operator are different.

Logical operations on strings	Description
bool && bool	This operator is a logical AND. When evaluating the components of the compound expression, all components that are joined by the AND must evaluate to TRUE. Following is an example: http.req.method.eq(GET) && http.req.url.query.contains(“viewReport && my_pagelabel”)
bool || bool	This operator is a logical OR. When evaluating the components of the compound expression, if any component of the expression belonging to OR evaluates to TRUE, the entire expression is TRUE. Following is an example: http.req.url.contains(“.js”) || http.res.header.(“Content-Type”). Contains(“javascript”)
bool	Performs a logical NOT on the expression.

Compound operations for numbers

You can configure compound numeric expressions. For example, the following expression returns a numeric value that is the sum of an HTTP header length and a URL length:

```
http.req.header.length + http.req.url.length
```

The following tables describe operators that you can use to configure compound expressions for numeric data.

Arithmetic operations on numbers	Description
num + num	Add the left expression value of the operator to the right expression value. Example: http.req.content_length + http.req.url.length
num – num	Subtract the right expression value of the operator from the left expression value.
num x num	Multiply the left expression value of the operator with the right expression value. Example: client.interface.rxthroughput * 9
num / num	Divide the left expression value of the operator by the right expression value.
num % num	Calculate the modulo, or the numeric remainder on a division of the value of the expression on the left of the operator by the value of the expression on the right. For example, the values “15 mod 4” equal 3, and “12 mod 4” equals 0.
~number	Returns a number after applying a bitwise logical negation of the number. The following example assumes that numeric.expression returns 12 (binary 1100): ~numeric.expression. The result of applying the ~ operator is -11 (a binary 1110011, 32 bits total with all ones to the left). All returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
number ^ number	Compares two bit patterns of equal length and performs an XOR operation on each pair of corresponding bits in each number argument, returning 1 if the bits are different, and 0 if they are the same. Returns a number after applying a bitwise XOR to the integer argument and the current number value. If the values in the bitwise comparison are the same, the returned value is a 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): numeric.expression1 ^ numeric.expression2The result of applying the ^ operator to the entire expression is 6 (binary 0110). All returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
number | number	Returns a number after applying a bitwise OR to the number values. If either value in the bitwise comparison is a 1, the returned value is a 1. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): numeric.expression1 | numeric.expression2 The result of applying the | operator to the entire expression is 14 (binary 1110). All returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
number & number	Compares two bit patterns of equal length and performs a bitwise AND operation on each pair of corresponding bits, returning 1 if both of the bits contain a value of 1, and 0 if either bits are 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): numeric.expression1 & numeric.expression2 The whole expression evaluates to 8 (binary 1000). All returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
num « num	Returns a number after a bitwise left shift of the number value by the right-side number argument number of bits. The number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: numeric.expression1 « numeric.expression2 The result of applying the LSHIFT operator is 96 (a binary 1100000). All returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
num » num	Returns a number after a bitwise right shift of the number value by the integer argument number of bits. The number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: numeric.expression1 » numeric.expression2 The result of applying the RSHIFT operator is 1 (a binary 0001). All returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Numeric operators that produce a result of TRUE or FALSE	Description
num == num	Determine if the value of the expression on the left of the operator is equal to the value of the expression on the right.
num!= num	Determine if the value of the expression on the left of the operator is not equal to the value of the expression on the right.
num > num	Determine if the value of the expression on the left of the operator is greater than the value of the expression on the right.
num < num	Determine if the value of the expression on the left of the operator is less than the value of the expression on the right.

Numeric operators that produce a result of TRUE or FALSE	Description
num >= num	Determine if the value of the expression on the left of the operator is greater than or equal to the value of the expression on the right.
num <= num	Determine if the value of the expression on the left of the operator is less than or equal to the value of the expression on the right

Functions for data types in the policy infrastructure

The Citrix ADC policy infrastructure supports the following numeric data types:

- Integer (32 bits)
- Unsigned long (64 bits)
- Double (64 bits)

Simple expressions can return all of these data types. Also, you can create compound expressions that use arithmetic operators and logical operators to evaluate or return the values of these data types. Also, you can use all of these values in policy expressions. Literal constants of type unsigned long can be specified by appending the string ul to the number. Literal constants of type double contain a period (.), an exponent, or both.

Arithmetic Operators, Logical Operators, and Type Promotion

In compound expressions, the following standard arithmetic and logical operators can be used for the double and unsigned long data types:

- +, -, *, and /

%, ~, ^, &,	, <<, and >> (do not apply to double)

-

- ==,!=, >, <, >=, and <=

All of these operators have the same meaning as in the C programming language.

In all cases of mixed operations between operands of type integer, unsigned long, and double. Type promotion is done to do the operation on the operands of the same type. The operation promotes a lower precedence type to the operand with the highest precedence type. The order of precedence (higher to lower) is as follows:

- Double
- Unsigned long
- Integer

So, an operation that returns a numeric result returns a result of the highest type involved in the operation.

For example, if the operands are of type integer and unsigned long, the integer operand is automatically converted to type unsigned long. This type conversion is done in simple expressions. The type of data identified by the expression prefix does not match the type of data that is passed as the argument to the function. In the operation HTTP.REQ.CONTENT_LENGTH.DIV(3ul), the prefix HTTP.REQ.CONTENT_LENGTH returns an integer that becomes an unsigned long. Unsigned long: the data type passed as the argument to the DIV() function, an unsigned long division is done. Similarly, the argument can be promoted in an expression. For example, HTTP.REQ.HEADER(“myHeader”).TYPECAST_DOUBLE_AT.DIV(5) promotes the integer 5 to type double and does double-precision division.

For information about expressions to cast data of one type to data of another type, see Typecasting data.

Specify the character set in expressions

September 14, 2021

The policy infrastructure on the Citrix ADC appliance supports ASCII and UTF-8 character sets. The default character set is ASCII. If the traffic for which you are configuring an expression consists of only ASCII characters, you need not specify the character set in the expression. The appliance allows all string and character literals which include binary characters. However, the UTF-8 character sets still require the string and character literals to be a valid UTF-8.

```
CLIENT.TCP.PAYLOAD(100).CONTAINS("\xff\x02")
```

In an expression, the SET_CHAR_SET() function must be introduced at the point in the expression after which data processing must be carried out in the specified character set. For example, in the expression HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding example/).CONTAINS_ANY(“Greek_ alphabet”), if the strings stored in the pattern set

“Greek_alphabet” are in UTF-8, you must include the SET_CHAR_SET(UTF_8) function immediately before the CONTAINS_ANY(“<string>”) function, as follows:

```
HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In
the preceding example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("Greek_ alphabet")
```

The SET_CHAR_SET() function sets the character set for all further processing (that is, for all subsequent functions) in the expression unless it is overridden later in the expression by another SET_CHAR_SET() function that changes the character set. Therefore, if all the functions in a given simple expression are intended for UTF-8, you can include the SET_CHAR_SET(UTF_8) function immediately after functions that identify text (for example, the HEADER(“<name>”) or BODY(<int>) functions). In the second example that follows the first paragraph above, if the ASCII arguments passed to the AFTER_REGEX() and BEFORE_REGEX() functions are changed to UTF-8 strings, you can include the SET_CHAR_SET(UTF_8) function immediately after the BODY(1000) function, as follows:

```
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).BEFORE_REGEX
(re/Wörterbuch/).CONTAINS_ANY("Greek_alphabet")
```

The UTF-8 character set is a superset of the ASCII character set, so expressions configured for the ASCII character set continue to work as expected if you change the character set to UTF-8.

Compound expressions with different character sets

In a compound expression, if one subset of expressions is configured to work with data in the ASCII character set and the rest of the expressions are configured to work with data in the UTF-8 character set, the character set specified for each individual expression is considered when the expressions are evaluated individually. However, when processing the compound expression, just before processing the operators, the appliance promotes the character set of the returned ASCII values to UTF-8. For example, in the following compound expression, the first simple expression evaluates data in the ASCII character set while the second simple expression evaluates data in the UTF-8 character set:

```
HTTP.REQ.HEADER("MyHeader")== HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

However, when processing the compound expression, just before evaluating the “is equal to” Boolean operator, the Citrix ADC appliance promotes the character set of the value returned by HTTP.REQ.HEADER(“MyHeader”) to UTF-8.

The first simple expression in the following example evaluates data in the ASCII character set. However, when the Citrix ADC appliance processes the compound expression, just before concatenating the results of the two simple expressions, the appliance promotes the character set of the value returned by HTTP.REQ.BODY(10) to UTF-8.

```
HTTP.REQ.BODY(10)+ HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

Therefore, the compound expression returns data in the UTF-8 character set.

Specify the character set based on the character set of traffic

You can set the character set to UTF-8 based on traffic characteristics. If you are not sure whether the character set of the traffic being evaluated is UTF-8, you can configure a compound expression in which the first expression checks for UTF-8 traffic and subsequent expressions set the character set to UTF-8. Following is an example of a compound expression that first checks the value of "charset" in the request's Content-Type header for "UTF-8" before checking whether the first 1000 bytes in the request contain the UTF-8 string Bücher:

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T
('=', '; ', '"').VALUE("charset").EQ("UTF-8")&& HTTP.REQ.BODY(1000).SET_CHAR_SET
(UTF_8).CONTAINS("Bücher")
```

If you are sure that the character set of the traffic being evaluated is UTF-8, the second expression in the example is sufficient.

Character and string literals in expressions

During expression evaluation, even if the current character set is ASCII, character literals and string literals, which are enclosed in single quotation marks (') and quotation marks (""), respectively, are considered to be literals in the UTF-8 character set. In a given expression, if a function is operating on character or string literals in the ASCII character set and you include a non-ASCII character in the literal, an error is returned.

> **Note**:
>
> The string literals in advanced policy expressions are now as long as the policy expression. The expression is allowed to be 1499 bytes or 8191 bytes long.

Values in hexadecimal and octal formats

When configuring an expression, you can enter values in octal and hexadecimal formats. However, each hexadecimal or octal byte is considered a UTF-8 byte. Invalid UTF-8 bytes result in errors regardless of whether the value is entered manually or pasted from the clipboard. For example, "\xce\x20" is an invalid UTF-8 character because "c8" cannot be followed by "20" (each byte in a multi-byte UTF-8 string must have the high bit set). Another example of an invalid UTF-8 character is "\xce \xa9," since the hexadecimal characters are separated by a white-space character.

Functions that return UTF-8 strings

Only the `text>.XPATH` and `<text>.XPATH_JSON` functions always return UTF-8 strings. The following MYSQL routines determine at runtime which character set to return, depending on the data in the protocol:

- MYSQL_CLIENT_T.USER
- MYSQL_CLIENT_T.DATABASE
- MYSQL_REQ_QUERY_T.COMMAND
- MYSQL_REQ_QUERY_T.TEXT
- MYSQL_REQ_QUERY_T.TEXT(<unsigned **int**>)
- MYSQL_RES_ERROR_T.SQLSTATE
- MYSQL_RES_ERROR_T.MESSAGE
- MYSQL_RES_FIELD_T.CATALOG
- MYSQL_RES_FIELD_T.DB
- MYSQL_RES_FIELD_T.TABLE
- MSQL_RES_FIELD_T.ORIGINAL_TABLE
- MYSQL_RES_FIELD_T.NAME
- MYSQL_RES_FIELD_T.ORIGINAL_NAME
- MYSQL_RES_OK_T.MESSAGE
- MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned **int**>)

Terminal connection settings for UTF-8

When you set up a connection to the Citrix ADC appliance by using a terminal connection (by using PuTTY, for example), you must set the character set for transmission of data to UTF-8.

Minimum and maximum functions in an advanced policy expression

The advanced policy expressions support the below minimum and maximum functions.

1. (<expression1>.max(<expression2>) - returns the maximum of the two values.
2. (<expression1>.min(<expression2>) - returns the minimum of the two values.

Classic expressions in advanced policy expressions

September 14, 2021

> **Warning:**
>
> Classic policy expressions are no longer supported from Citrix ADC 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Configure advanced policy expressions: Get started.

Classic expressions describe the basic characteristics of traffic. Sometimes, you might want to use a classic expression in an advanced policy expression.

The following is the syntax for all Advanced policy expressions that use a classic expression:

SYS.EVAL_CLASSIC_EXPR(“expression”)

> **Note:**
>
> The syntax and the metadata for the SYS.EVAL_CLASSIC_EXPR expression is getting deprecated. You can manually convert or use the nspepi tool to convert the Classic expression to the advanced expression.

Following are examples of the SYS.EVAL_CLASSIC_EXPR(“expression”) expression:

```
1 sys.eval_classic_expr("req.ssl.client.cipher.bits > 1000")
2 sys.eval_classic_expr("url contains abc")
3 sys.eval_classic_expr("req.ip.sourceip == 10.102.1.61 -netmask
     255.255.255.255")
4 sys.eval_classic_expr("time >= *:30:00GMT")
5 sys.eval_classic_expr("e1 || e2")
6 sys.eval_classic_expr("req.http.urllen > 50")
7 sys.eval_classic_expr("dayofweek == wedGMT")
8 <!--NeedCopy-->
```

> **Note:**
>
> When you upgrade the Citrix ADC to version 9.0 or higher, Integrated Caching policies are automatically upgraded to advanced policies, and the expressions in these policies are upgraded to the advanced policies.

Configure advanced policy expressions in a policy

September 14, 2021

You can configure an Advanced policy expression of up to 1,499 characters in a policy. The user interface for Advanced policy expressions depends to some extent on the feature for which you are configuring the expression, and on whether you are configuring an expression for a policy or for another use.

When configuring expressions on the command line, you delimit the expression by using quotation marks (“. . .” or ‘. . .’). Within an expression, you escape additional quotation marks by using a back-slash (). For example, the following are standard methods for escaping quotation marks in an expression:

```
"\"abc\""
```

```
 '\"abc\"'
```


You must also use a backslash to escape question marks and other backslashes on the command line. For example, the expression http.req.url.contains(“\?”) requires a backslash so that the question mark is parsed. Note that the backslash character will not appear on the command line after you type the question mark. On the other hand, if you escape a backslash (for example, in the expression ‘http.req.url.contains(“\\http”)’), the escape characters are echoed on the command line.

To make an entry more readable, you can escape the quotation marks for an entire expression. At the start of the expression you enter the escape sequence “q” plus one of the following special characters: /{<	~$^+=&%@‘?.

You enter only the special character at the end of the expression, as follows:

```
q@http.req.url.contains("sometext") && http.req.cookie.exists@

q~http.req.url.contains("sometext") && http.req.cookie.exists~
<!--NeedCopy-->
```

Note that an expression that uses the { delimiter is closed with }.

For some features (for example, Integrated Caching and Responder), the policy configuration dialog box provides a secondary dialog box for configuring expressions. This dialog enables you to choose from drop-down lists that show the available choices at each point during expression configuration. You cannot use arithmetic operators when using these configuration dialogs, but most other advanced policy expression features are available. To use arithmetic operators , write your expressions in free-form format.

Configure an Advanced policy syntax rule by using the CLI

> **Note:**
>
> Default syntax policy is now renamed as Advanced policy.

At the command prompt, type the following commands to configure a default syntax rule and verify the configuration:

1. `add cache|dns|rewrite|cs policy policyName **-rule** expression featureSpecificPa **-action**`
2. `show cache|dns|rewrite|cs policy policyName`
   Following is an example of configuring a caching policy:

   Example:


```
> add cache policy pol-cache -rule http.req.content_length.le(5) -
   action INVAL
 Done

> show cache policy pol-cache
        Name: pol-cache
        Rule: http.req.content_length.le(5)
        CacheAction: INVAL
        Invalidate groups: DEFAULT
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

Configure a default syntax policy expression by using the GUI

1. In the navigation pane, click the name of the feature where you want to configure a policy, for example, you can select Integrated Caching, Responder, DNS, Rewrite, or Content Switching, and then click **Policies**.
2. Click Add.
3. For most features, click in the **Expression** field. For content switching, click **Configure**.
4. Click the **Prefix** icon (the house) and select the first expression prefix from the drop-down list. For example, in Responder, the options are HTTP, SYS, and CLIENT. The next set of applicable options appear in a drop-down list.
5. Double-click the next option to select it, and then type a period (.). Again, a set of applicable options appears in another drop-down list.
6. Continue selecting options until an entry field (signalled by parentheses) appears. When you see an entry field, enter an appropriate value in the parentheses. For example, if you select GT(int) (greater-than, integer format), you specify an integer in the parentheses. Text strings are delimited by quotation marks. Following is an example:

   ```
   HTTP.REQ.BODY(1000).BETWEEN("this","that")
   ```


To insert an operator between two parts of a compound expression, click the Operators icon (the sigma), and select the operator type. Following is an example of a configured expression with a Boolean OR (signalled by double vertical bars,	):

7. `HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this","that")`
8. To insert a named expression, click the down arrow next to the Add icon (the plus sign) and select a named expression.
9. To configure an expression using drop-down menus, and to insert built-in expressions, click the Add icon (the plus sign). The **Add Expression** dialog box works in a similar way to the main dialog box, but it provides drop-down lists for selecting options, and it provides text fields for data entry instead of parentheses. This dialog box also provides a Frequently Used Expressions drop-down list that inserts commonly used expressions. When you are done adding the expression, click **OK**.
10. When finished, click **Create**. A message in the status bar indicates that the policy expression is configured successfully.

Test a default syntax expression by using the GUI

1. In the navigation pane, click the name of the feature for which you want to configure a policy (for example, you can select Integrated Caching, Responder, DNS, Rewrite, or Content Switching), and then click Policies.
2. Select a policy and click **Open**.
3. To test the expression, click the Evaluate icon (the check mark).
4. In the expression evaluator dialog box, select the Flow Type that matches the expression.
5. In the **HTTP Request Data** or **HTTP Response Data** field, paste the HTTP request or response that you want to parse with the expression, and click **Evaluate**. Note that you must supply a complete HTTP request or response, and the header and body should be separated by blank line. Some programs that trap HTTP headers do not also trap the response. If you are copying and pasting only the header, insert a blank line at the end of the header to form a complete HTTP request or response.

6. Click **Close** to close this dialog box.

Configure named advanced policy expressions

September 14, 2021

Instead of retyping the same expression multiple times in multiple policies, you can configure a named expression and refer to the name any time you want to use the expression in a policy. For example, you could create the following named expressions:

- ThisExpression:

  ```
  http.req.body(100).contains("this")
  ```

- ThatExpression:

  ```
  http.req.body(100).contains("that")
  ```

You can then use these named expressions in a policy expression. For example, the following is a legal expression based on the preceding examples:

ThisExpression	ThatExpression

You can use the name of an advanced policy expression as the prefix to a function. The named expression can be either a simple expression or a compound expression. The function must be one that can operate on the type of data that is returned by the named expression.

Example 1: Simple Named Expression as a Prefix

The following simple named expression, which identifies a text string, can be used as a prefix to the AFTER_STR(“<string>“)function, which works with text data:

```
HTTP.REQ.BODY(1000)
```

If the name of the expression is top1KB, you can use top1KB.AFTER_STR(“username”) instead of HTTP.REQ.BODY(1000).AFTER_STR(“username”).

Example 2: Compound named expression as a prefix

You can create a compound named expression called basic_header_value to concatenate the user name in a request, a colon (:), and the user’s password, as follows:

```
add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\" + HTTP
.REQ.USER.PASSWD"
```


You can then use the name of the expression in a rewrite action, as shown in the following example:

```
add rewrite action insert_b64encoded_authorization insert_http_header
authorization '"Basic " + basic_header_value.b64encode'-bypassSafetyCheck
 YES
```

In the example, in the expression that is used to construct the value of the custom header, the B64 encoding algorithm is applied to the string returned by the compound named expression.

You can also use a named expression (either by itself or as a prefix to a function) to create the text expression for the replacement target in a rewrite.

Configure a named default syntax expression by using the CLI

At the command prompt, type the following commands to configure a named expression and verify the configuration:

```
1 -  add policy expression \<name\>\<value\>
2
3 -  show policy expression \<name\>
4 <!--NeedCopy-->
```

Example:

```
1 > add policy expression myExp "http.req.body(100).contains(\"the other
     \")"
2 Done
3
4 > show policy expression myExp
5     1)      Name: myExp  Expr: "http.req.body(100).contains("the other"
         )"  Hits: 0 Type : ADVANCED
6 Done
7 <!--NeedCopy-->
```

The expression can be up to 1,499 characters.

Configure a named expression by using the GUI

1. In the navigation pane, expand **AppExpert**, and then click **Expressions**.
2. Click **Advanced Expressions**.
3. Click **Add**.
4. Enter a name and a description for the expression.
5. Configure the expression by using the process described in Configure advanced policy expression. A message in the status bar indicates that the policy expression is configured successfully.

Configure advanced policy expressions outside the context of a policy

September 14, 2021

A number of functions, including the following, can require an advanced policy expression that is not part of a policy:

- Integrated Caching selectors:

  You define multiple non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND relationship with the others.

- Load Balancing:

  You configure an expression for the TOKEN method of load balancing for a load balancing virtual server.

- Rewrite actions:

  Expressions define the location of the rewrite action and the type of rewriting to be performed, depending on the type of rewrite action that you are configuring. For example, a DELETE action only uses a target expression. A REPLACE action uses a target expression and an expression to configure the replacement text.

- Rate-based policies:

  You use advanced policy expressions to configure Limit Selectors. You can use these selectors when configuring policies to throttle the rate of traffic to various servers. You define up to five non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND with the others.

Configure an advanced policy expression outside a policy by using the CLI (cache selector example)

At the command prompt, type the following commands to configure an advanced policy expression outside a policy and verify the configuration:

```
1 -  add cache selector <selectorName> <rule>
2 -  show cache selector <selectorName>
3 <!--NeedCopy-->
```

Example:

```
1 > add cache selector mainpageSelector "http.req.cookie.value("ABC_def")
     "
2     "http.req.url.query.value("_ghi")"selector "mainpageSelector" added
```



```
Done
> show cache selector mainpageSelector
            Name: mainpageSelector
            Expressions:
                    1) http.req.cookie.value("ABC_def")
                    2) http.req.url.query.value("_ghi")
Done
<!--NeedCopy-->
```

Following is an equivalent command that uses the more readable q delimiter, as described in Configure advanced policy expressions in a policy:

```
> add cache selector mainpageSelector2 q~http.req.cookie.value("ABC_def
   ")~
    q~http.req.url.query.value("_ghi")~selector "mainpageSelector2"
       added
Done
> show cache selector mainpageSelector2
            Name: mainpageSelector2
            Expressions:
                    1) http.req.cookie.value("ABC_def")
                    2) http.req.url.query.value("_ghi")
Done
<!--NeedCopy-->
```

Advanced policy expressions: evaluating text

September 14, 2021

You can configure a policy with an advanced policy expression that evaluates text in a request or response. Advanced policy text expressions can range from simple expressions that perform string matching in HTTP headers to complex expressions that encode and decode text. You can configure text expressions to be case sensitive or case insensitive and to use or ignore spaces. You can also configure complex text expressions by combining text expressions with Boolean operators

You can use expression prefixes and operators for evaluating HTTP requests, HTTP responses, and VPN and Clientless VPN data. However, text expression prefixes are not restricted to evaluating these elements of your traffic.

About text expressions

September 14, 2021

You can configure various expressions for working with text that flows through the Citrix ADC appliance. Following are some examples of how you can parse text by using a default syntax expression:

- Determine that a particular HTTP header exists.

  For example, you may want to identify HTTP requests that contains a particular Accept-Language header for the purpose of directing the request to a particular server.

- Determine that a particular HTTP URL contains a particular string.

  For example, you may want to block requests for particular URLs. Note that the string can occur at the beginning, middle, or end of another string.

- Identify a POST request that is directed to a particular application.

  For example, you may want to identify all POST requests that are directed to a database application for the purpose of refreshing cached application data.

Note that there are specialized tools for viewing the data stream for HTTP requests and responses. You can use the tools to view the data stream.

About operations on text

A text-based expression consists of at least one prefix to identify an element of data and usually (although not always) an operation on that prefix. Text-based operations can apply to any part of a request or a response. Basic operations on text include various types of string matches.

For example, the following expression compares a header value with a string:

```
http.req.header("myHeader").contains("some-text")
```

Following expressions are examples of matching a file type in a request:

```
http.req.url.suffix.contains("jpeg")
```

```
http.req.url.suffix.eq("jpeg")
```

In the preceding examples, the contains operator permits a partial match and the eq operator looks for an exact match.

Other operations are available to format the string before evaluating it. For example, you can use text operations to strip out quotes and white spaces, to convert the string to all lowercase, or to concatenate strings.

> **Note**:
>
> Complex operations are available to perform matching based on patterns or to convert one type of text format to another type.

For more information, see the following topics:

- Pattern sets and data sets.
- Regular expressions.
- Typecasting data.

Compounding and precedence in text expressions

You can apply various operators to combine text prefixes or expressions. For example, the following expression concatenates the returned values of each prefix:

```
http.req.hostname + http.req.url
```

Following is an example of a compound text expression that uses a logical AND. Both components of this expression must be TRUE for a request to match the expression:

```
http.req.method.eq(post)&& http.req.body(1024).startswith("destination=")
```

> **Note:**
>
> For more information on operators for compounding, see Compound advanced expressions.

Categories of text expressions

The primary categories of text expressions that you can configure are:

- Information in HTTP headers, HTTP URLs, and the POST body in HTTP requests.

  For more information, see Expression prefixes for text in HTTP requests and responses.

- Information regarding a VPN or a clientless VPN.

  For more information, see Expression prefixes for VPNs and clientless VPNs.

- TCP payload information.

  For more information about TCP payload expressions, see Advanced policy expressions: Parsing HTTP, TCP, and UDP data.

- Text in a Secure Sockets Layer (SSL) certificate.

  For information about text expressions for SSL and SSL certificate data, see Advanced policy expressions: Parsing SSL certificates and Expressions for SSL certificate dates.

> **Note:**
>
> Parsing a document body, such as the body of a POST request, can affect performance. You may want to test the performance impact of policies that evaluate a document body.

Guidelines for text expressions

From a performance standpoint, it typically is best to use protocol-aware functions in an expression. For example, the following expression makes use of a protocol-aware function:

```
HTTP.REQ.URL.QUERY
```

The previous expression performs better than the following equivalent expression, which is based on string parsing:

```
HTTP.REQ.URL.AFTER_STR("?")
```

In the first case, the expression looks specifically at the URL query. In the second case, the expression scans the data for the first occurrence of a question mark.

There is also a performance benefit from structured parsing of text, as in the following expression:

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(For more information on typecasting, see Typecasting data. The typecasting expression, which collects comma-delimited data and structures it into a list, typically would perform better than the following unstructured equivalent:

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

Finally, unstructured text expressions typically have better performance than regular expressions. For example, the following is an unstructured text expression:

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```

The previous expression would generally provide better performance than the following equivalent, which uses a regular expression:

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

For more information on regular expressions, see Regular expressions.

Expression prefixes for text in HTTP requests and responses

September 14, 2021

An HTTP request or response typically contains text, such as in the form of headers, header values, URLs, and POST body text. You can configure expressions to operate on one or more of these text-based items in an HTTP request or response.

Refer to the Expression Prefix table for information on how to configure and extract text from different parts of an HTTP request or response.

Expression prefixes for VPNs and clientless VPNs

September 14, 2021

The Advanced policy engine provides prefixes that are specific to parsing VPN or Clientless VPN data. This data includes the following:

- Host names, domains, and URLs in VPN traffic.
- Protocols in the VPN traffic.
- Queries in the VPN traffic.

These text elements are often URLs and components of URLs. In addition to applying the text-based operations on these elements, you can parse these elements by using operations that are specific to parsing URLs. For more information, see Expressions for extracting segments of URLs

For information about VPN expression prefixes, see VPN expression table.

Basic operations on text

September 14, 2021

Basic operations on text include operations for string matching, calculating the length of a string, and controlling case sensitivity. You can include white space in a string that is passed as an argument to an expression, but the string cannot exceed 255 characters.

String comparison functions

The following table lists basic string matching operations in which the functions return a Boolean TRUE or FALSE.

Function	Description
`<text>.CONTAINS(<string>)`	Returns a Boolean TRUE value if the target contains `<string>`. Example: `http.req.url.contains(".jpeg")`
`<text>.EQ(<string>)`	Returns a Boolean TRUE value if the target is an exact match with `<string>`. For example, the following expression returns a Boolean TRUE for a URL with a host name of "myhostabc": `http.req.url.hostname.eq("myhostabc")`
`<text>.STARTSWITH(<string>)`	Returns a Boolean TRUE value if the target begins with `<string>`. For example, the following expression returns a Boolean TRUE for a URL with a host name of "myhostabc": `http.req.url.hostname.startswith("myhost")`
`<text>.ENDSWITH(<string>)`	Returns a Boolean TRUE value if the target ends with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of "myhostabc": `http.req.url.hostname.endswith("abc")`
`<text>.NE(<string>)`	Returns a Boolean TRUE value if the prefix is not equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with `SET_TEXT_MODE(IGNORECASE)` or `SET_TEXT_MODE(NOIGNORECASE)`, and with both ASCII and UTF-8 character sets.

Function	Description
<text>.GT(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically greater than the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.GE(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically greater than or equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.LT(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically lesser than the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.

Function	Description
`<text>.LE(<string>)`	Returns a Boolean TRUE value if the prefix is alphabetically lesser than or equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with `SET_TEXT_MODE(IGNORECASE)` or `SET_TEXT_MODE(NOIGNORECASE)`, and with both ASCII and UTF-8 character sets.

Calculate the length of a string

The `<text>.LENGTH` operation returns a numeric value that is equal to the number of characters (not bytes) in a string:

```
<text>.LENGTH
```

For example, you may want to identify request URLs that exceed a particular length. Following is an expression that implements this example:

```
HTTP.REQ.URL.LENGTH < 500
```

After taking a count of the characters or elements in a string, you can apply numeric operations to them. For more information, see Default Syntax Expressions: Working with Dates, Times, and Numbers.

Consider, ignore, and change text case

The following functions operate on the case (upper-case or lower-case) of the characters in the string.

Function	Description	
`<text>.SET_TEXT_MODE` (IGNORECASE	NOIGNORECASE)	This function turns case sensitivity on or off for all text operations.

Function	Description
<text>.TO_LOWER	Converts the target to lowercase for a text block of up to 2 kilobyte (KB). Returns UNDEF if the target exceeds 2 KB. For example, the string “ABCd:” is converted to “abcd:”.
<text>.TO_UPPER	Converts the target to uppercase. Returns UNDEF if the target exceeds 2 KB. For example, the string “abcD:” is converted to “ABCD:”.

Strip specific characters from a string

You can use the STRIP_CHARS(<string>) function to remove specific characters from the text that is returned by a default syntax expression prefix (the input string). All instances of the characters that you specify in the argument are stripped from the input string. You can use any text method on the resulting string, including the methods used for matching the string with a pattern set.

For example, in the expression CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(“.-_”), the STRIP_CHARS(<string>) function strips all periods (.), hyphens (-), and underscores (_) from the domain name returned by the prefix CLIENT.UDP.DNS.DOMAIN. If the domain name that is returned is “a.dom_ai_n-name”, the function returns the string “adomainname”.

In the following example, the resulting string is compared with a pattern set called “listofdomains”:

```
CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")
```

Note: You cannot perform a rewrite on the string that is returned by the STRIP_CHARS(<string>) function.

The following functions strip matching characters from the beginning and end of a given string input.

Function	Description
<text>.STRIP_START_CHARS(s)	Strips matching characters from the beginning of the input string until the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is TestLang and :/*en_us:is its value,HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:“)* the specified characters from the beginning of the value of the header until the first non-matching character e is found and returnsen_us: as a string.
<text>.STRIP_END_CHARS(s)	Strips matching characters from the end of the input string to the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is TestLang and :/*en_us:is its value,HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:“)* the specified characters from the end of the value of the header until the first non-matching character s is found and returns:/_en_us as a string.

Append a string to another string

You can use the APPEND() function to append the string representation of the argument to the string representation of the value returned by the preceding function. The preceding function can be one that returns a number, unsigned long, double, time value, IPv4 address, or IPv6 address. The argument can be a text string, number, unsigned long, double, time value, IPv4 address, or IPv6 address. The resulting string value is the same string value that is obtained by using the + operator.

Complex operations on text

September 14, 2021

In addition to simple string matching, you can configure expressions that examine string length and text block for patterns rather than specific strings.

Be aware of the following for any text-based operation:

- For any operation that takes a string argument, the string cannot exceed 255 characters.
- You can include white space when you specify a string in an expression.

Operations on the length of a string

The following operations extract strings by a character count.

Character Count Operation	Description
<text>.TRUNCATE(<count>)	Returns a string after truncating the end of the target by the number of characters in <count>. If the entire string is shorter than <count>, nothing is returned.
<text>.TRUNCATE(<character>, <count>)	Returns a string after truncating the text after <character> by the number of characters specified in <count>.
<text>.PREFIX(<character>, <count>)	Selects the longest prefix in the target that has at most <count> occurrences of <character>.
<text>.SUFFIX(<character>, <count>)	Selects the longest suffix in the target that has at most <count> occurrences of <character>. For example, consider the following response body: JLEwx. The following expression returns a value of “JLEwx”: http.res.body(100).suffix(‘L’,1) The following expression returns “LLEwx”: http.res.body(100).suffix(‘L’,2)

Character Count Operation	Description
<text>.SUBSTR(<starting_offset>, <length>)	Select a string with <length>number of characters from the target object. Begin extracting the string after the <starting_offset>. If the number of characters after the offset are fewer than the value of the <length> argument, select all the remaining characters.
<text>.SKIP(<character>, <count>)	Select a string from the target after skipping over the longest prefix that has at most <count> occurrences of <character>.

Operations on a portion of a string

Refer to the String operations table to know how to extract a subset of a larger string by using one of the operations.

Operations for comparing the alphanumeric order of two strings

The COMPARE operation examines the first nonmatching character of two different strings. This operation is based on lexicographic order, which is the method used when ordering terms in dictionaries.

This operation returns the arithmetic difference between the ASCII values of the first nonmatching characters in the compared strings. The following differences are examples:

- The difference between “abc” and “and” is -1 (based on the third pair-wise character comparison).
- The difference between “@” and “abc” is -33.
- The difference between “1” and “abc” is -47.

Following is the syntax for the COMPARE operation.

```
<text>.COMPARE(<string>)
```

Extract an integer from a string of bytes that represent text

Refer to the Integer extraction table to know how to treat a string of bytes that represent text as a sequence of bytes, extract 8 bits, 16 bits, or 32 bits from the sequence, and then convert the extracted bits to an integer.

Convert text to a hash value

You can convert a text string to a hash value by using the HASH function. This function returns a 31-bit positive integer as a result of the operation. Following is the format of the expression:

```
<text>.HASH
```

This function ignores case and white spaces. For example, after the operation, the two strings Ab c and a bc would produce the same hash value.

Encode and decode text by applying the Base64 encoding algorithm

The following two functions encode and decode a text string by applying the Base64 encoding algorithm

Function	Description
text.B64ENCODE	Encodes the text string (designated by text) by applying the Base64 encoding algorithm.
text.B64DECODE	Decodes the Base64-encoded string (designated by text) by applying the Base64 decoding algorithm. The operation raises an UNDEF if text is not in B64-encoded format.

Refine the search in a rewrite action by using the EXTEND function

The EXTEND function is used in rewrite actions that specify patterns or pattern sets and target the bodies of HTTP packets. When a pattern match is found, the EXTEND function extends the scope of the search by a predefined number of bytes on both sides of the matching string. A regular expression can then be used to perform a rewrite on matches in this extended region. Rewrite actions that are configured with the EXTEND function perform rewrites faster than rewrite actions that evaluate entire HTTP bodies using only regular expressions.

The format of the EXTEND function is EXTEND(m,n), where m and n are the number of bytes by which the scope of the search is extended before and after the matching pattern, respectively. When a match is found, the new search scope comprises m bytes that immediately precede the matching string, the string itself, and the n bytes that follow the string. A regular expression can then be used to perform a rewrite on a portion of this new string.

The EXTEND function can be used only if the rewrite action in which it is used fulfills the following requirements:

- The search is performed by using patterns or patterns sets (not regular expressions)

- The rewrite action evaluates only the bodies of HTTP packets.

Also, the EXTEND function can be used only with the following types of rewrite actions:

- replace_all
- insert_after_all
- delete_all
- insert_before_all

For example, you might want to delete all instances of "http://exampleurl.com/" and "http://exampleurl.au/" in the first 1000 bytes of the body. To do this, you can configure a rewrite action to search for all instances of the string exampleurl, extend the scope of the search on both sides of the string when a match is found, and then use a regular expression to perform the rewrite in the extended region. The following example extends the scope of the search by 20 bytes to the left and 50 bytes to the right of the matching string:

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)'-pattern
exampleurl -refineSearch 'extend(20,50).regex_select(re##http://exampleurl
.(com|au)##)'
```

Convert text to hexadecimal format

The following function converts text to hexadecimal format and extracts the resulting string:

```
<text>.BLOB_TO_HEX(<string>)
```

For example, this function converts the byte string "abc" to "61:62:63".

Encrypt and decrypt text

In default syntax expressions, you can use the ENCRYPT and DECRYPT functions to encrypt and decrypt text. Data encrypted by the ENCRYPT function on a given Citrix ADC appliance or high availability (HA) pair is intended for decryption by the DECRYPT function on the same Citrix ADC appliance or HA pair. The appliance supports the RC4, DES3, AES128, AES192, and AES256 encryption methods. The key value that is required for encryption is not user-specifiable. When an encryption method is set, the appliance automatically generates a random key value that is appropriate for the specified method. The default method is AES256 encryption, which is the most secure encryption method and the one that Citrix recommends.

You do not need to configure encryption unless you want to change the encryption method or you want the appliance to generate a new key value for the current encryption method.

Note: You can also encrypt and decrypt XML payloads. For information about the functions for encrypting and decrypting XML payloads, see Encrypt and decrypt XML payloads.

Configure encryption

During startup, the appliance runs the set ns encryptionParams command with, by default, the AES256 encryption method, and uses a randomly generated key value that is appropriate for AES256 encryption. The appliance also encrypts the key value and saves the command, with the encrypted key value, to the Citrix ADC configuration file. Therefore, the AES256 encryption method is enabled for the ENCRYPT and DECRYPT functions by default. The key value that is saved in the configuration file persists across reboots even though the appliance runs the command each time you restart it.

You can run the set ns encryptionParams command manually, or use the configuration utility, if you want to change the encryption method or if you want the appliance to generate a new key value for the current encryption method. To use the CLI to change the encryption method, set only the method parameter, as shown in "**Example 1: Changing the Encryption Method**." If you want the appliance to generate a new key value for the current encryption method, set the method parameter to the current encryption method and the keyValue parameter to an empty string (""), as shown in "**Example 2: Generating a New Key Value for the Current Encryption Method**." After you generate a new key value, you must save the configuration. If you do not save the configuration, the appliance uses the newly generated key value only until the next restart, after which it reverts to the key value in the saved configuration.

Configure encryption by using the GUI

1. Navigate to **System** > **Settings**.
2. In the **Settings** area, click **Change Encryption** parameters.
3. In the **Change Encryption Parameters** dialog box, do one of the following:
   - To change the encryption method, in the Method list, select the encryption method that you want.
   - To generate a new key value for the current encryption method, click Generate a new key for the selected method.
4. Click **OK**.

Use the ENCRYPT and DECRYPT functions

You can use the ENCRYPT and DECRYPT functions with any expression prefix that returns text. For example, you can use the ENCRYPT and DECRYPT functions in rewrite policies for cookie encryption. In the following example, the rewrite actions encrypt a cookie named MyCookie, which is set by a back-end service, and decrypt the same cookie when it is returned by a client:

```
1 add rewrite action my-cookie-encrypt-action replace "HTTP.RES.
     SET_COOKIE.COOKIE("MyCookie").VALUE(0)" "HTTP.RES.SET_COOKIE.COOKIE(
     "MyCookie").VALUE(0).ENCRYPT" -bypassSafetyCheck YES
2
```



```
3 add rewrite action my-cookie-decrypt-action replace "HTTP.REQ.COOKIE.
    VALUE("MyCookie")" "HTTP.REQ.COOKIE.VALUE("MyCookie").DECRYPT" -
    bypassSafetyCheck YES
4 <!--NeedCopy-->
```

After you configure policies for encryption and decryption, save the configuration to bring the policies into effect.

Configure encryption key for third-party encryption

In default syntax expressions, you can use ENCRYPT and DECRYPT functions for encrypting and decrypting text in a request or response. The data encrypted by the ENCRYPT function on an appliance (standalone, high availability, or cluster) is intended to be decrypted by the DECRYPT function by the same appliance. The appliance supports RC4, DES, Triple-DES, AES92, and AES256 encryption methods and each of these methods use a secret key for both encryption and decryption of data. You can use the any of these methods to encrypt and decrypt data in two ways - self-encryption and third-party encryption.

The self-encryption feature in an appliance (standalone, high availability or cluster) encrypts and then decrypts data by evaluating the header value. One example to understand this is the HTTP Cookie encryption. The expression evaluates the header, encrypts the HTTP cookie value in the Set-Cookie header in the outgoing response and then decrypts the cookie value when it is returned in the cookie header of a subsequent incoming request from the client. The key value is not user configurable, instead when an encryption method is configured in the set ns encryptionParams command, the appliance automatically generates a random key value for the configured method. By default, the command uses the AES256 encryption method, which is the highly secured method and Citrix recommends this method.

The third-party encryption feature encrypts or decrypts data with a third-party application. For example, a client may encrypt data in a request and the appliance decrypts the data before sending it the back-end server or vice versa. To perform this, the appliance and the third party application must share a secret key. On the appliance, you can directly configure the secret key using an encryption key object and key value is automatically generated by the appliance for a stronger encryption. The same key is manually configured on the third-party appliance so that both appliance and third-party application can use the same key for encrypting and decryption data.

Note: Using third-party encryption, you can also encrypt and decrypt XML payloads. For information about the functions for encrypting and decrypting XML payloads, see “Encrypting and Decrypting XML Payloads.

Cipher methods

A cipher method provides two functions: an encryption function that transforms a plaintext byte sequence into a ciphertext byte sequence, and a decryption function that transforms the ciphertext back to the plaintext. Cipher methods use byte sequences called keys to perform encryption and decryption. Cipher methods that use the same key for encryption and decryption are called symmetric. Cipher methods that use different keys for encryption and decryption are asymmetric. The most notable examples of asymmetric ciphers are in public key cryptography, which uses a public key available to anyone for encryption and a private key known only to the decrypter.

A good cipher method makes it infeasible to decrypt ("crack") ciphertext if you don't possess the key. "Infeasible" really means that cracking the cyphertext would take more time and computing resources than it is worth. As computers become more powerful and cheaper, ciphers that were formerly infeasible to crack become more feasible. Also, over time, flaws are found in cipher methods (or their implementations), making cracking easier. Newer cipher methods are therefore preferred over older ones. In general, longer length keys provide better security than shorter keys, at the cost of longer encryption and decryption times.

A cipher method can use stream ciphers or block ciphers. RC4 is the mostly secured stream ciphers and it is used only for legacy application. Block ciphers can include padding.

Stream ciphers

A stream cipher method operates on individual bytes. Only one stream cipher is available on Citrix ADC- appliances: RC4, which uses a 128 bit (16 bytes) key length. For a given key, RC4 generates a pseudo-random sequence of bytes, call a keystream, which is X-ORed with the plaintext to produce the ciphertext. RC4 is no longer considered secure and should be used only if required by legacy applications.

Block ciphers

A block cipher method operates on a fixed block of bytes. A Citrix ADC appliance provides two block ciphers: Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). DES uses a block size of 8 bytes and (on a Citrix ADC appliance) two choices for key length: 64 bits (8 bytes), of which 56 bits are data and 8 bits are parity, and Triple-DES, a 192 bit (24 bytes) key length. AES has a block size of 16 bytes and (on Citrix ADC) three choices for key length: 128 bits (16 bytes), 192 bits (24 bytes) and 256 bits (32 bytes).

Padding

If the plaintext for a block cipher is not an integral number of blocks, padding with more bytes might be required. For example, suppose the plaintext is "xyzzy" (hex 78797a7a79). For an 8-byte Triple-

DES block, this value would have to be padded to create 8 bytes. The padding scheme must allow the decryption function to determine the length of the original plaintext after decryption. Following are some padding schemes currently in use (n is the number of bytes added):

- PKCS7: Adds n bytes of value n each. For example, 78797a7a79030303. This is the padding scheme used by OpenSSL and ENCRYPT() policy function. The PKCS5 padding scheme is the same as PKCS7.
- ANSI X.923: Adds n-1 zero bytes and a final byte of value n. For example, 78797a7a79000003.
- ISO 10126: Adds n-1 random bytes and a final byte of value n. For example, 78797a7a79xxxx03, where xx can be any byte value. The DECRYPT() policy function accepts this padding scheme, which also allows it to accept the PKCS7 and ANSI X.923 schemes.
- ISO/IEC 7816-4: Adds a 0x80 byte and n-1 zero bytes. For example, 78797a7a79800000. This is also call OneAndZeros padding.
- Zero: Adds n zero bytes. Example: 78797a7a79000000. This can only be used with plaintext that does not include NUL bytes.

If padding is used and the plaintext is an integral number of blocks, an extra block is typically added so that the decryption function can unambiguously determine the original plaintext length. For PCKS7 and 8 byte block, this would be 0808080808080808.

Modes of operation

There are a number of different modes of operation for block ciphers, which specify how multiple blocks of plaintext are encrypted. Some modes use an initialization vector (IV), a block of data apart from the plaintext that is used to start the encryption process. It is a good practice to use a different IV for each encryption, so that the same plaintext produces different ciphertext. The IV does not need to be secret, and so is prepended to the ciphertext. Modes include:

- Electronic Codebook (ECB): Each block of plaintext is encrypted independently. An IV is not used. Padding is required if the plaintext is not a multiple of the cipher block size. The same plaintext and key always produces the same ciphertext. Because of this, ECB is considered less secure than other modes and should only be used for legacy applications.
- Cipher Block Chaining (CBC): Each block of plaintext is XORed with the previous ciphertext block, or the IV for the first block, before being encrypted. Padding is required if the plaintext is not a multiple of the cipher block size. This is the mode used with the Citrix ADC encryptionParams method.
- Cipher Feedback (CFB): The previous ciphertext block, or the IV for the first block, is encrypted and the output is XORed with the current plaintext block to create the current ciphertext block. The feedback can be 1 bit, 8 bits, or 128 bits. Since the plaintext is XORed with the cipher text, padding is not required.
- Output Feedback (OFB): A keystream is generated by applying the cipher successively to the IV and XORing the keystream blocks with the plaintext. Padding is not required.

Configure encryption keys for third-party encryption

Following are the configuration tasks performed in configuring encryption key.

1. Adding an encryption key. Configures an encryption key for a specified cipher method with a specified key value.
2. Modifying an encryption key. You can edit parameters for a configured encryption key.
3. Unsetting an encryption key. Sets parameters for a configured encryption key to their default values. An encryptionKey value with the name must exist. Sets padding to DEFAULT (determined by the method), Deletes an existing IV, which causes ENCRYPT() to generate a random IV. Deletes an existing comment. The method and key value cannot be reset.
4. Removing an encryption key. Deletes a configured encryption key. The key cannot have any references.
5. Show an encryption key. Displays parameters for the configured encryption key or all configured keys. If the name is omitted, the key value is not displayed.

Add an encryption key by using the CLI

At the command prompt, type:

```
add ns encryptionKey <name> -method <method> [-keyValue <keyvalue>][-
padding (OFF | ON)] [-iv <hexstring>] -keyValue <keyvalue> [-comment <
string>]
```

Where,

```
1 <method> = ( NONE | RC4 | DES3 | AES128 | AES192 | AES256 | DES | DES-
     CBC | DES-CFB | DES-OFB | DES-ECB | DES3-CBC | DES3-CFB | DES3-OFB |
      DES3-ECB | AES128-CBC | AES128-CFB | AES128-OFB | AES128-ECB |
     AES192-CBC | AES192-CFB | AES192-OFB | AES192-ECB | AES256-CBC |
     AES256-CFB | AES256-OFB | AES256-ECB ) <hexstring> = hex-encoded
     byte sequence
2 <!--NeedCopy-->
```

The above encryption methods specify the operation mode with CBC as the default mode of operation. Therefore, DES, DES2, AES128, AES192, and AES256 methods are equivalent to DES-CBC, DES3-CBC, AES128-CBC, AES192-CBC, and AES256-CBC methods.

Modify an encryption key by using the CLI

At the command prompt, type:

```
set ns encryptionKey <name> [-method <method>] [-keyValue <keyvalue>] [-
padding ( OFF | ON )] [-iv <string>] [-comment <string>]
```


Unset an encryption key by using the CLI

At the command prompt, type:

```
unset ns encryptionKey <name> [-padding] [-iv] [-comment]
```

Remove an encryption key by using the CLI

At the command prompt, type:

```
rm ns encryptionKey <name>
```

Show an encryption key by using the CLI

At the command prompt, type:

Example:

```
show ns encryptionKey [<name>]

add ns encryptionKey my_key -method aes256 -keyValue 26
    ea5537b7e0746089476e5658f9327c0b10c3b4778c673a5b38cee182874711  - iv
    c2bf0b2e15c15004d6b14bcdc7e5e365
set ns encryptionKey my_key -keyValue
    b8742b163abcf62d639837bbee3cef9fb5842d82d00dfe6548831d2bd1d93476
unset ns encryptionKey my_key -iv
rm ns encryptionKey my_key
show ns encryptionKey my_key
Name: my_key
Method: AES256
Padding: DEFAULT
Key Value: (not disclosed)
<!--NeedCopy-->
```

Add an encryption key by using the GUI

Navigate to **System** > **Encryption Keys** and click **Add** to create an encryption key.

Modify an encryption key by using the GUI

Navigate to **System** > **Encryption Keys** and click **Edit** to modify parameters for a configured encryption key.

Remove an encryption key by using the GUI

Navigate to **System** > **Encryption keys** and click **Delete**.

ENCRYPT and DECRYPT functions for third-party encryption

Following is the ENCRYPT function used for third-part encryption.

```
ENCRYPT (encryptionKey, out_encoding)
```

Where,

Input data for the appliance is the text to be encrypted

encryptionKey: An optional string parameter that specifies the configured encryption key object to provide the encryption method, secret key value and other encryption parameters. If omitted, the method uses the automatically generated key value associated with the set ns encryptionParamS command.

out_encoding: This value specifies how the output is encoded. If omitted, BASE64 encoding is used.

Input:

```
BASE64: original PEM base64-encoding: 6 bits (0..63) encoded as one
    ASCII character:
                0..23 = 'A'..'Z', 24..51 = 'a'..'z', 52..61 = '0'..'9
                    ', 62 = '+', 63 = '/', '=' = pad byte.
       BASE64URL:  URL and Filename safe base64-encoding: same as BASE64
            except 62 = '-', 63 = '_'
       HEX_UPPER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '.
       HEX_LOWER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'a'..'f
           '.
       HEX_COLONS: Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '; ':' between each hex byte. Matches BLOB_TO_HEX() output
           format
       HEX:  For input, accepts HEX_UPPER, HEX_LOWER, and HEX_COLONS
           format. For output, produces HEX_LOWER format
<!--NeedCopy-->
```

Output: The output is a text encrypted using the specified method and key and encoded using a specified output encoding. It inserts a generated IV before the encrypted text for block methods and modes that require an IV, and either no IV is specified for the encryptionKey or the encryptionKey is omitted.

Following is the DECRYPT function used for third-part decryption.

```
DECRYPT(encryptionKey, in_encoding)
```

Where,

Input data is an encrypted text using the specified method and key encoded using the specified input encoding. This text is expected to include a generated IV before the encrypted text for block methods

and modes that require an IV, and either no IV is specified for the encryptionKey or the encryptionKey is omitted.

encryptionKey—An optional string parameter that specifies the configured encryptionKey object to provide the encryption method, secret key and other encryption parameters. If omitted, the method and automatically generated key associated with the encryptionParams setting will be used

in_encoding—An optional enumeration parameter that specifies how the input is expected to be encoded. The values are the same as the out_encoding of ENCRYPT. If omitted, BASE64 encoding will be expected.

The output data is an unencoded decrypted text.

Variants and optional parameters

Following are the variants of these functions with the optional parameters:

Variant	Description
ENCRYPT	Use encryptionParams command and BASE64 output encoding parameter.
ENCRYPT(out_encoding)	Use encryptionParams and specified output encoding parameter.
ENCRYPT(encryptionKey)	Use the specified encryptionKey and BASE64 output encoding parameter.
ENCRYPT(encryptionKey, out_encoding)	Use the specified encryptionKey and output encoding parameter.
DECRYPT	Use encryptionParams command and BASE64 input encoding parameter.
DECRYPT(out_encoding)	Use encryptionParams command and the specified input encoding parameter.
DECRYPT(encryptionKey)	Use the specified encryptionKey and BASE64 input encoding parameter.
DECRYPT(encryptionKey, out_encoding)	Use the specified encryptionKey and input encoding parameter.

Configure HMAC keys

Citrix ADC appliances support a Hashed Message Authentication Code (HMAC) function that calculates a digest method or hash of input text by using a secret key shared between a message sender

and message receiver. The digest method (derived from an RFC 2104 technique) authenticates the sender and verifies that the message content has not been altered. For example, when a client sends a message with the shared HMAC key to a Citrix ADC appliance, advanced (PI) policy expressions use the HMAC function to compute the hash-based code on the selected text. Then, when the receiver receives the message with the secret key, it recomputes the HMAC by comparing it with the original HMAC to determine whether the message has been altered. The HMAC function is supported by standalone appliances and by appliances in a high availability configuration or in a cluster. Using it is similar to configuring an encryption key.

The add ns hmackey and set ns hmackey commands include a parameter that specifies the digest method and the shared secret key to use for the HMAC computation.

To configure a HMAC key, you must perform the following:

1. Adding an HMAC key. Configures an HMAC key with a specified key value.
2. Modifying an HMAC key. Modifies parameters for a configured HMAC key. The digest method can be changed without changing the key value, since the key value length is not determined by the digest. However, it is advisable to specify a new key when changing the digest.
3. Unsetting an HMAC key. Sets parameters for a configured HMAC key to their default values. An hmacKey object with the name must exist. The only parameter that can be unset is the comment, which is deleted.
4. Removing an HMAC key. Deletes a configured key. The key cannot have any references.
5. Show an HMAC key. Displays parameters for the configured HMAC ac key or all configured keys. If the name is omitted, the key value is not displayed.

Configure a unique and random HMAC key

You can automatically generate an unique HMAC key. If you appliance is a cluster configuration, the HMAC key is generated at the start of the process and distributed to all nodes and packet engines. This ensures the HMAC key is same for all packet engines and all nodes in the cluster.

At the command prompt, type:

```
add ns hmacKey <your_key> -digest <digest> -keyValue <keyvalue>
```

Example:

```
add ns hmacKey <name> -digest sha1 -keyValue AUTO
```

Where,

- Name syntax is correct and does not duplicate the name of an existing key.
- The “AUTO” keyValue can be used in the set commands to generate new keys for existing encrytionKey and hmacKey objects.

> **Note:**
>
> The automatic key generation is useful if the Citrix ADC appliance is encrypting and decrypting data with the key, or generating and verifying an HMAC key. Since the key value itself is already encrypted when displayed, you cannot retrieve the generated key value for use by any other party.

Example:

```
add ns hmacKey my_hmac_key -digest sha1 -keyValue 0c753c6c5ef859189cacdf95b506d02c179
```

The above encryption methods specify the operation mode with CBC as the default mode of operation. Therefore, DES, DES2, AES128, AES192, and AES256 methods are equivalent to DES-CBC, DES3-CBC, AES128-CBC, AES192-CBC, and AES256-CBC methods.

Modify an HMAC key by using the CLI

This command modifies the parameters configured for an HMAC key. You can change the digest without changing the key value, since the key value length is not determined by the digest. However, it is advisable to specify a new key when changing the digest. At the command prompt, type:

```
set ns hmacKey <name> [-digst <digest>] [-keyValue <keyvalue>]
[-comment <string>]

<!--NeedCopy-->
```

Unset an HMAC key by using the CLI

This command sets parameters configured for an HMAC key with their default values. An hmacKey object with the name must exist. The only parameter that you can unset is the comment option, which is deleted. At the command prompt, type:

```
unset ns hmacKey <name> -comment
```

Remove an HMAC key by using the CLI

This command deletes the configured hmac key. The key cannot have references. At the command prompt, type:

```
rm ns hmacKey <name>
```


Show an HMAC key by using the CLI

At the command prompt, type:

```
show ns encryptionKey [<name>]

add ns hmacKey my_hmac_key -digest sha1 -keyValue 0
    c753c6c5ef859189cacdf95b506d02c1797407d
set ns hmacKey my_hmac_key -keyValue
    f348c594341a840a1f641a1cf24aa24c15eb1317
rm ns hmacKey my_hmac_key
show ns hmacKey my_hmac_key
          Name: my_hmac_key
     Digest: SHA1
     Key Value: (not disclosed)
<!--NeedCopy-->
```

Advanced policy expressions: working with dates, times, and numbers

September 14, 2021

Most numeric data that the Citrix ADC appliance processes consists of dates and times. In addition to working with dates and times, the appliance processes other numeric data, such as the lengths of HTTP requests and responses. To process this data, you can configure advanced policy expressions that process numbers.

A numeric expression consists of an expression prefix that returns a number and sometimes, but not always, an operator that can perform an operation on the number. Examples of expression prefixes that return numbers are `SYS.TIME.DAY`, `HTTP.REQ.CONTENT_LENGTH`, and `HTTP.RES.BODY.LENGTH`. `Numeric` operators can work with any prefix expression that returns data in numeric format. The `GT(<int>)` operator, for example, can be used with any prefix expression, such as HTTP.REQ.CONTENT_LENGTH, that returns an integer.

Format of dates and times in an expression

September 14, 2021

When configuring an advanced policy expression in a policy that works with dates and times (for example, the Citrix ADC system time or a date in an SSL certificate), you specify a time format as follows:

`GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]`

Where:

- <yyyy> is a four-digit year after GMT or LOCAL.
- <month> is a three-character abbreviation for the month, for example, Jan, Dec.
- <d> is a day of the week or an integer for the date.

  You cannot specify the day as Monday, Tuesday, and so on. You specify either an integer for a specific day of the month, or you specify a date as the first, second, third weekday of the month, and so on. Following are examples of specifying a day of the week:

  - Sun_1 is the first Sunday of the month.
  - Sun_3 is the third Sunday of the month.
  - Wed_3 is the third Wednesday of the month.
  - 30 is an example of an exact date in a month.
- <h> is the hour, for example, 10h.
- <s> is the number of seconds, for example, 30s.

The following example expression is true if the date is between 2008 Jan and 2009 Jan, based on GMT.

```
http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)
```

The following example expression is true for March and all months that follow March in the calendar year, based on GMT:

```
sys.time.ge(GMT 2008 Mar)
```

When you specify a date and time, note that the format is case sensitive and must preserve the exact number of blank spaces between entries.

```
**Note:**

In an expression that requires two time values, both must use GMT or
    both must use LOCAL. You cannot mix the two in an expression.

Unlike when you use the SYS.TIME prefix in an advanced policy
    expression, if you specify SYS.TIME in a rewrite action, the Citrix
    ADC returns a string in conventional date format (for example, Sun,
    06 Nov 1994 08:49:37 GMT). For example, the following rewrite action
     replaces the http.res.date header with the Citrix ADC system time
    in a conventional date format:

add rewrite action sync_date replace http.res.date sys.time
```


Expressions for the Citrix ADC system time

September 14, 2021

The SYS.TIME expression prefix extracts the Citrix ADC system time. You can configure expressions that establish whether a particular event occurred at a particular time or within a particular time range according to the Citrix ADC system time.

The following table describes the expressions that you can create by using the SYS.TIME prefix.

- **SYS.TIME.BETWEEN(<time1>, <time2>):**

  Returns a Boolean TRUE if the returned value is later than <time1> and earlier than <time2>.

  You format the <time1>, <time2> arguments as follows:

  - They must both be GMT or both LOCAL.
  - <time2> must be later than <time1>.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following:

  - sys.time.between(GMT 2004, GMT 2006)
  - sys.time.between(GMT 2004 Jan, GMT 2006 Nov)
  - sys.time.between(GMT 2004 Jan, GMT 2006)
  - sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3)
  - sys.time.between(GMT 2005 May 1, GMT May 2005 1)
  - sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)

- **SYS.TIME.DAY:**

  Returns the current day of the month as a number from 1 through 31.

- **SYS.TIME.EQ(<time>):**

  Returns a Boolean TRUE if the current time is equal to the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.eq(GMT 2005) (TRUE in this example.)
  - sys.time.eq(GMT 2005 Dec) (FALSE in this example.)
  - sys.time.eq(LOCAL 2005 May) (Evaluates to TRUE or FALSE in this example, depending on the current time zone.)
  - sys.time.eq(GMT 10h) (TRUE in this example.)
  - sys.time.eq(GMT 10h 30s) (TRUE in this example.)
  - sys.time.eq(GMT May 10h) (TRUE in this example.)
  - sys.time.eq(GMT Sun) (TRUE in this example.)

  - sys.time.eq(GMT May Sun_1) (TRUE in this example.)

- **SYS.TIME.NE(<time>):**

  Returns a Boolean TRUE if the current time is not equal to the <time> argument.

- **SYS.TIME.GE(<time>):**

  Returns a Boolean TRUE if the current time is later than or equal to <time>.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.ge(GMT 2004) (TRUE in this example.)
  - sys.time.ge(GMT 2005 Jan) (TRUE in this example.)
  - sys.time.ge(LOCAL 2005 May) (TRUE or FALSE in this example, depending on the current time zone.)
  - sys.time.ge(GMT 8h) (TRUE in this example.)
  - sys.time.ge(GMT 30m) (FALSE in this example.)
  - sys.time.ge(GMT May 10h) (TRUE in this example.)
  - sys.time.ge(GMT May 10h 0m) (TRUE in this example.)
  - sys.time.ge(GMT Sun) (TRUE in this example.)
  - sys.time.ge(GMT May Sun_1) (TRUE in this example.)

- **SYS.TIME.GT(<time>):**

  Returns a Boolean TRUE if the time value is later than the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.gt(GMT 2004) (TRUE in this example.)
  - sys.time.gt(GMT 2005 Jan) (TRUE in this example.)
  - sys.time.gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - sys.time.gt(GMT 8h) (TRUE in this example.)
  - sys.time.gt(GMT 30m) (FALSE in this example.)
  - sys.time.gt(GMT May 10h) (FALSE in this example.)
  - sys.time.gt(GMT May 10h 0m) (TRUE in this example.)
  - sys.time.gt(GMT Sun) (FALSE in this example.)
  - sys.time.gt(GMT May Sun_1) (FALSE in this example.)

- **SYS.TIME.HOURS:**

  Returns the current hour as an integer from 0 to 23.

- **SYS.TIME.LE(<time>):**

  Returns a Boolean TRUE if the current time value precedes or is equal to the <time> argument.

For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

- sys.time.le(GMT 2006) (TRUE in this example.)
- sys.time.le(GMT 2005 Dec) (TRUE in this example.)
- sys.time.le(LOCAL 2005 May) (TRUE or FALSE depending on the current timezone.)
- sys.time.le(GMT 8h) (FALSE in this example.)
- sys.time.le(GMT 30m) (TRUE in this example.)
- sys.time.le(GMT May 10h) (TRUE in this example.)
- sys.time.le(GMT Jun 11h) (TRUE in this example.)
- sys.time.le(GMT Wed) (TRUE in this example.)
- sys.time.le(GMT May Sun_1) (TRUE in this example.)

- **SYS.TIME.LT(<time>):**

  Returns a Boolean TRUE if the current time value precedes the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.lt(GMT 2006) (TRUE in this example.)
  - sys.time.lt.time.lt(GMT 2005 Dec) (TRUE in this example.)
  - sys.time.lt(LOCAL 2005 May) (TRUE or FALSE depending on the current time zone.)
  - sys.time.lt(GMT 8h) (FALSE in this example.)
  - sys.time.lt(GMT 30m) (TRUE in this example.)
  - sys.time.lt(GMT May 10h) (FALSE in this example.)
  - sys.time.lt(GMT Jun 11h) (TRUE in this example.)
  - sys.time.lt(GMT Wed) (TRUE in this example.)
  - sys.time.lt(GMT May Sun_1) (FALSE in this example.)

- **SYS.TIME.MINUTES:**

  Returns the current minute as an integer from 0 to 59.

- **SYS.TIME.MONTH:**

  Extracts the current month and returns an integer from 1 (January) to 12 (December).

- **SYS.TIME.RELATIVE_BOOT:**

  Calculates the number of seconds to the closest previous or scheduled reboot, and returns an integer.

  If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.

- **SYS.TIME.RELATIVE_NOW:**

Calculates the number of seconds between the current Citrix ADC system time and the specified time, and returns an integer showing the difference.

If the designated time is in the past, the integer is negative; if it is in the future, the integer is positive.

- **SYS.TIME.SECONDS:**

  Extracts the seconds from the current Citrix ADC system time, and returns that value as an integer from 0 to 59.

- **SYS.TIME.WEEKDAY:**

  Returns the current weekday as a value from 0 (Sunday) to 6 (Saturday).

- **SYS.TIME.WITHIN (<time1>, <time2>):**

  If you omit an element of time in <time1>, for example, the day or hour, it is assumed to have the lowest value in its range. If you omit an element in <time2>, it is assumed to have the highest value of its range.

  The ranges for the elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. If you specify the year, you must do so in both <time1> and <time2>.

  For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.within(GMT 2004, GMT 2006) (TRUE in this example.)
  - sys.time.within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.)
  - sys.time.within(GMT Feb, GMT) (TRUE, May is in the range of February to December.)
  - sys.time.within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.)
  - sys.time.within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE in this example.)
  - sys.time.within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the Citrix ADC system time zone.)

- **SYS.TIME.YEAR:**

  Extracts the year from the current system time and returns that value as a four-digit integer.

Expressions for SSL certificate dates

September 14, 2021

You can determine the validity period for SSL certificates by configuring an expression that contains the following prefix:

```
CLIENT.SSL.CLIENT_CERT
```

The following example expression matches a particular time for expiration with the information in the certificate:

```
client.ssl.client_cert.valid_not_after.eq(GMT 2009)
```

The following table describes time-based operations on SSL certificates. To obtain the expression you want, replace *certificate* in the expression in the first column with the prefix expression, "CLIENT.SSL.CLIENT_CERT".

- `<certificate>.VALID_NOT_AFTER`:

  Returns the last day before certificate expiration. The return format is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).

- **<certificate>.VALID_NOT_AFTER.BETWEEN(<time1>, <time2>):**

  Returns a Boolean TRUE value if the certificate validity is between the <time1> and <time2> arguments. Both <time1> and <time2> must be fully specified. Following are examples:

  GMT 1995 Jan is fully specified.

  GMT Jan is not fully specified

  GMT 1995 20 is not fully specified.

  GMT Jan Mon_2 is not fully specified.

  The <time1> and <time2> arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>.

  For example, if it is GMT 2005 May 1 10h 15m 30s, and the first Sunday of the month, you can specify the following (evaluation results are in parentheses).

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the Citrix ADC system time zone.)

- **<certificate>.VALID_NOT_AFTER.DAY:**

  Extracts the last day of the month that the certificate is valid, and returns a number from 1 through 31, as appropriate for the date.

- **<certificate>.VALID_NOT_AFTER.EQ(<time>):**

  Returns a Boolean TRUE if the time is equal to the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results for this example are in parentheses):

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone)
  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)
  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GE(<time>):**

  Returns a Boolean TRUE if the time value is greater than or equal to the argument <time>.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GT(<time>):**

  Returns a Boolean TRUE if the time value is greater than the argument <time>.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . . .gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)

  - . . .gt(GMT May 10h) (FALSE)
  - . . .gt(GMT Sun) (FALSE)
  - . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.HOURS:**

  Extracts the last hour that the certificate is valid and returns that value as an integer from 0 to 23.

- **<certificate>.VALID_NOT_AFTER.LE(<time>):**

  Returns a Boolean TRUE if the time precedes or is equal to the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .le(GMT 8h) (FALSE)
  - . . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.LT(<time>):**

  Returns a Boolean TRUE if the time precedes the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following:

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.MINUTES:**

  Extracts the last minute that the certificate is valid and returns that value as an integer from 0 to 59.

- **<certificate>.VALID_NOT_AFTER.MONTH:**

  Extracts the last month that the certificate is valid and returns that value as an integer from 1 (January) to 12 (December).

- **<certificate>.VALID_NOT_AFTER.RELATIVE_BOOT:**

  Calculates the number of seconds to the closest previous or scheduled reboot and returns an integer. If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_AFTER.RELATIVE_NOW;**

  Calculates the number of seconds between the current system time and the specified time and returns an integer. If the time is in the past, the integer is negative; if it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_AFTER.SECONDS:**

  Extracts the last second that the certificate is valid and returns that value as an integer from 0 to 59.

- **<certificate>.VALID_NOT_AFTER.WEEKDAY:**

  Extracts the last weekday that the certificate is valid. Returns a number between 0 (Sunday) and 6 (Saturday) to give the weekday in the time value.

- **<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>):**

  Returns a Boolean TRUE if the time lies within all the ranges defined by the elements in <time1> and <time2>.

  If you omit an element of time from <time1>, it is assumed to have the lowest value in its range. If you omit an element from <time2>, it is assumed to have the highest value of its range. If you specify a year in <time1>, you must specify it in <time2>.

  The ranges for elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. For the result to be TRUE, each element in the time must exist in the corresponding range that you specify in <time1>, <time2>.

  For example, if time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are in parentheses):

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.)
  - . . .within(GMT Feb, GMT) (TRUE, May is in the range for February to December)
  - . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday lies within the range of the first Sunday through the third Sunday)

  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the Citrix ADC system time zone)

- **<certificate>.VALID_NOT_AFTER.YEAR:**

  Extracts the last year that the certificate is valid and returns a four-digit integer.

- **<certificate>.VALID_NOT_BEFORE:**

  Returns the date that the client certificate becomes valid.

  The return format is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).

- **<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>):**

  Returns a Boolean TRUE if the time value is between the two time arguments. Both <time1> and <time2> arguments must be fully specified.

  Following are examples:

  GMT 1995 Jan is fully specified.
  GMT Jan is not fully specified.
  GMT 1995 20 is not fully specified.
  GMT Jan Mon_2 is not fully specified.
  The time arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the Citrix ADC system time zone.)

- **<certificate>.VALID_NOT_BEFORE.DAY:**

  Extracts the last day of the month that the certificate is valid and returns that value as a number from 1 through 31 representing that day.

- **<certificate>.VALID_NOT_BEFORE.EQ(<time>):**

  Returns a Boolean TRUE if the time is equal to the <time> argument.

For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

- . . .eq(GMT 2005) (TRUE)
- . . .eq(GMT 2005 Dec) (FALSE)
- . . .eq(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
- . . .eq(GMT 10h) (TRUE)
- . . .eq(GMT 10h 30s) (TRUE)
- . . .eq(GMT May 10h) (TRUE)
- . . .eq(GMT Sun) (TRUE)
- . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GE(<time>):**

  Returns a Boolean TRUE if the time is greater than (after) or equal to the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results are in parentheses):

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GT(<time>):**

  Returns a Boolean TRUE if the time occurs after the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results are in parentheses):

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . . .gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)
  - . . .gt(GMT May 10h) (FALSE)
  - . . .gt(GMT May 10h 0m) (TRUE)
  - . . .gt(GMT Sun) (FALSE)

- . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.HOURS:**

  Extracts the last hour that the certificate is valid and returns that value as an integer from 0 to 23.

- **<certificate>.VALID_NOT_BEFORE.LE(<time>)

  Returns a Boolean TRUE if the time precedes or is equal to the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May) (TRUE or FALSE,depending on the current time zone.)
  - . . .le(GMT 8h) (FALSE)
    . - . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.LT(<time>):**

  Returns a Boolean TRUE if the time precedes the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.MINUTES:**

  Extracts the last minute that the certificate is valid. Returns the current minute as an integer from 0 to 59.

- **<certificate>.VALID_NOT_BEFORE.MONTH:**

  Extracts the last month that the certificate is valid. Returns the current month as an integer from 1 (January) to 12 (December).

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_BOOT:**

  Calculates the number of seconds to the closest previous or scheduled Citrix ADC reboot and returns an integer. If the closest boot time is in the past, the integer is negative; if it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_NOW:**

  Returns the number of seconds between the current Citrix ADC system time and the specified time as an integer. If the designated time is in the past, the integer is negative. If it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_BEFORE.SECONDS:**

  Extracts the last second that the certificate is valid. Returns the current second as an integer from 0 to 59.

- **<certificate>.VALID_NOT_BEFORE.WEEKDAY:**

  Extracts the last weekday that the certificate is valid. Returns the weekday as a number between 0 (Sunday) and 6 (Saturday).

- **<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>, <time2>):**

  Returns a Boolean TRUE if each element of time exists within the range defined in the <time1>, <time2> arguments.

  If you omit an element of time from <time1>, it is assumed to have the lowest value in its range. If you omit an element of time from <time2>, it is assumed to have the highest value in its range. If you specify a year in <time1>, it must be specified in <time2>. The ranges for elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59.

  For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are in parentheses):

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.)
  - . . .within(GMT Feb, GMT) (TRUE, May is in the range of February to December.)
  - . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.)
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the Citrix ADC system time zone)

- **<certificate>.VALID_NOT_BEFORE.YEAR:**

  Extracts the last year that the certificate is valid. Returns the current year as a four-digit integer.

Expressions for HTTP request and response dates

September 14, 2021

The following expression prefixes return the contents of the HTTP Date header as text or as a date object. These values can be evaluated as follows:

- As a number. The numeric value of an HTTP Date header is returned in the form of the number of seconds since Jan 1, 1970.

  For example, the expression http.req.date.mod(86400) returns the number of seconds since the beginning of the day. These values can be evaluated using the same operations as other non-date-related numeric data. For more information, see Expression prefixes for numeric data other than date and time.

- As an HTTP header. Date headers can be evaluated using the same operations as other HTTP headers.

  For more information, see Default syntax expressions: Parsing HTTP, TCP, and UDP data.

- As text. Date headers can be evaluated using the same operations as other strings.

For more information, see Advanced Policy Expressions: Evaluating Text.

Prefix	Description
HTTP.REQ.DATE	Returns the contents of the HTTP Date header as text or as a date object. The date formats recognized are: RFC822. Sun, 06 Jan 1980 08:49:37 GMT, RFC850. Sunday, 06-Jan-80 09:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.
HTTP.RES.DATE	Returns the contents of the HTTP Date header as text or as a date object. The date formats recognized are: RFC822. Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.

Generate the day of the week, as a string, in short and long formats

September 14, 2021

The functions, WEEKDAY_STRING_SHORT and WEEKDAY_STRING, generate the day of the week, as a string, in short and long formats, respectively. The strings that are returned are always in English. The prefix used with these functions must return the day of the week in integer format and the acceptable range for the value returned by the prefix is 0-6. Therefore, you can use any prefix that returns an integer in the acceptable range. An UNDEF condition is raised if the returned value is not in this range or if memory allocation fails.

Following are the descriptions of the functions:

Function	Description
<prefix>.WEEKDAY_STRING_SHORT	Returns the day of the week in short format. The short form is always 3 characters long with an initial capital and the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT returns Sun if the value returned by the WEEKDAY function is 0 and Sat if the value returned by the prefix is 6.
<prefix>.WEEKDAY_STRING	Returns the day of the week in long format. The long form always has an initial capital, with the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING returns Sunday if the value returned by the WEEKDAY function is 0 and Saturday if the value returned by the prefix is 6.

Expression prefixes for numeric data other than date and time

September 14, 2021

In addition to configuring expressions that operate on time, you can configure expressions for the following types of numeric data:

- The length of HTTP requests, the number of HTTP headers in a request, and so on.

  For more information, see Expressions for numeric HTTP payload data other than dates.

- IP and MAC addresses.

  For more information, see Expressions for IP addresses and IP subnets.

- Client and server data in regard to interface IDs and transaction throughput rate.

  For more information, see Expressions for numeric client and server data.

- Numeric data in client certificates other than dates.

  For information on these prefixes, including the number of days until certificate expiration and the encryption key size, see Prefixes for numeric data in SSL certificates.

Converting numbers to text

September 14, 2021

The following functions produce binary strings from a number returned by an expression prefix. These functions are particularly useful in the TCP rewrite feature as replacement strings for binary data. For more information about the TCP rewrite feature, see Rewrite.

All the functions return a value of type text. The endianness that some of the functions accept as a parameter is either LITTLE_ENDIAN or BIG_ENDIAN.

Function	Description
<number>.SIGNED8_STRING	Produces an 8-bit signed binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STR
<number>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED
<number>.SIGNED16_STRING(<endianness>)	Produces a 16-bit signed binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)

Function	Description
<number>.UNSIGNED16_STRING(<endianness>)	Produces a 16-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LIT-TLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)
<number>.SIGNED32_STRING(<endianness>)	Produces a 32-bit signed binary string representing the number. Example: HTTP.REQ.BODY(100).AFTER_STR(“delim”).GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)
<unsigned_long_number>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIG
<unsigned_long_number>.UNSIGNED16_STRING	Produces a 16-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LIT-TLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UNSI
<unsigned_long_number>.UNSIGNED32_STRING(<endianness>)	Produces a 32-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).AFTER_STR(“delim2”).GET_UNSIGNED3 BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)

Virtual server based expressions

September 14, 2021

The `SYS.VSERVER("<vserver-name>")` expression prefix enables you to identify a virtual server. You can use the following functions with this prefix to retrieve information related to the specified virtual server:

- **THROUGHPUT.** Returns the throughput of the virtual server in Mbps (Megabits per second). The value returned is an unsigned long number.

Usage: SYS.VSERVER(“vserver”).THROUGHPUT

- **CONNECTIONS.** Returns the number of connections being managed by the virtual server. The value returned is an unsigned long number.

  Usage: SYS.VSERVER(“vserver”).CONNECTIONS

- **STATE.** Returns the state of the virtual server. The value returned is UP, DOWN, or OUT_OF_SERVICE. One of these values can therefore be passed as an argument to the EQ() operator to perform a comparison that results in a Boolean TRUE or FALSE.

  Usage: SYS.VSERVER(“vserver”).STATE

- **HEALTH.** Returns the percentage of services in an UP state for the specified virtual server. The value returned is an integer.

  Usage: SYS.VSERVER(“vserver”).HEALTH

- **RESPTIME.** Returns the response time as an integer representing the number of microseconds. Response time is the average TTFB (Time To First Byte) from all the services bound to the virtual server.

  Usage: SYS.VSERVER(“vserver”).RESPTIME

- **SURGECOUNT.** Returns the number of requests in the surge queue of the virtual server. The value returned is an integer.

  Usage: SYS.VSERVER(“vserver”).SURGECOUNT

Example 1:

The following rewrite policy aborts rewrite processing if the number of connections at the load balancing virtual server LBvserver exceeds 10000:

```
add rewrite policy norewrite_pol sys.vserver("LBvserver").connections.gt
(10000)norewrite
```

Example 2:

The following rewrite action inserts a custom header, TP, whose value is the throughout at the virtual server LBvserver:

```
add rewrite action tp_header insert_http_header TP SYS.VSERVER("LBvserver")
.THROUGHPUT
```

Example 3:

The following audit log message action writes the average TTFB of the services bound to a virtual server, to the newnslog log file:

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS
Response Time to Servers:\" + sys.vserver(\"ssllb\").resptime + \" millisec
\""-logtoNewnslog YES -bypassSafetyCheck YES
```


Advanced policy expressions: Parsing HTTP, TCP, and UDP data

September 14, 2021

You can configure advanced policy expressions to evaluate the payload in an HTTP request or response. The payload associated with an HTTP connection includes HTTP headers (standard or custom headers), body, and connection URL. Also, you can evaluate and process the payload in a TCP or a UDP packet. For HTTP connections, for example, you can check whether a particular HTTP header is present or if the URL includes a particular query parameter.

You can configure expressions to transform the URL encoding and apply HTML or XML "safe" coding for subsequent evaluation. You can also use XPATH and JSON prefixes to evaluate date in XML and JSON files, respectively.

For more information about authentication expressions such as AAA.USER, AAA.LOGIN, see authentication, authorization, and auditing login and for AAA.AUTHENTICATION expression, see Citrix ADC AAA user authentication topics.

You can also use text-based and numeric Advanced policy expressions to evaluate HTTP request and response data. For more information, see Advanced policy expressions: Evaluating text and Default Syntax Expressions: Working with Dates, Times, and Numbers.

Expressions for identifying the protocol in an incoming IP packet

September 14, 2021

The following table lists the expressions that you can use to identify the protocol in an incoming packet.

Expression	Description
CLIENT.IP.PROTOCOL	Identifies the protocol in IPv4 packets sent by clients.
CLIENT.IPV6.PROTOCOL	Identifies the protocol in IPv6 packets sent by clients.
SERVER.IP.PROTOCOL	Identifies the protocol in IPv4 packets sent by servers.
SERVER.IPV6.PROTOCOL	Identifies the protocol in IPv6 packets sent by servers.

Arguments to the PROTOCOL function

You can pass the Internet Assigned Numbers Authority (IANA) protocol number to the PROTOCOL function. For example, if you want to determine whether the protocol in an incoming packet is TCP, you can use CLIENT.IP.PROTOCOL.EQ(6), where 6 is the IANA-assigned protocol number for TCP. For some protocols, you can pass an enumeration value instead of the protocol number. For example, instead of CLIENT.IP.PROTOCOL.EQ(6), you can use CLIENT.IP.PROTOCOL.EQ(TCP). The following table lists the protocols for which you can use enumeration values, and the corresponding enumeration values for use with the PROTOCOL function.

Protocol	Enumeration value
Transmission Control Protocol (TCP)	TCP
User Datagram Protocol (UDP)	UDP
Internet Control Message Protocol (ICMP)	ICMP
IP Authentication Header (AH), for providing authentication services in IPv4 and IPv6	AH
Encapsulating Security Payload (ESP) protocol	ESP
General Routing Encapsulation (GRE)	GRE
IP-within-IP Encapsulation Protocol	IPIP
Internet Control Message Protocol for IPv6 (ICMPv6)	ICMPv6
Fragment Header for IPv6	FRAGMENT

Use case scenarios

The protocol expressions can be used in both request-based and response-based policies. You can use the expressions in various Citrix ADC features, such as load balancing, WAN optimization, content switching, rewrite, and listen policies. You can use the expressions with functions such as EQ() and NE(), to identify the protocol in a policy and perform an action.

Following are some use cases for the expressions:

- In Branch Repeater load balancing configurations, you can use the expressions in a listen policy for the wildcard virtual server. For example, you can configure the wildcard virtual server with the listen policy CLIENT.IP.PROTOCOL.EQ(TCP) so that the virtual server processes only TCP traffic and simply bridges all non-TCP traffic. Even though you can use an Access Control List instead of the listen policy, the listen policy provides better control over what traffic is processed.

- For content switching virtual servers of type ANY, you can configure content switching policies that switch requests on the basis of the protocol in incoming packets. For example, you can configure content switching policies to direct all TCP traffic to one load balancing virtual server and all non-TCP traffic to another load balancing virtual server.
- You can use the client-based expressions to configure persistence based on the protocol. For example, you can use CLIENT.IP.PROTOCOL to configure persistence on the basis of the protocols in incoming IPv4 packets.

Expressions for HTTP and cache-control headers

September 14, 2021

One common method of evaluating HTTP traffic is to examine the headers in a request or a response. A header can perform a number of functions, including the following:

- Provide cookies that contain data about the sender.
- Identify the type of data that is being transmitted.
- Identify the route that the data has traveled (the Via header).

> **Note**
>
> If an operation is used to evaluate both header and text data, the header-based operation always overrides the text-based operation. For example, the AFTER_STR operation, when applied to a header, overrides text-based AFTER_STR operations for all instances of the current header type.

Prefixes for HTTP headers

The Prefixes for HTTP headers table for expression prefixes that extract HTTP headers.

Operations for HTTP headers

The Operations for HTTP headers table for operations that you can specify with the prefixes for HTTP headers.

Prefixes for cache-control headers

The following prefixes apply specifically to Cache-Control headers.

HTTP Header Prefix	Description
HTTP.REQ.CACHE_CONTROL	Returns a Cache-Control header in an HTTP request.
HTTP.RES.CACHE_CONTROL	Returns a Cache-Control header in an HTTP response.

Operations for cache-control headers

You can apply any of the operations for HTTP headers to Cache-Control headers.

In addition, the following operations identify specific types of Cache-Control headers. See RFC 2616 for information about these header types.

HTTP Header Operation	Description
`Cache-Control header.NAME(<integer>)`	Returns as a text value the name of the Cache-Control header that corresponds to the nth component in a name-value list, as specified by `<integer>`. The index of the name-value component is 0-based. If the `<integer>` that is specified by the integer argument is greater than the number of components in the list, a zero-length text object is returned. Following is an example: `http.req.cache_control.name(3).contains("some_text")`
Cache-Control header.IS_INVALID	Returns a Boolean TRUE if the Cache-Control header is not present in the request or response. Following is an example: `http.req.cache_control.is_invalid`
`Cache-Control header.IS_PRIVATE`	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: `http.req.cache_control.is_private`
Cache-Control header.IS_PUBLIC	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: http.req.cache_control.is_public

HTTP Header Operation	Description
Cache-Control header.IS_NO_STORE	Returns a Boolean TRUE if the Cache-Control header has the value No-Store. Following is an example: http.req.cache_control.is_no_store
Cache-Control header.IS_NO_CACHE	Returns a Boolean TRUE if the Cache-Control header has the value No-Cache. Following is an example: http.req.cache_control.is_no_cache
Cache-Control header.IS_MAX_AGE	Returns a Boolean TRUE if the Cache-Control header has the value Max-Age. Following is an example: http.req.cache_control.is_max_age
Cache-Control header.IS_MIN_FRESH	Returns a Boolean TRUE if the Cache-Control header has the value Min-Fresh. Following is an example: http.req.cache_control.is_min_fresh
Cache-Control header.IS_MAX_STALE	Returns a Boolean TRUE if the Cache-Control header has the value Max-Stale. Following is an example: http.req.cache_control.is_max_stale
Cache-Control header.IS_MUST_REVALIDATE	Returns a Boolean TRUE if the Cache-Control header has the value Must-Revalidate. Following is an example: http.req.cache_control.is_must_revalidate
Cache-Control header.IS_NO_TRANSFORM	Returns a Boolean TRUE if the Cache-Control header has the value No-Transform. Following is an example: http.req.cache_control.is_no_transform
Cache-Control header.IS_ONLY_IF_CACHED	Returns a Boolean TRUE if the Cache-Control header has the value Only-If-Cached. Following is an example: http.req.cache_control.is_only_if_cached
Cache-Control header.IS_PROXY_REVALIDATE	Returns a Boolean TRUE if the Cache-Control header has the value Proxy-Revalidate. Following is an example: http.req.cache_control.is_proxy_revalidate
Cache-Control header.IS_S_MAXAGE	Returns a Boolean TRUE if the Cache-Control header has the value S-Maxage. Following is an example: http.req.cache_control.is_s_maxage

HTTP Header Operation	Description
Cache-Control header.IS_UNKNOWN	Returns a Boolean TRUE if the Cache-Control header is of an unknown type. Following is an example: http.req.cache_control.is_unknown
Cache-Control header.MAX_AGE	Returns the value of the Cache-Control header Max-Age. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.max_age.le(3)
Cache-Control header.MAX_STALE	Returns the value of the Cache-Control header Max-Stale. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.max_stale.le(3)
Cache-Control header.MIN_FRESH	Returns the value of the Cache-Control header Min-Fresh. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.min_fresh.le(3)
Cache-Control header.S_MAXAGE	Returns the value of the Cache-Control header S-Maxage. If this header is absent or invalid, 0 is returned.Following is an example: http.req.cache_control.s_maxage.eq(2)

Expressions for extracting segments of URLs

September 14, 2021

You can extract URLs and portions of URLs, such as the host name, or a segment of the URL path. For example, the following expression identifies HTTP requests for image files by extracting image file suffixes from the URL:

```
http.req.url.suffix.eq("jpeg")|| http.req.url.suffix.eq("gif")
```

Most expressions for URLs operate on text and are described in Expression Prefixes for Text in HTTP Requests and Responses. This section discusses the GET operation. The GET operation extracts text when used with the following prefixes:

- HTTP.REQ.URL.PATH
- VPN.BASEURL.PATH
- VPN.CLIENTLESS_BASEURL.PATH

The following table describes prefixes for HTTP URLs.

URL Prefix	Description
HTTP.REQ.URL.PATH.GET(<n>)	Returns a slash- (“/”) separated list from the URL path. For example, consider the following URL:<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>. The following expression returns dir1 from this URL:<http.req.url.path.get(1)>. The following expression returns dir2: http.req.url.path.get(2)
HTTP.REQ.URL.PATH.GET_REVERSE(<n>)	Returns a slash- (“/”) separated list from the URL path, starting from the end of the path. For example, consider the following URL: <http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>. The following expression returns index.html from this URL: <http.req.url.path.get_reverse(0)>. The following expression returns dir3: http.req.url.path.get_reverse(1)

Expressions for HTTP status codes and numeric HTTP payload data other than dates

September 14, 2021

The following table describes prefixes for numeric values in HTTP data other than dates.

Prefix	Description
HTTP.REQ.CONTENT_LENGTH	Returns the length of an HTTP request as a number. Following is an example: http.req.content_length < 500
HTTP.RES.CONTENT_LENGTH	Returns the length of the HTTP response as a number. Following is an example: http.res.content_length <= 1000
HTTP.RES.STATUS	Returns the response status code

Prefix	Description
HTTP.RES.IS_REDIRECT	Returns a Boolean TRUE if the response code is associated with a redirect. Following are the redirect response codes: 300 (Multiple Choices), 301 (Moved Permanently), 302 (Found), 303 (See Other), 305 (Use Proxy), and 307 (Temporary Redirect). Note: Status code 304 is not considered a redirect HTTP response status code. Status code 306 is unused.

SIP expressions

September 14, 2021

The Citrix ADC Advanced policy expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions are intended to be used in policies for any supported protocol that operates on a request/response basis. These expressions can be used in content switching, rate limiting, responder, and rewrite policies.

Certain limitations apply to SIP expressions used with responder policies. Only the DROP, NOOP or RESPONDWITH actions are allowed on a SIP load balancing virtual server. Responder policies can be bound to a load balancing virtual server, an override global bind point, a default global bind point, or a sip_udp policy label.

The header format used by the SIP protocol is similar to that used by the HTTP protocol, so many of the new expressions look and function much like their HTTP analogs. Each SIP header consists of a line that includes the SIP method, the URL, and the version, followed by a series of name-value pairs that look like HTTP headers.

Following is a sample SIP header that is referred to in the expressions tables beneath it:

```
INVITE sip:16@www.sip.com:5060;transport=udp SIP/2.0
Record-Route: <sip:200.200.100.22;lr=on>
Via: SIP/2.0/UDP 200.200.100.22;branch=z9hG4bK444b.c8e103d1.0;rport
   =5060;
    received=10.102.84.18
Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;
    received=10.102.84.160
From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53
   cc0185
To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
```



```
Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180
Max-Forwards: 69CSeq: 101 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:12@10.102.84.180:5060;transport=udp>
Expires: 180
Accept: application/sdp
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 277
Content-Type: application/sdp
Content-Disposition: session;handling=optiona
<!--NeedCopy-->
```

SIP reference tables

The following tables contain lists of expressions that operate on SIP headers. The first table contains expressions that apply to request headers. Most response-based expressions are nearly the same as the corresponding request-based expressions. To create a response expression from the corresponding request expression, you change the first two sections of the expression from SIP.REQ to SIP.RES, and make other obvious adjustments. The second table contains those response expressions that are unique to responses and have no request equivalents. You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

SIP request expressions

Expression	Description
SIP.REQ.METHOD	Operates on the method of the SIP request. The supported SIP request methods are ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, OPTIONS, PRACK, PUBLISH, REFER, REGISTER, SUBSCRIBE, and UPDATE. This expression is a derivative of the text class, so all operations that are applicable to text are applicable to this method. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns INVITE.

Expression	Description
SIP.REQ.URL	Operates on the SIP request URL. This expression is a derivative of the text class, so all operations that are applicable to text are applicable to this method. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression re-turnssip:16@10.102.84.181:5060;transport=udp.
SIP.REQ.URL.PROTOCOL	Returns the URL protocol. For example, for a SIP URL ofsip:16@www.sip.com:5060;transport=udp, this expression returns sip.
SIP.REQ.URL.HOSTNAME	Returns the hostname portion of the SIP URL. For example, for a SIP URL ofsip:16@www.sip.com:5060;transport=udp, this expression returns www.sip.com:5060.
SIP.REQ.URL.HOSTNAME.PORT	Returns the port portion of the SIP URL hostname. If no port is specified, this expression returns the default SIP port, 5060. For example, for a SIP hostname of www.sip.com:5060, this expression returns 5060.
SIP.REQ.URL.HOSTNAME.DOMAIN	Returns the domain name portion of the SIP URL hostname. If the host is an IP address, then this expression returns an incorrect result. For example, for a SIP hostname of www.sip.com:5060, this expression returns sip.com. For a SIP hostname of 192.168.43.15:5060, this expression returns an error.
SIP.REQ.URL.HOSTNAME.SERVER	Returns the server portion of the host. For example, for a SIP hostname of www.sip.com:5060, this expression returns www.

Expression	Description
SIP.REQ.URL.USERNAME	Returns the username that precedes the @ character. For example, for a SIP URL of sip:16@www.sip.com:5060;transport=udp, this expression returns 16.
SIP.REQ.VERSION	Returns the SIP version number in the request. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns SIP/2.0.
SIP.REQ.VERSION.MAJOR	Returns the major version number (the number to the left of the period). For example, for a SIP version number of SIP/2.0, this expression returns 2.
SIP.REQ.VERSION.MINOR	Returns the minor version number (the number to the right of the period). For example, for a SIP version number of SIP/2.0, this expression returns 0.
SIP.REQ.CONTENT_LENGTH	Returns the contents of the Content-Length header. This expression is a derivative of thesip_header_t class, so all operations that are available for SIP headers can be used. For example, for a SIP Content-Length header of Content-Length: 277, this expression returns 277.
SIP.REQ.TO	Returns the contents of the To header. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185.
SIP.REQ.TO.ADDRESS	Returns the SIP URI, which is found in the sip_url object. All operations that are available for SIP URIs can be used. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:16@sip_example.com.

Expression	Description
SIP.REQ.TO.DISPLAY_NAME	Returns the display name portion of the To header. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 16.
SIP.REQ.TO.TAG	Returns the “tag” value from the “tag” name value pair in the TO header. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 00127f54ec85a6d90cc14f45-53cc0185.
SIP.REQ.FROM	Returns the contents of the From header. For example, for a SIP From header of From: “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:12@sip_example.com.
SIP.REQ.FROM.ADDRESS	Returns the SIP URI, which is found in the sip_url object. All operations that are available for SIP URIs can be used. For example, for a SIP From header of From: “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:12@sip_example.com.
SIP.REQ.FROM.DISPLAY_NAME	Returns the display name portion of the To header. For example, for a SIP From header of From: “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 12.
SIP.REQ.FROM.TAG	Returns the “tag” value from the “tag” name/value pair in the TO header. For example, for a SIP From header of From: “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 00127f54ec85a6d90cc14f45-53cc0185.

Expression	Description
SIP.REQ.VIA	Returns the complete Via header. If there are multiple Via headers in the request, returns the last Via header. For example, for the two Via headers in the sample SIP header, this expression returns Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re
SIP.REQ.VIA.SENTBY_ADDRESS	Returns the address that sent the request. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 10.102.84.180.
SIP.REQ.VIA.SENTBY_PORT	Returns the port that sent the request. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 5060.
SIP.REQ.VIA.RPORT	Returns the value from the rport name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 5060.
SIP.REQ.VIA.BRANCH	Returns the value from the branch name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns z9hG4bK03e76d0b.
SIP.REQ.VIA.RECEIVED	Returns the value from the received name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 10.102.84.160.

Expression	Description
SIP.REQ.CALLID	Returns the contents of the Callid header. This expression is a derivative of the sip_header_t class, so all operations that are available for SIP headers can be used. For example, for a SIP Callid header ofCall-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180, this expression returns00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180.
SIP.REQ.CSEQ	Returns the CSEQ number from the CSEQ, as an integer. For example, for a SIP CSEQ header of CSeq: 101 INVITE, this expression returns 101.
SIP.REQ.HEADER(<header_name>)	Returns the specified SIP header. For <header_name>, substitute the name of the header that you want. For example, to return the SIP From header, you would type SIP.REQ.HEADER(“From”).
SIP.REQ.HEADER(<header_name>).INSTANCE(<line_number>)	Returns the specified instance of the specified SIP header. Multiple instances of the same SIP header can occur. Where you want a specific instance of such a SIP header (for example, a specific Via header), you can specify that header by typing a number as the <line_number>. Header instances are matched from last (0) to first. In other words, SIP.REQ.HEADER(“Via”).INSTANCE(0) returns the last instance of the Via header, while SIP.REQ.HEADER(“Via”).INSTANCE(1) returns the last instance but one of the Via header, and so on. For example, if used on the example SIP header, SIP.REQ.HEADER(“Via”).INSTANCE(1) returnsVia: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060.

Expression	Description
SIP.REQ.HEADER(<header_name>).VALUE(<line_	Returns the contents of the specified instance of the specified SIP header. The usage is nearly the same as the previous expression. For example, if used on the SIP header example in the preceding table entry, SIP.REQ.HEADER("Via").VALUE(1) returns SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060.
SIP.REQ.HEADER(<header_name>).COUNT	Returns the number of instances of a particular header as an integer. For example, if used on the SIP header example above, SIP.REQ.HEADER("Via").COUNT returns 2.
SIP.REQ.HEADER(<header_name>).EXISTS	Returns a boolean value of true or false, depending upon whether the specified header exists or not. For example, if used on the SIP header example above, SIP.REQ.HEADER("Expires").EXISTSreturns true, while SIP.REQ.HEADER("Caller-ID").EXISTS returns false.
SIP.REQ.HEADER(<header_name>).LIST	Returns the comma-separated parameter list in the specified header. For example, if used on the SIP header example above, SIP.REQ.HEADER("Allow").LIST returns ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,U You can append the string .GET(<list_item_number>) to select a specific list item. For example, to get the first item (ACK) from the above list, you would type SIP.REQ.HEADER("Allow").LIST.GET(0). To extract the second item (BYE), you would type SIP.REQ.HEADER("Allow").LIST.GET(1). Note: If the specified header contains a list of name/value pairs, the entire name/value pair is returned.

Expression	Description
SIP.REQ.HEADER(<header_name>).TYPECAST_SI	Typecasts <header_name> to <in_header_name>. Any text can be typecasted to thesip_header_t class, after which all header-based operations can be used. After you perform this operation, you can apply all operations that can be used with <in_header_name>. For example, the expression SIP.REQ.CONTENT_LENGTH.TYPECAST_SIP_HEADER_T typecasts all instances of the Content-Length header. After you perform this operation, you can apply all header operations to all instances of the specified header.
SIP.REQ.HEADER(<header_name>).CONTAINS(<string>)	Returns boolean true if the specified text string is present in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).EQUALS_ANY	Returns boolean true if any pattern associated with <patset> matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).CONTAINS_ANY(<patset>)	Returns boolean true if any pattern associated with <patset> matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).CONTAINS_IN	Returns the index of the matching pattern associated with <patset> if that pattern matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.

Expression	Description
SIP.REQ.HEADER(<header_name>).EQUALS_INDEX(<patset>)	Returns the index of the matching pattern associated with <patset> if that pattern matches any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).SUBSTR(<stri	If the specified string is present in any instance of the specified header, this expression returns that string. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re returns “rport=5060”.SIP.REQ.HEADER(“Via”).SUBSTR(“rport=5061”) returns an empty string.
SIP.REQ.HEADER(<header_name>).AFTER_STR(<string>)	If the specified string is present in any instance of the specified header, this expression returns the string immediately after that string. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER(“Via”).AFTER_STR(“rport=”) returns 5060.

Expression	Description
SIP.REQ.HEADER(<header_name>).REGEX_MATC	Returns boolean true if the specified regular expression (regex) matches any instance of the specified header. You must specify the regular expression in the following format: re<delimiter>regular expression<same delimiter>. The regular expression cannot be larger than 1499 characters in length. It must conform to the PCRE regular expression library. See http://www.pcre.org/pcre.txt for documentation on PCRE regular expression syntax. The pcrepattern man page also has useful information on specifying patterns by using PCRE regular expressions. The regular expression syntax supported in this expression has some differences from PCRE. Back references are not allowed. You should avoid recursive regular expressions; although some work, many do not. The dot (.) metacharacter matches newlines. Unicode is not supported.SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option specified in the regular expression.
SIP.REQ.HEADER(<header_name>).REGEX_SELECT(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER("Via").REGEX_SELECT("received=[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}") returns received=10.102.84.160.

Expression	Description
SIP.REQ.HEADER(<header_name>).AFTER_REGE	If the specified regex matches any text in any instance of the specified header, this expression returns the string immediately after that text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER("Via").AFTER_REGEX("received=") returns10.102.84.160.
SIP.REQ.HEADER(<header_name>).BEFORE_REGEX(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the string immediately before that text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER("Via").BEFORE_REGEX("[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}") returns received=.
SIP.REQ.FULL_HEADER	Returns the entire SIP header, including the terminating CR/LF.
SIP.REQ.IS_VALID	Returns boolean true if the request format is valid.
SIP.REQ.BODY(<length>)	Returns the request body, up to the specified length. If the specified length is greater than the length of the request body, this expression returns the entire request body.
SIP.REQ.LB_VSERVER	Returns the name of the load balancing virtual server (LB vserver) that is serving the current request.
SIP.REQ.CS_VSERVER	Returns the name of the content switching virtual server (CS vserver) that is serving the current request.

SIP response expressions

Expression	Description
SIP.RES.STATUS	Returns the SIP response status code. For example, if the first line of the response is SIP/2.0 100 Trying, this expression returns 100.
SIP.RES.STATUS_MSG	Returns the SIP response status message. For example, if the first line of the response is SIP/2.0 100 Trying, this expression returns Trying.
SIP.RES.IS_REDIRECT	Returns boolean true if the response code is a redirect.
SIP.RES.METHOD	Returns the response method extracted from the request method string in the CSeq header.

Operations for HTTP, HTML, and XML encoding and “safe” characters

September 14, 2021

The following operations work with the encoding of HTML data in a request or response and XML data in a POST body.

- **<text>.HTML_XML_SAFE:**
  Transforms special characters into XML safe format, as in the following examples:

  A left-pointing angle bracket (<) is converted to <
  A right-pointing angle bracket (>) is converted to >
  An ampersand (&) is converted to &
  This operation safeguards against cross-site scripting attacks. Maximum length of the transformed text is 2048 bytes. This is a read-only operation.

  After applying the transformation, additional operators that you specify in the expression are applied to the selected text. Following is an example:

  http.req.url.query.html_xml_safe. contains(“myQueryString”)

- **<text>.HTTP_HEADER_SAFE:**
  Converts all new line (‘\n’) characters in the input text to ‘%0A’ to enable the input to be used safely in HTTP headers.

  This operation safeguards against response-splitting attacks.

  The maximum length of the transformed text is 2048 bytes. This is a read-only operation.

- **<text>.HTTP_URL_SAFE:**
  Converts unsafe URL characters to '%xx' values, where "xx" is a hex-based representation of the input character. For example, the ampersand (&) is represented as %26 in URL-safe encoding. The maximum length of the transformed text is 2048 bytes. This is a read-only operation.

  Following are URL safe characters. All others are unsafe:

  - Alpha-numeric characters: a-z, A-Z, 0-9
  - Asterix: "*"
  - Ampersand: "&"
  - At-sign: "@"
  - Colon: ":"
  - Comma: ","
  - Dollar: "$"
  - Dot: "."
  - Equals: "="
  - Exclamation mark: "!"
  - Hyphen: "-"
  - Open and close parentheses: "(", ")"
  - Percent: "%"
  - Plus: "+"
  - Semicolon: ";"
  - Single quote: ""'
  - Slash: "/"
  - Question mark: "?"
  - Tilde: "~"
  - Underscore: "_"
- **<text>.MARK_SAFE:**

  Marks the text as safe without applying any type of data transformation.

<text>.SET_TEXT_MODE(URLENCODED NOURLENCODED)

- Transforms all %HH encoding in the byte stream. This operation works with characters (not bytes). By default, a single byte represents a character in ASCII encoding. However, if you specify URLENCODED mode, three bytes can represent a character.

  In the following example, a PREFIX(3) operation selects the first 3 characters in a target.

  ```
  http.req.url.hostname.prefix(3)
  ```

  In the following example, the Citrix ADC can select up to 9 bytes from the target:


```
http.req.url.hostname.set_text_mode(urlencoded).prefix(3)
```

<text>.SET_TEXT_MODE(PLUS_AS_SPACE NO_PLUS_AS_SPACE):

- Specifies how to treat the plus character (+). The PLUS_AS_SPACE option replaces a plus character with white space. For example, the text “hello+world” becomes “hello world.” The NO_PLUS_AS_SPACE option leaves plus characters as they are.

<text>.SET_TEXT_MODE(BACKSLASH_ENCODE NO_BACKSLASH_ENCODED):

- Specifies whether or not backslash decoding is performed on the text object represented by <text>.

  If BACKSLASH_ENCODED is specified, the SET_TEXT_MODE operator performs the following operations on the text object:

  - All occurrences of “\XXX” will be replaced with the character “Y” (where XXX represents a number in the octal system and Y represents the ASCII equivalent of XXX). The valid range of octal values for this type of encoding is 0 to 377. For example, the encoded text “http\72//” and http\072//” will both be decoded to <http://>, where the colon (:) is the ASCII equivalent of the octal value “72”.
  - All occurrences of “\xHH” will be replaced with the character “Y” (HH represents a number in the hexadecimal system and Y denotes the ASCII equivalent of HH. For example, the encoded text “http\x3a//” will be decoded to <http://>, where the colon (:) is the ASCII equivalent of the hexadecimal value “3a”.
  - All occurrences of “\uWWXX” will be replaced with the character sequence “YZ” (Where WW and XX represent two distinct hexadecimal values and Y and Z represent their ASCII equivalents of WW and XX respectively. For example, the encoded text “http%u3a2f/” and “http%u003a//” will both be decoded to <http://>, where “3a” and “2f” are two hexadecimal values and the colon (:) and forward slash (“/”) represent their ASCII equivalents respectively.
  - All occurrences of “\b”, “\n”, “\t”, “\f”, and “\r” are replaced with the corresponding ASCII characters.

  If NO_BACKSLASH_ENCODED is specified, backslash decoding is not performed on the text object.

<text>.SET_TEXT_MODE(BAD_ENCODE_RAISE_ NO_BAD_ENCODE_RAISE_UNDEF):

- Performs the associated undefined action if either the URLENCODED or the BACKSLASH_ENCODED mode is set and bad encoding corresponding to the specified encoding mode is encountered in the text object represented by <text>.

  If NO_BAD_ENCODE_RAISE_UNDEF is specified, the associated undefined action will not be performed when bad encoding is encountered in the text object represented by<text>.

Expressions for TCP, UDP, and VLAN data

September 14, 2021

TCP and UDP data take the form of a string or a number. For expression prefixes that return string values for TCP and UDP data, you can apply any text-based operations. For more information, see Advanced policy expressions: Evaluating text.

For expression prefixes that return numeric value, such as a source port, you can apply an arithmetic operation. For more information, see Basic operations on expression prefixes and Compound operations for numbers.

The following table describes prefixes that extract TCP and UDP data.

GET Operation	Description
`CLIENT.TCP.PAYLOAD(<integer>)`	Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the `<integer>` argument. You can apply any text-based operation to this prefix.
CLIENT.TCP.SRCPORT	Returns the ID of the current packet's source port as a number.
CLIENT.TCP.DSTPORT	Returns the ID of the current packet's destination port as a number.
CLIENT.TCP.OPTIONS	Returns the TCP options set by the client. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this prefix. For the TCP options set by the server, see the SERVER.TCP.OPTIONS prefix.

GET Operation	Description
CLIENT.TCP.OPTIONS.COUNT	Returns the number of TCP options that the client has set.
CLIENT.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or option kind) is specified as the argument. The value is returned as a string of bytes in big endian format (or network byte order). Parameters: type - Type value
CLIENT.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). Parameters: m - TCP option enumeration constant
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns a Boolean TRUE if Repeater TCP options exist.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the branch repeater's IPv4 address from the Repeater TCP options.
CLIENT.TCP.REPEATER_OPTION.MAC	Returns the branch repeater's MAC address from the Repeater TCP options.
CLIENT.UDP.DNS.DOMAIN	Returns the DNS domain name.
CLIENT.UDP.DNS.DOMAIN.EQ("<hostname>")	Returns a Boolean TRUE if the domain name matches the <hostname> argument. The compari- son is case insensitive. Following is an example: client.udp.dns.domain.eq("www.mycompany.com")
CLIENT.UDP.DNS.IS_AAAAREC	Returns a Boolean TRUE if the record type is AAAA. These types of records indicate an IPv6 address in forward lookups.

GET Operation	Description
CLIENT.UDP.DNS.IS_ANYREC	Returns a Boolean TRUE if it is of any record type.
CLIENT.UDP.DNS.IS_AREC	Returns a Boolean TRUE if the record is type A. Type A records provide the host address.
CLIENT.UDP.DNS.IS_CNAMEREC	Returns a Boolean TRUE if the record is of type CNAME. In systems that use multiple names to identify a resource, there is one canonical name and a number of aliases. The CNAME provides the canonical name.
CLIENT.UDP.DNS.IS_MXREC	Returns a Boolean TRUE if the record is of type MX (mail exchanger). This DNS record describes a priority and a host name. The MX records for the same domain name specify the email servers in the domain and the priority for each server.
CLIENT.UDP.DNS.IS_NSREC	Returns a Boolean TRUE if the record is of type NS. This is a name server record that includes a host name with an associated A record. This enables locating the domain name that is associated with the NS record.
CLIENT.UDP.DNS.IS_PTRREC	Returns a Boolean TRUE if the record is of type PTR. This is a domain name pointer and is often used to associate a domain name with an IPv4 address.
CLIENT.UDP.DNS.IS_SOAREC	Returns a Boolean TRUE if the record is of type SOA. This is a start of authority record.
CLIENT.UDP.DNS.IS_SRVREC	Returns a Boolean TRUE if the record is of type SRV. This is a more general version of the MX record.
CLIENT.UDP.DSTPORT	Returns the numeric ID of the current packet's UDP destination port.
CLIENT.UDP.SRCPORT	Returns the numeric ID of the current packet's UDP source port.
CLIENT.UDP.RADIUS	Returns RADIUS data for the current packet.

GET Operation	Description
CLIENT.UDP.RADIUS.ATTR_TYPE(<type>)	Returns the value for the attribute type specified as the argument.
CLIENT.UDP.RADIUS.USERNAME	Returns the RADIUS user name.
CLIENT.TCP.MSS	Returns the maximum segment size (MSS) for the current connection as a number.
CLIENT.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the Citrix ADC.
SERVER.TCP.DSTPORT	Returns the numeric ID of the current packet's destination port.
SERVER.TCP.SRCPORT	Returns the numeric ID of the current packet's source port.
SERVER.TCP.OPTIONS	Returns the TCP options set by the server. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this prefix. For the TCP options set by the client, see the CLIENT.TCP.OPTIONS prefix.
SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options that the server has set.
SERVER.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or option kind) is specified as the argument. The value is returned as a string of bytes in big endian format (or network byte order). Parameters: type - Type value

GET Operation	Description
SERVER.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). Parameters: m - TCP option enumeration constant
SERVER.VLAN	Operates on the VLAN through which the current packet entered the Citrix ADC.
SERVER.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the Citrix ADC.

Expressions for evaluating a DNS message and identifying its carrier protocol

September 14, 2021

You can evaluate DNS requests and responses by using expressions that begin with DNS.REQ and DNS.RES, respectively. You can also identify the transport layer protocol that is being used to send the DNS messages.

The following functions return the contents of a DNS query.

Function	Description
DNS.REQ.QUESTION.DOMAIN	Return the domain name (the value of the QNAME field) in the question section of the DNS query. The domain name is returned as a text string, which can be passed to EQ(), NE(), and any other functions that work with text.

Function	Description
DNS.REQ.QUESTION.TYPE	Return the query type (the value of the QTYPE field) in the DNS query. The field indicates the type of resource record (for example, A, NS, or CNAME) for which the name server is being queried. The returned value can be compared to one of the following values by using the EQ() and NE()functions: A, AAAA, NS, SRV, PTR, CNAME, SOA, MX, and ANY. Note: You can use only the EQ() and NE() functions with the TYPE function. Example: DNS.REQ.QUESTION.TYPE.EQ(MX)

The following functions return the contents of a DNS response.

Function	Description
DNS.RES.HEADER.RCODE	Return the response code (the value of the RCODE field) in the header section of the DNS response. You can use only the EQ() and NE() functions with the RCODE function. Following are the possible values: NOERROR, FORMERR, SERVFAIL, NXDOMAIN, NOTIMP, and REFUSED.
DNS.RES.QUESTION.DOMAIN	Return the domain name (the value of the QNAME field) in the question section of the DNS response. The domain name is returned as a text string, which can be passed to EQ(), NE(), and any other functions that work with text.

Function	Description
DNS.RES.QUESTION.TYPE	Return the query type (the value of the QTYPE field) in the question section of the DNS response. The field indicates the type of resource record (for example, A, NS, or CNAME) that is contained in the response. The returned value can be compared to one of the following values by using the EQ()and NE() functions: A, AAAA, NS, SRV, PTR, CNAME, SOA, MX, and ANY. You can use only the EQ() and NE() functions with the TYPE function. Example: DNS.RES.QUESTION.TYPE.EQ(SOA)

The following functions return the transport layer protocol name.

Function	Description
DNS.REQ.TRANSPORT	Return the name of the transport layer protocol that was used to send the DNS query. Possible values returned are TCP and UDP. You can use only the EQ() and NE() functions with the TRANSPORT function. Example: DNS.REQ.TRANSPORT.EQ(TCP)
DNS.RES.TRANSPORT	Return the name of the transport layer protocol that was used for the DNS response. Possible values returned are TCP and UDP. You can use only the EQ() and NE() functions with the TRANSPORT function. Example: DNS.RES.TRANSPORT.EQ(TCP)

XPath and HTML, XML, or JSON expressions

September 14, 2021

The Advanced policy infrastructure supports expressions for evaluating and retrieving data from HTML, XML, and JavaScript Object Notation (JSON) files. This enables you to find specific nodes in

an HTML, XML, or JSON document, determine if a node exists in the file, locate nodes in XML contexts (for example, nodes that have specific parents or a specific attribute with a given value), and return the contents of such nodes. Additionally, you can use XPath expressions in rewrite expressions.

The Advanced policy expression implementation for XPath comprises an Advanced policy expression prefix (such as "HTTP.REQ.BODY") that designates HTML or XML text, and the XPATH operator that takes the XPath expression as its argument.

HTML files are a largely free-form collection of tags and text elements. You can use the XPATH_HTML operator, which takes an XPath expression as its argument, to process HTML files. JSON files are either a collection of name/value pairs or an ordered list of values. You can use the XPATH_JSON operator, which takes an XPath expression as its argument, to process JSON files.

- **<text>.XPATH(xpathex):**

  Operate on an XML file and return a Boolean value.

  For example, the following expression returns a Boolean TRUE if a node called "creator" exists under the node "Book" within the first 1000 bytes of the XML file.

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%)
  ```

  Parameters:

  xpathex - XPath Boolean expression

- **<text>.XPATH(xpathex):**

  Operate on an XML file and return a value of data type "double."

  For example, the following expression converts the string "36" (a price value) to a value of data type "double" if the string is in the first 1000 bytes of the XML file:

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%)
  ```

  Parameters:

  xpathex - XPath numeric expression

Example:

```
    <Book>
    <creator>
        <Person>
            <name>Milton</name>
        </Person>
    </creator>
    <title>Paradise Lost</title>
    </Book>
<!--NeedCopy-->
```


- **<text>.XPATH(xpathex):**

  Operate on an XML file and return a node-set or a string. Node-sets are converted to corresponding strings by using the standard XPath string conversion routine.

  For example, the following expression selects all the nodes that are enclosed by "/Book/creator" (a node-set) in the first 1000 bytes of the body:

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%)
  ```

  Parameters:

  xpathex - XPath expression

- **<text>.XPATH_HTML(xpathex)**

  Operate on an HTML file and return a text value.

  For example, the following expression operates on an HTML file and returns the text enclosed in <title\></title\> tags if the title HTML element is found in the first 1000 bytes:

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML(xp%/html/head/title%)
  ```

  Parameters:

  xpathex - XPath text expression

- **<text>.XPATH_HTML_WITH_MARKUP(xpathex)**

  Operate on an HTML file and return a string that contains the entire selected portion of the document, including markup such as including the enclosing element tags.

  The following expression operates on the HTML file and selects all content within the <\title> tag, including markup.

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML_WITH_MARKUP( xp%/html/head/title%)
  ```

  The portion of the HTML body that is selected by the expression is marked for further processing.

  Parameters:

  xpathex - XPath expression

- **<text>.XPATH_JSON(xpathex)**

  Operate on a JSON file and return a Boolean value.

  For example, consider the following JSON file:

  { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } }

  The following expression operates on the JSON file and returns a Boolean TRUE if the JSON file contains a node named "creator," whose parent node is "Book," in the first 1000 bytes:

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%)
  ```


Parameters:

xpathex - XPath Boolean expression

- **<text>.XPATH_JSON(xpathex)**

  Operate on a JSON file and return a value of data type “double.”

  For example, consider the following JSON file:

  { “Book”:{ “creator”:{ “person”:{ “name”:’<name>’ } }, “title”:’<title>’, “price”:”36” } }

  The following expression operates on the JSON file and converts the string “36” to a value of data type “double” if the string is present in the first 1000 bytes of the JSON file.

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%)
  ```

  Parameters:

  xpathex - XPath numeric expression

- **<text>.XPATH_JSON(xpathex)**

  Operate on a JSON file and return a node-set or a string. Node-sets are converted to corresponding strings by using the standard XPath string conversion routine.

  For example, consider the following JSON file:

  { “Book”:{ “creator”:{ “person”:{ “name”:’<name>’ } }, “title”:’<title>’ } }

  The following expression selects all the nodes that are enclosed by “/Book” (a node-set) in the first 1000 bytes of the body of the JSON file and returns the corresponding string value, which is “<name><title>”:

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%)
  ```

  Parameters:

  xpathex - XPath expression

- **<text>.XPATH_JSON_WITH_MARKUP(xpathex)**

  Operate on an XML file and return a string that contains the entire portion of the document for the result node, including markup such as including the enclosing element tags.

  For example, consider the following JSON file:

  {“Book”:{ “creator”:{ “person”:{ “name”:’<name>’ } }, “title”:’<title>’ } }

  The following expression operates on the JSON file and selects all the nodes that are enclosed by “/Book/creator” in the first 1000 bytes of the body, which is “creator:{ person:{ name:’<name>’ } }.”

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creator%)
  ```


The portion of the JSON body that is selected by the expression is marked for further processing.

Parameters:

xpathex - XPath expression

- **<text>.XPATH_WITH_MARKUP(xpathex):**

  Operate on an XML file and return a string that contains the entire portion of the document for the result node, including markup such as including the enclosing element tags.

  For example, the following expression operates on an XML file and selects all the nodes enclosed by "/Book/creator" in the first 1000 bytes of the body.

  ```
  HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%)
  ```

  The portion of the JSON body that is selected by the expression is marked for further processing.

  Parameters:

  xpathex - XPath expression

Encrypt and decrypt XML payloads

September 14, 2021

You can use the XML_ENCRYPT() and XML_DECRYPT() functions in Advanced policy expressions to encrypt and decrypt, respectively, XML data. These functions conform to the W3C XML Encryption standard defined at "http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/." XML_ENCRYPT() and XML_DECRYPT() support a subset of the XML Encryption specification. In the subset, data encryption uses a bulk cipher method (RC4, DES3, AES128, AES192, or AES256), and an RSA public key is used to encrypt the bulk cipher key.

Note: If you want to encrypt and decrypt text in a payload, you must use the ENCRYPT and DECRYPT functions. For more information about these functions, see Encrypt and decrypt text.

The XML_ENCRYPT() and XML_DECRYPT() functions are not dependent on the encryption/decryption service that is used by the ENCRYPT and DECRYPT commands for text. The cipher method is specified explicitly as an argument to the XML_ENCRYPT() function. The `XML_DECRYPT()` function obtains the information about the specified cipher method from the `<xenc:EncryptedData>` element. Following are synopses of the XML encryption and decryption functions:

- `XML_ENCRYPT(<certKeyName>, <method> [, <flags>])**. Returns an <xenc:EncryptedData>` element that contains the encrypted input text and the encryption key, which is itself encrypted by using RSA.
- `XML_DECRYPT(<certKeyName>).` Returns the decrypted text from the input `<xenc:EncryptedData>` element, which includes the cipher method and the RSA-encrypted key.

Note: The <xenc:EncryptedData> element is defined in the W3C XML Encryption specification.

Following are descriptions of the arguments:

- **certKeyName:** Selects an X.509 certificate with an RSA public key for XML_ENCRYPT() or an RSA private key for XML_DECRYPT(). The certificate key must have been previously created by an `add ssl certKey` command.
- **method:** Specifies which cipher method to use for encrypting the XML data. Possible values: RC4, DES3, AES128, AES192, AES256.
- **flags:** A bitmask specifying the following optional key information (<ds:KeyInfo>) to be included in the <xenc:EncryptedData> element that is generated by `XML_ENCRYPT()`:
  - **1** - Include a KeyName element with the certKeyName. The element is <ds:KeyName>.
  - **2** - Include a KeyValue element with the RSA public key from the certificate. The element is <ds:KeyValue>.
  - **4** - Include an X509IssuerSerial element with the certificate serial number and issuer DN. The element is <ds:X509IssuserSerial>.
  - **8** - Include an X509SubjectName element with the certificate subject DN. The element is <ds:X509SubjectName>.
  - **16** - Include an X509Certificate element with the entire certificate. The element is <ds:X509Certificate>.

Use the XML_ENCRYPT() and XML_DECRYPT() functions in expressions

The XML encryption feature uses SSL certificate-key pairs to provide X.509 certificates (with RSA public keys) for key encryption and RSA private keys for key decryption. Therefore, before you use the XML_ENCRYPT() function in an expression, you must create an SSL certificate-key pair. The following command creates an SSL certificate-key pair, my-certkey, with the X.509 certificate, my-cert.pem, and the private key file, my-key.pem.

```
add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem -passcrypt
kxPeMRYnitY=
```

The following CLI commands create rewrite actions and policies for encrypting and decrypting XML content.

```
1 add rewrite action my-xml-encrypt-action replace "HTTP.RES.BODY(10000).
    XPATH_WITH_MARKUP(xp%/%)" "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp
    %/%).XML_ENCRYPT("my-certkey", AES256, 31)" -bypassSafetyCheck YES
2
3 add rewrite action my-xml-decrypt-action replace "HTTP.REQ.BODY(10000).
    XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)" "HTTP.REQ.BODY(10000).
    XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT("my-certkey"
    )" -bypassSafetyCheck YES
```



```
4
5 add rewrite policy my-xml-encrypt-policy "HTTP.REQ.URL.CONTAINS("xml-
     encrypt")" my-xml-encrypt-action
6
7 add rewrite policy my-xml-decrypt-policy "HTTP.REQ.BODY(10000).XPATH(xp
     %boolean(//xenc:EncryptedData)%)" my-xml-decrypt-action
8
9 bind rewrite global my-xml-encrypt-policy 30
10
11 bind rewrite global my-xml-decrypt-policy 30
12 <!--NeedCopy-->
```

In the above example, the rewrite action my-xml-encrypt-action encrypts the entire XML document (XPATH_WITH_MARKUP(xp%/%)) in the request by using the AES-256 bulk encryption method and the RSA public key from my-certkey to encrypt the bulk encryption key. The action replaces the document with an <xenc:EncryptedData> element containing the encrypted data and an encrypted key. The flags represented by 31 include all of the optional <ds:KeyInfo> elements.

The action my-xml-decrypt-action decrypts the first <xenc:EncryptedData> element in the response (XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)). This requires the prior addition of the xenc XML namespace by use of the following CLI command:

```
add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc##
```

The my-xml-decrypt-action action uses the RSA private key in my-certkey to decrypt the encrypted key and then uses the bulk encryption method specified in the element to decrypt the encrypted contents. Finally, the action replaces the encrypted data element with the decrypted content.

The rewrite policy my-xml-encrypt-policy applies my-xml-encrypt-action to requests for URLs containing xml-encrypt. The action encrypts the entire response from a service configured on the Citrix ADC appliance.

The rewrite policy my-xml-decrypt-policy applies my-xml-decrypt-action to requests that contain an <xenc:EncryptedData> element ((XPATH(xp%//xenc:EncryptedData%) returns a non-empty string). The action decrypts the encrypted data in requests that are bound for a service configured on the Citrix ADC appliance.

Advanced policy expressions: parsing SSL

September 14, 2021

There are advanced policy expressions to parse SSL certificates and SSL client hello messages.

Parse SSL certificates

You can use advanced policy expressions to evaluate X.509 Secure Sockets Layer (SSL) client certificates. A client certificate is an electronic document that can be used to authenticate a user's identity. A client certificate contains (at a minimum) version information, a serial number, a signature algorithm ID, an issuer name, a validity period, a subject (user) name, a public key, and signatures.

You can examine both SSL connections and data in client certificates. For example, you may want to send SSL requests that use low-strength ciphers to a particular load balancing virtual server farm. The following command is an example of a Content Switching policy that parses the cipher strength in a request and matches cipher strengths that are less than or equal to 40:

```
1 add cs policy p1 -rule "client.ssl.cipher_bits.le(40)"
2 <!--NeedCopy-->
```

As another example, you can configure a policy that determines whether a request contains a client certificate:

```
1 add cs policy p2 -rule "client.ssl.client_cert exists"
2 <!--NeedCopy-->
```

Or, you can configure a policy that examines particular information in a client certificate. For example, the following policy verifies that the certificate has one or more days before expiration:

```
1 add cs policy p2 -rule "client.ssl.client_cert exists && client.ssl.
     client_cert.days_to_expire.ge(1)"
2 <!--NeedCopy-->
```

> **Note**
>
> For information on parsing dates and times in a certificate, see Format of Dates and Times in an Expression and Expressions for SSL Certificate Dates.

Prefixes for text-based SSL and certificate data

The following table describes expression prefixes that identify text-based items in SSL transactions and client certificates.

Table 1. Prefixes That Return Text or Boolean Values for SSL and Client Certificate Data

Prefix	Description
CLIENT.SSL.CLIENT_CERT	Returns the SSL client certificate in the current SSL transaction.

Prefix	Description
CLIENT.SSL.CLIENT_CERT.TO_PEM	Returns the SSL client certificate in binary format.
CLIENT.SSL.CIPHER_EXPORTABLE	Returns a Boolean TRUE if the SSL cryptographic SSL cryptographic cipher is exportable.
CLIENT.SSL.CIPHER_NAME	Returns the name of the SSL Cipher if invoked from an SSL connection, and a NULL string if invoked from a non-SSL connection.
CLIENT.SSL.IS_SSL	Returns a Boolean TRUE if the current connection is SSL-based.

Prefixes for numeric data in SSL certificates

The following table describes prefixes that evaluate numeric data other than dates in SSL certificates. These prefixes can be used with the operations that are described in Basic Operations on Expression Prefixes and Compound Operations for Numbers.

Table 2. Prefixes That Evaluate Numeric Data Other Than Dates in SSL Certificates

Prefix	Description
CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE	Returns the number of days that the certificate is valid, or returns -1 for expired certificates.
CLIENT.SSL.CLIENT_CERT.PK_SIZE	Returns the size of the public key used in the certificate.
CLIENT.SSL.CLIENT_CERT.VERSION	Returns the version number of the certificate. If the connection is not SSL-based, returns zero (0).
CLIENT.SSL.CIPHER_BITS	Returns the number of bits in the cryptographic key. Returns 0 if the connection is not SSL-based.

Prefix	Description
CLIENT.SSL.VERSION	Returns a number that represents the SSL protocol version, as follows: 0. The transaction is not SSL-based.; 0x002. The transaction is SSLv2.; 0x300. The transaction is SSLv3.; 0x301. The transaction is TLSv1.; 0x302. The transaction is TLS 1.1.; 0x303. The transaction is TLS 1.2; 0x304. The transaction is TLS 1.3.

> **Note**
>
> For expressions related to expiration dates in a certificate, see Expressions for SSL Certificate Dates.

Expressions for SSL certificates

You can parse SSL certificates by configuring expressions that use the following prefix:

CLIENT.SSL.CLIENT_CERT

This section discusses the expressions that you can configure for certificates, except expressions that examine certificate expiration. Time-based operations are described in Advanced Policy Expressions: Working with Dates, Times, and Numbers.

The following table describes operations that you can specify for the CLIENT.SSL.CLIENT_CERT prefix.

Table 3. Operations That Can Be Specified with the CLIENT.SSL.CLIENT_CERT Prefix

SSL Certificate Operation	Description
`<certificate>.EXISTS`	Returns a Boolean TRUE if the client has an SSL certificate.
`<certificate>.ISSUER`	Returns the Distinguished Name (DN) of the Issuer in the certificate as a name-value list. An equals sign (“=”) is the delimiter for the name and the value, and the slash (“/”) is the delimiter that separates the name-value pairs. Following is an example of the returned DN: `/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.com`

SSL Certificate Operation	Description
`<certificate>.ISSUER.IGNORE_EMPTY_ELEMENTS`	Returns the Issuer and ignores the empty elements in a name-value list. For example, consider the following: Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com. The following Rewrite action returns a count of 6 based on the preceding Issuer definition: `sh rewrite action insert_ssl_header Name: insert_ssl Operation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT`. However, if you change the value to the following, the returned count is 9: `CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT`

Parse SSL client hello

You can parse the SSL client hello message by configuring expressions that use the following prefix:

Prefix	Description
CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXC(	Matches the hex code provided in the expression with the hex codes of cipher suites received in the client hello message.
CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION	Version received in the client hello message header.
CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE	Returns true if a client or server initiates session renegotiation.
CLIENT.SSL.CLIENT_HELLO.IS_REUSE	Returns true if the appliance reuses the SSL session based on the non-zero session-ID received in the client-hello message.

Prefix	Description
CLIENT.SSL.CLIENT_HELLO.IS_SCSV	Returns true if Signaling Cipher Suite Value (SCSV) capability is advertised in the client hello message. The hex code for fallback SCSV is 0x5600.
CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET	Returns true if session ticket extension with non-zero length is advertised in the client-hello message.
CLIENT.SSL.CLIENT_HELLO.LENGTH	Length received in the client hello message header.
CLIENT.SSL.CLIENT_HELLO.SNI	Returns the server name received in the Server Name extension of the client hello message.
CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPRC	Returns true if the application protocol in the ALPN extension received in the client hello message matches the protocol provided in the expression.

These expressions can be used at CLIENTHELLO_REQ bind point. For more information, see SSL policy binding.

Advanced policy expressions: IP and MAC addresses, throughput, VLAN IDs

September 14, 2021

You can use Advanced policy expression prefixes that return IPv4 and IPv6 addresses, MAC addresses, IP subnets, useful client and server data such as the throughput rates at the interface ports (Rx, Tx, and RxTx), and the IDs of the VLANs through which packets are received. You can then use various operators to evaluate the data that is returned by these expression prefixes.

Expressions for IP addresses and IP subnets

You can use Advanced policy expressions to evaluate addresses and subnets that are in Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) format. Expression prefixes for IPv6 addresses and subnets include IPv6 in the prefix. Expression prefixes for IPv4 addresses and subnets include IP

in the prefix. Following is an example of an expression that identifies whether a request has originated from a particular IPv4 subnet.

```
client.ip.src.in_subnet(147.1.0.0/16)
<!--NeedCopy-->
```

Following are two examples of Rewrite policies that examine the subnet from which the packet is received and perform a rewrite action on the Host header. With these two policies configured, the rewrite action that is performed depends on the subnet in the request. These two policies evaluate IP addresses that are in the IPv4 address format.

```
- add rewrite action URL1-rewrite-action replace "http.req.header(\"
   Host\")" "\"www.mycompany1.com\""
- add rewrite policy URL1-rewrite-policy "http.req.header(\"Host\").
   contains(\"www.test1.com\") && client.ip.src.in_subnet(147.1.0.0/16)
   " URL1-rewrite-action
- add rewrite action URL2-rewrite-action replace "http.req.header(\"
   Host\")" "\"www.mycompany2.com\""
- add rewrite policy URL2-rewrite-policy "http.req.header(\"Host\").
   contains(\"www.test2.com\") && client.ip.src.in_subnet
   (10.202.0.0/16)" URL2-rewrite-action
<!--NeedCopy-->
```

> **Note**
>
> The preceding examples are commands that you type at the Citrix ADC command-line interface (CLI) and, therefore, each quotation mark must be preceded by a backslash (\). For more information, see Configuring advanced policy expressions in a policy."

Prefixes for IPV4 addresses and IP subnets

The following table describes prefixes that return IPv4 addresses and subnets, and segments of IPv4 addresses. You can use numeric operators and operators that are specific to IPv4 addresses with these prefixes. For more information about numeric operations, see "Basic Operations on Expression Prefixes" and "Compound Operations for Numbers."

Table 1. Prefixes That Evaluate IP and MAC Addresses

Prefix	Description
CLIENT.IP.SRC	Returns the source IP of the current packet as an IP address or as a number.
CLIENT.IP.DST	Returns the destination IP of the current packet as an IP address or as a number.

Prefix	Description
SERVER.IP.SRC	Returns the source IP of the current packet as an IP address or as a number.
SERVER.IP.DST	Returns the destination IP of the current packet as an IP address or as a number.

Operations for IPV4 Addresses

The Prefix for IPV4 operations table describes the operators that can be used with prefixes that return an IPv4 address.

About IPv6 expressions

The IPv6 address format allows more flexibility than the older IPv4 format. IPv6 addresses are in the hexadecimal format (RFC 2373). In the following examples, Example 1 is an IPv6 address, Example 2 is a URL that includes the IPv6 address, and Example 3 includes the IPv6 address and a port number.

Example 1:

```
9901:0ab1:22a2:88a3:3333:4a4b:5555:6666
<!--NeedCopy-->
```

Example 2:

```
http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/
<!--NeedCopy-->
```

Example 3:

```
https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/
<!--NeedCopy-->
```

In Example 3, the brackets separate the IP address from the port number (8080).

Note that you can only use the '+' operator to combine IPv6 expressions with other expressions. The output is a concatenation of the string values that are returned from the individual expressions. You cannot use any other arithmetic operator with an IPv6 expression. The following syntax is an example:

```
client.ipv6.src + server.ip.dst
<!--NeedCopy-->
```


For example, if the client source IPv6 address is `ABCD:1234::ABCD`, and the server destination IPv4 address is `10.100.10.100`, the preceding expression returns `"ABCD:1234::ABCD10.100.10.100"`.

Note that when the Citrix ADC appliance receives an IPv6 packet, it assigns a temporary IPv4 address from an unused IPv4 address range and changes the source address of the packet to this temporary address. At response time, the outgoing packet's source address is replaced with the original IPv6 address.

> **Note**
>
> You can combine an IPv6 expression with any other expression except an expression that produces a Boolean result.

Expression prefixes for IPv6 addresses

The IPv6 addresses that are returned by the expression prefixes in the following table can be treated as text data. For example, the prefix client.ipv6.dst returns the destination IPv6 address as a string that can be evaluated as text.

The following table describes expression prefixes that return an IPv6 address.

Table 3. IPv6 Expression Prefixes That Return Text

Prefix	Description
CLIENT.IPV6	Operates on the IPv6 address in with the current packet.
CLIENT.IPV6.DST	Returns the IPv6 address in the destination field of the IP header.
CLIENT.IPV6.SRC	Returns the IPv6 address in the source field of the IP header. Following are examples: `client.ipv6.src.in_subnet(2007::2008/64)` `client.ipv6.src.get1.le(2008)`
SERVER.IPV6	Operates on the IPv6 address in with the current packet.
SERVER.IPV6.DST	Returns the IPv6 address in the destination field of the IP header.
SERVER.IPV6.SRC	Returns the IPv6 address in the source field of the IP header. Following are examples: `server.ipv6.src.in_subnet(2007::2008/64)` `server.ipv6.src.get1.le(2008)`

Operations for IPv6 prefixes

The following table describes the operators that can be used with prefixes that return an IPv6 address:

Table 4. Operations That Evaluate IPv6 Addresses

IPv6 Operation	Description
<ipv6>.EQ(<IPv6_address>	Returns a Boolean TRUE if the IP address value is same as the <IPv6_address> argument. Following is an example: client.ipv6.dst.eq(ABCD:1234::ABCD)
<ipv6>.GET1. . .GET8	Returns a segment of an IPv6 address as a number. The following example expressions retrieve segments from the ipv6 address 1000:1001:CD10:0000:0000:89AB:4567:CDEF: client.ipv6.dst.get5 extracts 0000, which is the fifth set of bits in the address. client.ipv6.dst.get6 extracts 89AB. client.ipv6.dst.get7 extracts 4567. You can perform numeric operations on these segments. Note that you cannot perform numeric operations when you retrieve an entire IPv6 address. This is because expressions that return an entire IPv6 address, such as CLIENT.IPV6.SRC, return the address in text format.
<ipv6>.IN_SUBNET(<subnet>)	Returns a Boolean TRUE if the IPv6 address value is in the subnet specified by the <subnet> argument. Following is an example: client.ipv6.dst.eq(1000:1001:CD10 :0000:0000:89AB:4567:CDEF/60)
<ipv6>.IS_IPV4	Returns a Boolean TRUE if this is an IPv4 client, and returns a Boolean FALSE if it is not.
<ipv6>.SUBNET(<n>)	Returns the IPv6 address after applying the subnet mask specified as the argument. The subnet mask can take values between 0 and 128. For example: CLIENT.IPV6.SRC.SUBNET(24)

Expressions for MAC addresses

A MAC address consists of colon-delimited hexadecimal values in the format ##:##:##:##:##:##, where each "#" represents either a number from 0 through 9 or a letter from A through F. Default syntax expression prefixes and operators are available for evaluating source and destination MAC addresses.

Prefixes for MAC addresses

The following table describes prefixes that return MAC addresses.

Table 5. Prefixes That Evaluate MAC Addresses

Prefix	Description
`client.ether.dstmac`	Returns the MAC address in the destination field of the Ethernet header.
`client.ether.srcmac`	Returns the MAC address in the source field of the Ethernet header.

Operations for MAC addresses

The following table describes the operators that can be used with prefixes that return a MAC address.

Table 6. Operations on MAC Addresses

Prefix	Description
`<mac address>.EQ(<address>)`	Returns a Boolean TRUE if the MAC address value is same as the `<address>` argument.
`<mac address>.GET1. . .GET4`	Returns a numeric value extracted from the segment of the MAC address that is specified in the GET operation. For example, if the MAC address is 12:34:56:78:9a:bc, the following returns 34: `client.ether.dstmac.get2`

Expressions for numeric client and server data

The following table describes prefixes for working with numeric client and server data, including throughput, port numbers, and VLAN IDs.

Table 7. Prefixes that evaluate numeric client and server data

Prefix	Description
client.interface.rxthroughput	Returns an integer representing the raw received traffic throughput in kilobytes per second (KBps) for the previous seven seconds.
client.interface.txthroughput	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
client.interface.rxtxthroughput	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
server.interface.rxthroughput	Returns an integer representing the raw received traffic throughput in KBps for the previous seven seconds.
server.interface.txthroughput	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
server.interface.rxtxthroughput	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
server.vlan.id	Returns a numeric ID of the VLAN through which the current packet entered the Citrix ADC.
client.vlan.id	Returns a numeric ID for the VLAN through which the current packet entered the Citrix ADC.

Advanced policy expressions: stream analytics functions

September 14, 2021

Stream Analytics expressions begin with the ANALYTICS.STREAM(<identifier_name>) prefix. The following list describes the functions that can be used with this prefix.

- **COLLECT_STATS**

  Collect statistical data from the requests that are evaluated against the policy and create a record for each request.

- **REQUESTS**

  Return the number of requests that exist for the specified record grouping. The value returned is of type unsigned long.

- **BANDWIDTH**

  Return the bandwidth statistic for the specified record grouping. The value returned is of type unsigned long.

- **RESPTIME**

  Return the response time statistic for the specified record grouping. The value returned is of type unsigned long.

- **CONNECTIONS**

  Return the number of concurrent connections that exist for the specified record grouping. The value returned is of type unsigned long.

- **IS_TOP(n)**

  Return a Boolean TRUE if the statistical value for the specified record grouping is one among the top n groups. Otherwise, return a Boolean FALSE.

- **CHECK_LIMIT**

  Return a Boolean TRUE if the statistic for the specified record grouping has hit the preconfigured limit. Otherwise, return a Boolean FALSE.

Advanced policy expressions: DataStream

September 14, 2021

The policy infrastructure on the Citrix ADC appliance includes expressions that you can use to evaluate and process database server traffic when the appliance is deployed between a farm of application servers and their associated database servers.

This topic includes the following sections:

- Expressions for the MySQL Protocol
- Expressions for Evaluating Microsoft SQL Server Connections

Expressions for the MySQL protocol

The following expressions evaluate traffic associated with MySQL database servers. You can use the request-based expressions (expressions that begin with MYSQL.CLIENT and MYSQL.REQ) in policies to

make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

- **MYSQL.CLIENT.** Operates on the client properties of a MySQL connection.
- **MYSQL.CLIENT.CAPABILITIES.** Returns the set of flags that the client has set in the capabilities field of the handshake initialization packet during authentication. Examples of the flags that are set are CLIENT_FOUND_ROWS, CLIENT_COMPRESS, and CLIENT_SSL.
- **MYSQL.CLIENT.CHAR_SET.** Returns the enumeration constant assigned to the character set that the client uses. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the character set enumeration constants:
  - LATIN2_CZECH_CS
  - DEC8_SWEDISH_CI
  - CP850_GENERAL_CI
  - GREEK_GENERAL_CI
  - LATIN1_GERMAN1_CI
  - HP8_ENGLISH_CI
  - KOI8R_GENERAL_CI
  - LATIN1_SWEDISH_CI
  - LATIN2_GENERAL_CI
  - SWE7_SWEDISH_CI
  - ASCII_GENERAL_CI
  - CP1251_BULGARIAN_CI
  - LATIN1_DANISH_CI
  - HEBREW_GENERAL_CI
  - LATIN7_ESTONIAN_CS
  - LATIN2_HUNGARIAN_CI
  - KOI8U_GENERAL_CI
  - CP1251_UKRAINIAN_CI
  - CP1250_GENERAL_CI
  - LATIN2_CROATIAN_CI
  - CP1257_LITHUANIAN_CI
  - LATIN5_TURKISH_CI
  - LATIN1_GERMAN2_CI
  - ARMSCII8_GENERAL_CI
  - UTF8_GENERAL_CI
  - CP1250_CZECH_CS
  - CP866_GENERAL_CI

- KEYBCS2_GENERAL_CI
- MACCE_GENERAL_CI
- MACROMAN_GENERAL_CI
- CP852_GENERAL_CI
- LATIN7_GENERAL_CI
- LATIN7_GENERAL_CS
- MACCE_BIN
- CP1250_CROATIAN_CI
- LATIN1_BIN
- LATIN1_GENERAL_CI
- LATIN1_GENERAL_CS
- CP1251_BIN
- CP1251_GENERAL_CI
- CP1251_GENERAL_CS
- MACROMAN_BIN
- CP1256_GENERAL_CI
- CP1257_BIN
- CP1257_GENERAL_CI
- ARMSCII8_BIN
- ASCII_BIN
- CP1250_BIN
- CP1256_BIN
- CP866_BIN
- DEC8_BIN
- GREEK_BIN
- HEBREW_BIN
- HP8_BIN
- KEYBCS2_BIN
- KOI8R_BIN
- KOI8U_BIN
- LATIN2_BIN
- LATIN5_BIN
- LATIN7_BIN
- CP850_BIN
- CP852_BIN
- SWE7_BIN
- UTF8_BIN
- GEOSTD8_GENERAL_CI
- GEOSTD8_BIN

  - LATIN1_SPANISH_CI
  - UTF8_UNICODE_CI
  - UTF8_ICELANDIC_CI
  - UTF8_LATVIAN_CI
  - UTF8_ROMANIAN_CI
  - UTF8_SLOVENIAN_CI
  - UTF8_POLISH_CI
  - UTF8_ESTONIAN_CI
  - UTF8_SPANISH_CI
  - UTF8_SWEDISH_CI
  - UTF8_TURKISH_CI
  - UTF8_CZECH_CI
  - UTF8_DANISH_CI
  - UTF8_LITHUANIAN_CI
  - UTF8_SLOVAK_CI
  - UTF8_SPANISH2_CI
  - UTF8_ROMAN_CI
  - UTF8_PERSIAN_CI
  - UTF8_ESPERANTO_CI
  - UTF8_HUNGARIAN_CI
  - INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE.** Returns the name of the database specified in the authentication packet that the client sends to the database server. This is the databasename attribute.
- MYSQL.CLIENT.USER. Returns the user name (in the authentication packet) with which the client is attempting to connect to the database. This is the user attribute.
- **MYSQL.REQ.** Operates on a MySQL request.
- **MYSQL.REQ.COMMAND.** Identifies the enumeration constant assigned to the type of command in the request. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the enumeration constant values:

  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST
  - CREATE_DB
  - DROP_DB
  - REFRESH

  - SHUTDOWN
  - STATISTICS
  - PROCESS_INFO
  - CONNECT
  - PROCESS_KILL
  - DEBUG
  - PING
  - TIME
  - DELAYED_INSERT
  - CHANGE_USER
  - BINLOG_DUMP
  - TABLE_DUMP
  - CONNECT_OUT
  - REGISTER_SLAVE
  - STMT_PREPARE
  - STMT_EXECUTE
  - STMT_SEND_LONG_DATA
  - STMT_CLOSE
  - STMT_RESET
  - SET_OPTION
  - STMT_FETCH

- **MYSQL.REQ.QUERY.** Identifies the query in the MySQL request.
- **MYSQL.REQ.QUERY.COMMAND.** Returns the first keyword in the MySQL query.
- **MYSQL.REQ.QUERY.SIZE.** Returns the size of the request query in integer format. The SIZE method is similar to the CONTENT_LENGTH method that returns the length of an HTTP request or response.
- **MYSQL.REQ.QUERY.TEXT.** Returns a string covering the entire query.
- **MYSQL.REQ.QUERY.TEXT(<n>).** Returns the first n bytes of the MySQL query as a string. This is similar to HTTP.BODY(<n>).

  Parameters:

  n - Number of bytes to be returned

- **MYSQL.RES.** Operates on a MySQL response.
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** Checks whether the response has at least i number of rows and returns a Boolean TRUE or FALSE to indicate the result.

  Parameters:

  i - Number of rows

- **MYSQL.RES.ERROR.** Identifies the MySQL error object. The error object includes the error number and the error message.
- **MYSQL.RES.ERROR.MESSAGE.** Returns the error message that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.NUM.** Returns the error number that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.SQLSTATE.** Returns the value of the SQLSTATE field in the server's error response. The MySQL server translates error number values to SQLSTATE values.
- **MYSQL.RES.FIELD(<i>).** Identifies the packet that corresponds to the ith individual field in the server's response. Each field packet describes the properties of the associated column. The packet count (i) begins at 0.

  Parameters:

  i - Packet number
- **MYSQL.RES.FIELD(<i>).CATALOG.** Returns the catalog property of the field packet.
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** Returns the character set of the column. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.
- **MYSQL.RES.FIELD(<i>).DATATYPE.** Returns an enumeration constant that represents the data type of the column. This is the type (also called enum_field_type) attribute of the column. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. The possible values for the various data types are:
  - DECIMAL
  - TINY
  - SHORT
  - LONG
  - FLOAT
  - DOUBLE
  - NULL
  - TIMESTAMP
  - LONGLONG
  - INT24
  - DATE
  - TIME
  - DATETIME
  - YEAR
  - NEWDATE

  - VARCHAR (new in MySQL 5.0)
  - BIT (new in MySQL 5.0)
  - NEWDECIMAL (new in MySQL 5.0)
  - ENUM
  - SET
  - TINY_BLOB
  - MEDIUM_BLOB
  - LONG_BLOB
  - BLOB
  - VAR_STRING
  - STRING
  - GEOMETRY

- **MYSQL.RES.FIELD(<i>).DB.** Returns the database identifier (db) attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).DECIMALS.** Returns the number of positions after the decimal point if the type is DECIMAL or NUMERIC. This is the decimals attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).FLAGS.** Returns the flags property of the field packet. Following are the possible hexadecimal flag values:
  - 0001: NOT_NULL_FLAG
  - 0002: PRI_KEY_FLAG
  - 0004: UNIQUE_KEY_FLAG
  - 0008: MULTIPLE_KEY_FLAG
  - 0010: BLOB_FLAG
  - 0020: UNSIGNED_FLAG
  - 0040: ZEROFILL_FLAG
  - 0080: BINARY_FLAG
  - 0100: ENUM_FLAG
  - 0200: AUTO_INCREMENT_FLAG
  - 0400: TIMESTAMP_FLAG
  - 0800: SET_FLAG
- **MYSQL.RES.FIELD(<i>).LENGTH.** Returns the length of the column. This is the value of the length attribute of the field packet. The value that is returned might be larger than the actual value. For example, an instance of a VARCHAR(2) column might return a value of 2 even when it contains only one character.
- **MYSQL.RES.FIELD(<i>).NAME.** Returns the column identifier (the name after the AS clause, if any). This is the name attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** Returns the original column identifier (before the AS clause, if any). This is the org_name attribute of the field packet.

- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** Returns the original table identifier of the column (before the AS clause, if any). This is the org_table attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).TABLE.** Returns the table identifier of the column (after the AS clause, if any). This is the table attribute of the field packet.
- **MYSQL.RES.FIELDS_COUNT.** Returns the number of field packets in the response (the field_count attribute of the OK packet).
- **MYSQL.RES.OK.** Identifies the OK packet sent by the database server.
- **MYSQL.RES.OK.AFFECTED_ROWS.** Returns the number of rows affected by an INSERT, UPDATE, or DELETE query. This is the value of the affected_rows attribute of the OK packet.
- **MYSQL.RES.OK.INSERT_ID.** Identifies the unique_id attribute of the OK packet. If an auto-increment identity is not generated by the current MySQL statement or query, the value of unique_id, and hence the value returned by the expression, is 0.
- **MYSQL.RES.OK.MESSAGE.** Returns the message property of the OK packet.
- **MYSQL.RES.OK.STATUS.** Identifies the bit string in the server_status attribute of the OK packet. Clients can use the server status to check whether the current command is a part of a running transaction. The bits in the server_status bit string correspond to the following fields (in the given order):
  - IN TRANSACTION
  - AUTO_COMMIT
  - MORE RESULTS
  - MULTI QUERY
  - BAD INDEX USED
  - NO INDEX USED
  - CURSOR EXISTS
  - LAST ROW SEEN
  - DATABASE DROPPED
  - NO BACKSLASH ESCAPES
- **MYSQL.RES.OK.WARNING_COUNT.** Returns the warning_count attribute of the OK packet.
- **MYSQL.RES.ROW(<i>).** Identifies the packet that corresponds to the ith individual row in the database server's response.

  Parameters:

  i - Row number
- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** Checks whether the jth column of the ith row of the table is NULL. Following C conventions, both indexes i and j

start from 0. Therefore, row i and column j are actually the (i+1)th row and the (j+1)th column, respectively.

Parameters:

i - Row number

j - Column number

- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** Checks whether the jth column of the ith row of the table is NULL. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the (i+1)th row and the (j+1)th column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).** Returns an integer value from the jth column of the ith row of the table. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the (i+1)th row and the (j+1)th column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.ROW(<i>).TEXT_ELEM(j).** Returns a string from the jth column of the ith row of the table. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the (i+1)th row and the (j+1)th column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.TYPE.** Returns an enumeration constant for the response type. Its values can be ERROR, OK, and RESULT_SET. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.

Expressions for evaluating Microsoft SQL server connections

The following expressions evaluate traffic associated with Microsoft SQL Server database servers. You can use the request-based expressions (expressions that begin with MSSQL.CLIENT and MSSQL.REQ)

in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MSSQL.RES) to evaluate server responses to user-configured health monitors.

Expression	Description
MSSQL.CLIENT.CAPABILITIES	Returns the OptionFlags1, OptionFlags2, OptionFlags3, and TypeFlags fields of the LOGIN7authentication packet, in that order, as a 4-byte integer. Each field is 1 byte long and specifies a set of client capabilities.
MSSQL.CLIENT.DATABASE	Returns the name of the client database. The value returned is of type text.
MSSQL.CLIENT.USER	Returns the user name with which the client authenticated. The value returned is of type text.
MSSQL.REQ.COMMAND	Returns an enumeration constant that identifies the type of command in the request sent to a Microsoft SQL Server database server. The value returned is of type text. Examples of the values of the enumeration constant are QUERY, RESPONSE, RPC, and ATTENTION. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.
MSSQL.REQ.QUERY.COMMAND	Returns the first keyword in the SQL query. The value returned is of type text.
MSSQL.REQ.QUERY.SIZE	Returns the size of the SQL query in the request. The value returned is a number.
MSSQL.REQ.QUERY.TEXT	Returns the entire SQL query as a string. The value returned is of type text.
MSSQL.REQ.QUERY.TEXT(<n>)	Returns the first n bytes of the SQL query. The value returned is of type text. Parameters: n - Number of bytes
MSSQL.REQ.RPC.NAME	Returns the name of the procedure that is being called in a remote procedure call (RPC) request. The name is returned as a string.

Expression	Description
MSSQL.REQ.RPC.IS_PROCID	Returns a Boolean value that indicates whether the remote procedure call (RPC) request contains a procedure ID or an RPC name. A return value of TRUEindicates that the request contains a procedure ID and a return value of FALSE indicates that the request contains an RPC name.
MSSQL.REQ.RPC.PROCID	Returns the procedure ID of the remote procedure call (RPC) request as an integer.
MSSQL.REQ.RPC.BODY **Note:** Not available for releases before 10.1.	Returns the body of the SQL request as a string in the form of parameters represented as “a=b” clauses separated by commas, where “a” is the RPC parameter name and “b” is its value.
MSSQL.REQ.RPC.BODY(n) **Note:** Not available for releases before 10.1.	Returns part of the body of the SQL request as a string in the form of parameters represented as “a=b” clauses separated by commas, where “a” is the RPC parameter name and “b” is its value. Parameters are returned from only the first “n” bytes of the request, skipping the SQL header. Only complete name-value pairs are returned.
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	Checks whether the response has at least i number of rows. The value returned is a Boolean TRUE or FALSEvalue. Parameters: i - Number of rows
MSSQL.RES.DONE.ROWCOUNT	Returns a count of the number of rows affected by an INSERT, UPDATE, or DELETE query. The value returned is of type unsigned long.
MSSQL.RES.DONE.STATUS	Returns the status field from the DONE token sent by a Microsoft SQL Server database server. The value returned is a number.
MSSQL.RES.ERROR.MESSAGE	Returns the error message from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the MsgText field in the ERROR token. The value returned is of type text.

Expression	Description
MSSQL.RES.ERROR.NUM	Returns the error number from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the Number field in the ERROR token. The value returned is a number.
MSSQL.RES.ERROR.STATE	Returns the error state from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the State field in the ERROR token. The value returned is a number.
MSSQL.RES.FIELD(<i>).DATATYPE	Returns the data type of the ith field in the server response. The EQ(<m>) and NE(<m>) functions, which return Boolean values to indicate the result of a comparison, are used with this prefix. For example, the following expression returns a Boolean TRUE if the DATATYPE function returns a value of datetime for the third field in the response: MSSQL.RES.FIELD(<2>).DATATYPE.EQ(datetime) Parameters: i - Row number
MSSQL.RES.FIELD(<i>).LENGTH	Returns the maximum possible length of the ith field in the server response. The value returned is a number. Parameters: i - Row number
MSSQL.RES.FIELD(<i>).NAME	Returns the name of the ith field in the server response. The value returned is of type text. Parameters: i - Row number
MSSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>)	Returns a value of type double from the jth column of the ith row of the table. If the value is not a double value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number

Expression	Description
MSSQL.RES.ROW(<i>).NUM_ELEM(j)	Returns an integer value from the jth column of ith row of the table. If the value is not an integer value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number
MSSQL.RES.ROW(<i>).IS_NULL_ELEM(j)	Checks whether the jth column of the ith row of the table is NULL and returns a Boolean TRUE or FALSE to indicate the result. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row iand column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number
MSSQL.RES.ROW(<i>).TEXT_ELEM(j)	Returns a text string from the jth column of ith row of the table. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number
MSSQL.RES.TYPE	Returns an enumeration constant that identifies the response type. Following are the possible return values: ERROR, OK, and RESULT_SET. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.

Typecasting data

September 14, 2021

You can extract data of one type (for example, text or an integer) from requests and responses and

transform it to data of another type. For example, you can extract a string and transform the string to time format. You can also extract a string from an HTTP request body and treat it like an HTTP header or extract a value from one type of request header and insert it in a response header of a different type.

After typecasting the data, you can apply any operation that is appropriate for the new data type. For example, if you typecast text to an HTTP header, you can apply any operation that is applicable to HTTP headers to the returned value.

For more information about typecasting data, see the Typecasting Operations pdf.

Regular Expressions

September 14, 2021

When you want to perform string matching operations that are more complex than the operations that you perform with the CONTAINS("<string>") or EQ("<string>") operators, you use regular expressions. The policy infrastructure on the Citrix® Citrix ADC® appliance includes operators to which you can pass regular expressions as arguments for text matching. The names of the operators that work with regular expressions include the string REGEX. The regular expressions that you pass as arguments must conform to the regular expression syntax that is described in "<http://www.pcre.org/pcre.txt>." You can learn more about regular expressions at "<http://www.regular-expressions.info/quickstart.html>" and at "<http://www.silverstones.com/thebat/Regex.html>."

The target text for an operator that works with regular expressions can be either text or the value of an HTTP header. Following is the format of a default syntax expression that uses a regular expression operator to operate on text:

```
<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)
```

The string <text> represents the default syntax expression prefix that identifies a text string in a packet (for example, HTTP.REQ.URL). The string <regex_operator> represents the regular expression operator. The regular expression always begins with the string re. A pair of matching delimiters, represented by <delimiter>, enclose the string <regex_pattern>, which represents the regular expression.

The following example expression checks whether the URL in an HTTP packet contains the string *.jpeg (where * is a wildcard) and returns a Boolean TRUE or FALSE to indicate the result. The regular expression is enclosed within a pair of slash marks (/), which act as delimiters.

```
http.req.url.regex_match(re/.<asterisk>\.jpeg/)
```


Regular expression operators can be combined to define or refine the scope of a search. For example, `<text>.AFTER_REGEX(reregex_pattern1).BEFORE_REGEX(reregex_pattern2)` specifies that the target for string matching is the text between the patterns regex_pattern1 and regex_pattern2. You can use a text operator on the scope that is defined by the regular expression operators. For example, you can use the CONTAINS("<string>") operator to check whether the defined scope contains the string abc:

```
<text>.AFTER_REGEX(re/regex_pattern1).BEFORE_REGEX(re/regex_pattern2/).
CONTAINS("abc")
```

> **Note**
>
> The process of evaluating a regular expression inherently takes more time than that for an operator such as CONTAINS ("<string>") or EQ("<string>"), which work with simple string arguments. You should use regular expressions only if your requirement is beyond the scope of other operators.

Basic characteristics of regular expressions

September 14, 2021

Following are notable characteristics of regular expressions as defined on the Citrix ADC appliance:

- A regular expression always begins with the string “re” followed by a pair of delimiting characters (called delimiters) that enclose the regular expression that you want to use.

For example, re#<regex_pattern># uses the number sign (#) as a delimiter.

- A regular expression cannot exceed 1499 characters.
- Digit matching can be done by using the string \d (a backslash followed by d).
- White space can be represented by using \s (a backslash followed by s).
- A regular expression can contain white spaces.

Following are the differences between the Citrix ADC syntax and the PCRE syntax:

- The Citrix ADC does not allow back references in regular expressions.
- You should not use recursive regular expressions.
- The dot meta-character also matches the newline character.
- Unicode is not supported.
- The operation SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option in the regular expression.

Operations for regular expressions

September 14, 2021

The following table describes the operators that work with regular expressions. The operation performed by a regular expression operator in a given default syntax expression depends on whether the expression prefix identifies text or HTTP headers. Operations that evaluate headers override any text-based operations for all instances of the specified header type. When you use an operator, replace <text> with the default syntax expression prefix that you want to configure for identifying text.

Regular Expression Operation	Description
<text>.BEFORE_REGEX(<regular expression>)	Selects the text that precedes the string that matches the <regular expression> argument. If the regular expression does not match any data in the target, the expression returns a text object of length 0. The following expression selects the string “text” from “text/plain”. http.res.header(“content-type”).before_regex(re#/#)
<text>.AFTER_REGEX(<regular expression>)	Selects the text that follows the string that matches the <regular expression> argument. If the regular expression does not match any text in the target, the expression returns a text object of length 0. The following expression extracts “Example” from “myExample”: http.req.header(“etag”).after_regex(re/my/)
<text>.REGEX_SELECT(<regular expression>)	Selects a string that matches the <regular expression> argument. If the regular expression does not match the target, a text object of length 0 is returned. The following example extracts the string “NS-CACHE-9.0: 90” from a Via header: http.req.header(“via”).regex_select(re!NS-CACHE-\d.\d:\s*\d{1,3}!)

Regular Expression Operation	Description
<text>.REGEX_MATCH(<regular expression>)	Returns TRUE if the target matches a <regular expression> argument of up to 1499 characters. The regular expression must be of the following format: re<delimiter>regular expression< delimiter> Both delimiters must be the same. Additionally, the regular expression must conform to the Perl-compatible (PCRE) regular expression library syntax. For more information, go to http://www.pcre.org/pcre.txt. In particular, see the pcrepattern man page. However, note the following: Back-references are not allowed. Recursive regular expressions are not recommended. The dot metacharacter also matches the newline character. The Unicode character set is not supported. SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option specified in the regular expression. The following are examples: http.req.hostname.regex_match(re/[[:alpha:]]+(abc){2,3}/) and http.req.url.set_text_mode(urlencoded).regex_match(re#(ab The following example matches ab and aB: http.req.url.regex_match(re/a(?i)b/) The following example matches ab, aB, Ab and AB: http.req.url.set_text_mode(ignorecase).regex_match(re/ab/) The following example performs a case-insensitive, multiline match in which the dot meta-character also matches a newline character: http.req.body.regex_match(re/(?ixm) (^ab (.*) cd$) /)

Configuring classic policies and expressions

September 14, 2021

Some Citrix ADC features use classic policies and classic expressions. As with default syntax policies, classic policies can be either global or specific to a virtual server. However, to a certain extent, the configuration method and bind points for classic policies are different from those of default syntax policies. As with default syntax expressions, you can configure named expressions and use a named expression in multiple classic policies.

The following table summarizes Citrix ADC features that can be configured by using classic policies.

Click here to view the table.

Configure a classic policy

September 14, 2021

You can configure classic policies and classic expressions by using either the configuration utility or the command-line interface. A policy rule cannot exceed 1,499 characters. When configuring the policy rule, you can use named classic expressions. For more information about named expressions, see Create named classic expressions. After configuring the policy, you bind it either globally or to a virtual server.

Note that there are small variations in the policy configuration methods for various Citrix ADC features.

Note: You can embed a classic expression in a default syntax expression by using the syntax SYS.EVAL_CLASSIC_EXPR(classic_expression), specifying the classic_expression as the argument.

Create a classic policy by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  add cmp policy <name> -rule <expression> -action <action>
2
3 -  show cmp policy [<policyName>]
4 <!--NeedCopy-->
```


Example

The following commands first create a compression action and then create a compression policy that applies the action:

```
> add cmp action cmp-act-compress compress
 Done
> show cmp action cmp-act-compress
1)      Name: cmp-act-compress  Compression Type: compress
 Done
> add cmp pol cmp-pol-compress -rule ExpCheckIp -resAction cmp-act-
   compress
 Done
> show cmp pol cmp-pol-compress
1)      Name: cmp-pol-compress  Rule: ExpCheckIp
        Response action: cmp-act-compress       Hits: 0
 Done
>
<!--NeedCopy-->
```

Create a policy with classic expressions by using the GUI

1. In the navigation pane, expand the feature for which you want to configure a policy and, depending on the feature, do the following:
   - For Content Switching, Cache Redirection, and the application firewall, click **Policies**.
   - For SSL, click Policies, and then in the details pane, click the **Policies** tab.
   - For **System Authentication**, click **Authentication**, and then in the details pane, click the **Policies** tab.
   - For Filter, SureConnect, and Priority Queuing, expand Protection Features, select the desired function, and then in the details pane, click the **Policies** tab.
   - For the Citrix Gateway, expand Citrix Gateway, expand Policies, select the desired function, and then in the details pane, click the **Policies** tab.
2. For most features, click the **Add** button.
3. In the **Create** <feature name> Policy dialog box, in the Name* text box, enter a name for the policy.

   Note: You must begin a policy name with a letter or underscore. A policy name can consist of 1 to 31 characters, including letters, numbers, hyphen (-), period (.), pound sign (#), space ( ), and underscore (_).
4. For most features, you associate an action or a profile. For example, you may be required to select an action, or, in the case of an Citrix Gateway or application firewall policy, you select a

profile to associate with the policy. A profile is a set of configuration options that operate as a set of actions that are applied when the data being analyzed matches the policy rule.

5. Create an expression that describes the type of data that you want this policy to match.

   Depending on the type of policy you want to create, you can choose a predefined expression, or you can create a new expression.

   Named expressions are predefined expressions that you can reference by name in a policy rule.

6. Click **Create** to create your new policy.
7. Click **Close** to return to the Policies screen for the type of policy you were creating.

Configure a classic expression

September 14, 2021

Classic expressions consist of the following expression elements, listed in hierarchical order:

- **Flow Type.** Specifies whether the connection is incoming or outgoing. The flow type is REQ for incoming connections and RES for outgoing connections.
- **Protocol.** Specifies the protocol, the choices for which are HTTP, SSL, TCP, and IP.
- **Qualifier.** The protocol attribute, which depends on the selected protocol.
- **Operator.** The type of test you want to perform on the connection data. Your choice of operator depends upon the connection information you are testing. If the connection information you are testing is text, you use text operators. If it is a number, you use standard numeric operators.
- **Value.** The string or number against which the connection data element—defined by the flow type, protocol, and qualifier—is tested. The value can be either a literal or an expression. The literal or expression must match the data type of the connection data element.

In a policy, classic expressions can be combined to create more complex expressions using Boolean and comparative operators.

Expression elements are parsed from left to right. The leftmost element is either REQ or RES and designates a request or a response, respectively. Successive terms define a specific connection type and a specific attribute for that connection type. Each term is separated from any preceding or following term by a period. Arguments appear in parentheses and follow the expression element to which they are passed.

The following classic expression fragment returns the client source IP for an incoming connection.

```
REQ.IP.SOURCEIP
```

The example identifies an IP address in a request. The expression element SOURCEIP designates the source IP address. This expression fragment may not be useful by itself. You can use an additional

expression element, an operator, to determine whether the returned value meets specific criteria. The following expression tests whether the client IP is in the subnet 200.0.0.0/8 and returns a Boolean TRUE or FALSE:

```
REQ.IP.SOURCEIP == 200.0.0.0 -netmask 255.0.0.0
```

Create a classic policy expression by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  set appfw policy \<name\> -rule \<expression\> -action \<action\>
2
3 -  show appfw policy \<name\>
4 <!--NeedCopy-->
```

Example

```
1  > set appfw policy GenericApplicationSSL_ 'HTTP.REQ.METHOD.EQ("get")'
      APPFW_DROP
2   Done
3  > show appfw policy GenericApplicationSSL_
4          Name: GenericApplicationSSL_    Rule: HTTP.REQ.METHOD.EQ("get")
5          Profile: APPFW_DROP     Hits: 0
6          Undef Hits: 0
7          Policy is bound to following entities
8          1) REQ VSERVER app_u_GenericApplicationSSLPortalPages
              PRIORITY : 100
9   Done
10 <!--NeedCopy-->
```

Add an expression for a classic policy by using the GUI

This procedure documents the Add Expression dialog box. Depending on the feature for which you are configuring a policy, the route by which you arrive at this dialog box may be different.

1. Perform steps 1-4 in “To create a policy with classic expressions by using the GUI”.
2. In the **Add Expression** dialog box, in Expression Type, click the type of expression you want to create.
3. Under **Flow Type**, click the down arrow and choose a flow type.

The flow type is typically REQ or RES. The REQ option specifies that the policy applies to all incoming connections or requests. The RES option applies the policy to all outgoing connections or responses.

For Application Firewall policies, you should leave the expression type set to General Expression, and the flow type set to REQ. The Application Firewall treats each request and response as a single paired entity, so all Application Firewall policies begin with REQ.

1. Under Protocol, click the down arrow and choose the protocol you want for your policy expression. Your choices are:
   - HTTP. Evaluates HTTP requests that are sent to a Web server. For classic expressions, HTTP includes HTTPS requests.
   - SSL. Evaluates SSL data associated with the current connection.
   - TCP. Evaluates the TCP data associated with the current connection.
   - IP. Evaluates the IP addresses associated with the current connection.
2. Under Qualifier, click the down arrow and choose a qualifier for your policy.
   The qualifier defines the type of data to be evaluated. The list of qualifiers that appears depends on which protocol you selected in step 4.
   The following choices appear for the HTTP protocol:
   - METHOD. Filters HTTP requests that use a particular HTTP method.
   - URL. Filters HTTP requests for a specific Web page.
   - URLQUERY. Filters HTTP requests that contain a particular query string.
   - VERSION. Filters HTTP requests on the basis of the specified HTTP protocol version.
   - HEADER. Filters on the basis of a particular HTTP header.
   - URLLEN. Filters on the basis of the length of the URL.
   - URLQUERY. Filters on the basis of the query portion of the URL.
   - URLQUERYLEN. Filters on the basis of the length of the query portion of the URL only.
3. Under Operator, click the down arrow and choose the operator for your policy expression. Some common operators are:

Operator	Description
==	Matches the specified value exactly or is exactly equal to the specified value.
!=	Does not match the specified value.
>	Is greater than the specified value.
<	Is less than the specified value.
>=	Is greater than or equal to the specified value.
<=	Is less than or equal to the specified value.
CONTAINS	Contains the specified value.

Operator	Description
CONTENTS	Returns the contents of the designated header, URL, or URL query.
EXISTS	The specified header or query exists.
NOTCONTAINS	Does not contain the specified value.
NOTEXISTS	The specified header or query does not exist.

1. If a Value text box appears, type a string or numeric value, as appropriate. For example, chose REQ as the Flow Type, HTTP as the Protocol, and HEADER as the qualifier, and then type the value of the header string in the Value field and the header type for which you want to match the string in the Header Name text box.
2. Click **OK**.
3. To create a compound expression, click Add. Note that the type of compounding that is done depends on the following choices in the Create Policy dialog box:
   - **Match Any Expression.** The expressions are in a logical OR relationship.
   - **Match All Expressions.** The expressions are in a logical AND relationship.
   - **Tabular Expressions.** Click the AND, OR, and parentheses buttons to control evaluation.
   - **Advanced Free-Form.** Enter the expressions components directly into the Expression field, and click the AND, OR, and parentheses buttons to control evaluation.

Bind a classic policy

September 14, 2021

Depending on the policy type, you can bind a classic policy either globally or to a virtual server. Policy bind points are described in the table, “Policy Type and Bind Points for Policies in Features That Use Classic Policies.”

Note: You can bind a classic policy to multiple bind points.

Bind a classic policy globally by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  bind cmp global <policyName> [-priority <positive_integer>]
2
```



```
-  show cmp global
<!--NeedCopy-->
```

Example

```
> bind cmp global cmp-pol-compress -priority 2
 Done
> show cmp global
1)      Policy Name: cmp-pol-compress   Priority: 2
2)      Policy Name: ns_nocmp_xml_ie    Priority: 8700
3)      Policy Name: ns_nocmp_mozilla_47        Priority: 8800
4)      Policy Name: ns_cmp_mscss       Priority: 8900
5)      Policy Name: ns_cmp_msapp       Priority: 9000
6)      Policy Name: ns_cmp_content_type        Priority: 10000
 Done
>
<!--NeedCopy-->
```

Bind a classic policy to a virtual server by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  bind lb vserver <name> [<targetVserver>] [-policyName <string> [-
   priority <positive_integer>]

-  show lb vserver<name>
<!--NeedCopy-->
```

Example

```
> bind lb vserver lbtemp -policyName cmp-pol-compress -priority 1
 Done
> show lb vserver lbtemp
        lbtemp (10.102.29.101:80) - HTTP        Type: ADDRESS
        State: UP
        Last state change was at Tue Oct 27 06:40:38 2009 (+557 ms)
        Time since last state change: 0 days, 02:00:40.330
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
```



```
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state
           changed to UP
        Group: vserver-grp
        Mode: IP
        Persistence: COOKIEINSERT (version 0)   Persistence Backup:
           SOURCEIP   Persistence Mask: 255.255.255.255
        Persistence Timeout: 2 min      Backup Persistence Timeout: 2
           min
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
1) http-one (10.102.29.252: 80) - HTTP State: UP        Weight: 1
        Persistence Cookie Value : NSC_wtfswfs-hsq=
           ffffffff096e03ed45525d5f4f58455e445a4a423660
1)      Policy : cmp-pol-compress Priority:1
 Done
>
<!--NeedCopy-->
```

Bind a classic policy globally by using the GUI

Note: This procedure documents the Global Bindings dialog box. Depending on the feature for which you want to globally bind a policy, the route by which you arrive at this dialog box may be different.

1. In the navigation pane, expand the feature for which you want to globally bind a classic policy, and then locate the policy that you want to bind globally.

   Note: You cannot globally bind policies for Content Switching, Cache Redirection, SureConnect, Priority Queuing, or
   Citrix Gateway Authorization.

2. In the details pane, click Global Bindings.
3. In the Bind/Unbind <feature name> Policy(s) to Global dialog box, click **Insert Policy**.
4. In the **Policy Name** column, click the name of an existing policy that you want to globally bind, or click New Policy to open the Create <feature name> Policy dialog box.
5. After you have selected the policy or created a new policy, in the Priority column, type the priority value.

The lower the number, the sooner this policy is applied relative to other policies. For example, a policy assigned a priority of 10 is applied before a policy with a priority of 100. You can use the same priority for different policies. All features that use classic policies implement only the first policy that a connection matches, so policy priority is important for getting the results you intend.

As a best practice, leave room to add policies by setting priorities with intervals of 50 (or 100) between each policy.

6. Click **OK**.

Bind a classic policy to a virtual server by using the GUI

1. In the navigation pane, expand the feature that contains the virtual server to which you want to bind a classic policy (for example, if you want to bind a classic policy to a content switching virtual server, expand Traffic Management > Content Switching), and then click Virtual Servers.
2. In the details pane, select the virtual server, and then click Open.
3. In the Configure <Feature> Virtual Server dialog box, on the Policies tab, click the feature icon for the type policy that you want, and then click Insert Policy.
4. In the Policy Name column, click the name of an existing policy that you want to bind to a virtual server, or click A to open the Create <feature name> Policy dialog box.
5. After you have selected the policy or created a new policy, in the Priority column, set the priority.

   If you are binding a policy to a content switching virtual server, in the Target column, select a load balancing virtual server to which traffic that matches the policy should be sent.
6. Click **OK**.

View classic policies

September 14, 2021

You can view classic policies by using either the configuration utility or the command line. You can view details such as the policy's name, expression, and bindings.

View a classic policy and its binding information by using the CLI

At the command prompt, type the following commands to view a classic policy and its binding information:

```
show <featureName> policy [policyName]
```


Example

```
> show appfw policy GenericApplicationSSL_
        Name: GenericApplicationSSL_    Rule: ns_only_get_adv
        Profile: GenericApplicationSSL_Prof1    Hits: 0
        Undef Hits: 0
        Policy is bound to following entities
        1) REQ VSERVER app_u_GenericApplicationSSLPortalPages
           PRIORITY : 100
 Done
<!--NeedCopy-->
```

Note: If you omit the policy name, all policies are listed without the binding details.

View classic policies and policy bindings by using the GUI

1. In the navigation pane, expand the feature whose policies you want to view, (for example, if you want to view application firewall policies, expand Application Firewall), and then click Policies.
2. In the details pane, do one or more of the following:
   - To view details for a specific policy, click the policy. Details appear in the Details area of the configuration pane.
   - To view bindings for a specific policy, click the policy, and then click Show Bindings.
   - To view global bindings, click the policy, and then click Global Bindings. Note that you cannot bind a Content Switching, Cache Redirection, SureConnect, Priority Queuing, or Citrix Gateway Authorization policy globally.

Create named classic expressions

September 14, 2021

A named classic expression is a classic expression that can be referenced through an assigned name. Often, you need to configure classic expressions that are large or complex and form a part of a larger compound expression. You might also configure classic expressions that you need to use frequently and in multiple compound expressions or classic policies. In these scenarios, you can create the classic expression you want, save it with a name of your choice, and then reference the expression from compound expressions or policies through its name. This saves configuration time and improves the readability of complex compound expressions. Additionally, any modifications to a named classic expression need to be made only once.

Some named expressions are built-in, and a subset of these are read-only. Built-in named expressions are divided into four categories: General, Anti-Virus, Personal Firewall, and Internet Security. General

named expressions have a wide variety of uses. For example, from the General category, you can use the expressions ns_true and ns_false to specify a value of TRUE or FALSE, respectively, to be returned for all traffic. You can also identify data of a particular type (for example, HTM, DOC, or GIF files), determine whether caching headers are present, or determine whether the round trip time for packets between a client and the Citrix ADC is high (over 80 milliseconds).

Anti-Virus, Personal Firewall, and Internet Security named expressions test clients for the presence of a particular program and version and are used primarily in Citrix Gateway policies.

Note: You cannot modify or delete built-in named expressions.

Create a named classic expression by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  add expression <name> <value> [-comment <string>] [-
     clientSecurityMessage <string>]
2 -  show expression [<name> | -type CLASSIC
3 <!--NeedCopy-->
```

Example

```
1 > add expression classic_ne "REQ.HTTP.URL CONTAINS www.example1.com" -
     comment "Checking the URL for www.example1.com"
2  Done
3 > show expression classic_ne
4 1)      Name: classic_ne  Expr: REQ.HTTP.URL CONTAINS www.example1.com
      Hits: 0 Type : CLASSIC
5         Comment: "Checking the URL for www.example1.com"
6  Done
7 >
8 <!--NeedCopy-->
```

Create a named classic expression by using the GUI

1. In the navigation pane, expand AppExpert, expand Expressions, and then click Classic Expressions.
2. In the details pane, click Add.

   Note: Some of the built-in expressions in the Expressions list are read-only.
3. In the Create Policy Expression dialog box, specify values for the following parameters:

- Expression Name*—name
- Client Security Message—clientSecurityMessage
- Comments—comment

*A required parameter

4. To create the expression, do one of the following:
   - You can choose inputs to this expression from the Named Expressions drop-down list.
   - You can create a new expression, as described in Add an expression for a classic policy by using the GUI.
5. When you are done, click **Close**. Verify that your new expression was created by scrolling to the bottom of the Classic Expressions list to view it.

Expressions reference-advanced policy expressions

September 16, 2021

> **Warning:**
>
> Q and S prefixes are deprecated from Citrix ADC 12.0 build 56.20 onwards and are no longer >supported in Advanced policy expressions.
>
> The SYS.EVAL_CLASSIC_EXPR expression is deprecated from NetScaler 12.0 build 56.20 onwards >and as an alternative, Citrix recommends you to use the Advanced policy expressions. However, >the expression is removed and no longer available on the Citrix ADC appliance release 13.1 >onwards.

The following table is a listing of default syntax expression prefixes, with cross-references to descriptions of these prefixes and the operators that you can specify for them. Note that some prefixes can work with multiple types of operators. For example, a cookie can be parsed by using operators for text or operators for HTTP headers.

You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

Note: The Description column in the following table contains cross-references to additional information about prefix usage and applicable operators for the prefix.

For more information, refer Expression PDF to view the table.

Expressions reference-classic expressions

September 14, 2021

> **Warning**
>
> Classic policy expressions are no longer supported from Citrix ADC 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies

The subtopics listed in the table of contents on the left side of your screen contain tables listing the Citrix ADC classic expressions.

In the table of operators, the result type of each operator is shown at the beginning of the description. In the other tables, the level of each expression is shown at the beginning of the description. For named expressions, each expression is shown as a whole.

Operators

Expression Element	Definition
==	Boolean. Returns TRUE if the current expression equals the argument. For text operations, the items being compared must exactly match one another. For numeric operations, the items must evaluate to the same number.
!=	Boolean. Returns TRUE if the current expression does not equal the argument. For text operations, the items being compared must not exactly match one another. For numeric operations, the items must not evaluate to the same number.
CONTAINS	Boolean. Returns TRUE if the current expression contains the string that is designated in the argument.
NOTCONTAINS	Boolean. Returns TRUE if the current expression does not contain the string that is designated in the argument.
CONTENTS	Text. Returns the contents of the current expression.

Expression Element	Definition
EXISTS	Boolean. Returns TRUE if the item designated by the current expression exists.
NOTEXISTS	Boolean. Returns TRUE if the item designated by the current expression does not exist.
>	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than the argument.
<	Boolean. Returns TRUE if the current expression evaluates to a number that is less than the argument.
>=	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than or equal to the argument.
<=	Boolean. Returns TRUE if the current expression evaluates to a number that is less than or equal to the argument.

General expressions

Expression Element	Definition
REQ	Flow Type. Operates on incoming (or request) packets.
REQ.HTTP	Protocol. Operates on HTTP requests.
REQ.HTTP.METHOD	Qualifier. Designates the HTTP method.
REQ.HTTP.URL	Qualifier. Designates the URL.
REQ.HTTP.URLTOKENS	Qualifier. Designates the URL token.
REQ.HTTP.VERSION	Qualifier. Designates the HTTP version.
REQ.HTTP.HEADER	Qualifier. Designates the HTTP header.
REQ.HTTP.URLLEN	Qualifier. Designates the number of characters in the URL.
REQ.HTTP.URLQUERY	Qualifier. Designates the query portion of the URL.

Expression Element	Definition
REQ.HTTP.URLQUERYLEN	Qualifier. Designates the length of the query portion of the URL.
REQ.SSL	Protocol. Operates on SSL requests.
REQ.SSL.CLIENT.CERT	Qualifier. Designates the entire client certificate.
REQ.SSL.CLIENT.CERT.SUBJECT	Qualifier. Designates the client certificate subject.
REQ.SSL.CLIENT.CERT.ISSUER	Qualifier. Designates the issuer of the client certificate.
REQ.SSL.CLIENT.CERT.SIGALGO	Qualifier. Designates the validation algorithm used by the client certificate.
REQ.SSL.CLIENT.CERT.VERSION	Qualifier. Designates the client certificate version.
REQ.SSL.CLIENT.CERT.VALIDFROM	Qualifier. Designates the date before which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.VALIDTO	Qualifier. Designates the date after which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.SERIALNUMBER	Qualifier. Designates the serial number of the client certificate.
REQ.SSL.CLIENT.CIPHER.TYPE	Qualifier. Designates the encryption protocol used by the client.
REQ.SSL.CLIENT.CIPHER.BITS	Qualifier. Designates the number of bits used by the client's SSL key.
REQ.SSL.CLIENT.SSL.VERSION	Qualifier. Designates the SSL version that the client is using.
REQ.TCP	Protocol. Operates on incoming TCP packets.
REQ.TCP.SOURCEPORT	Qualifier. Designates the source port of the incoming packet.
REQ.TCP.DESTPORT	Qualifier. Designates the destination port of the incoming packet.
REQ.IP	Protocol. Operates on incoming IP packets.
REQ.IP.SOURCEIP	Qualifier. Designates the source IP of the incoming packet.

Expression Element	Definition
REQ.IP.DESTIP	Qualifier.Designates the destination IP of the incoming packet.
RES	Flow Type. Operates on outgoing (or response) packets.
RES.HTTP	Protocol. Operates on HTTP responses.
RES.HTTP.VERSION	Qualifier. Designates the HTTP version.
RES.HTTP.HEADER	Qualifier. Designates the HTTP header.
RES.HTTP.STATUSCODE	Qualifier. Designates the status code of the HTTP response.
RES.TCP	Protocol. Operates on incoming TCP packets.
RES.TCP.SOURCEPORT	Qualifier. Designates the source port of the outgoing packet.
RES.TCP.DESTPORT	Qualifier. Designates the destination port of the outgoing packet.
RES.IP	Protocol. Operates on outgoing IP packets.
RES.IP.SOURCEIP	Qualifier. Designates the source IP of the outgoing packet. This can be in IPv4 or IPv6 format. For example: add expr exp3 “sourceip == 10.102.32.123 –netmask 255.255.255.0 && destip == 2001::23/120”.
RES.IP.DESTIP	Qualifier. Designates the destination IP of the outgoing packet.

Client security expressions

The expressions to configure client settings on the Access Gateway with the following software:

- Antivirus
- Personal firewall
- Antispam
- Internet Security

For example usage, see http://support.citrix.com/article/CTX112599.

Actual Expression	Definition
CLIENT.APPLICATION.AV(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated anti-virus program and version.
CLIENT.APPLICATION.AV(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated anti-virus program and version.
CLIENT.APPLICATION.PF(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated personal firewall program and version.
CLIENT.APPLICATION.PF(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated personal firewall program and version.
CLIENT.APPLICATION.IS(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated internet security program and version.
CLIENT.APPLICATION.IS(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated internet security program and version.
CLIENT.APPLICATION.AS(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated anti-spam program and version.
CLIENT.APPLICATION.AS(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated anti-spam program and version.

Network-based expressions

Expression	Definition
REQ	Flow Type. Operates on incoming, or request, packets.
REQ.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
REQ.INTERFACE.ID	Qualifier. Operates on the ID of the designated Citrix ADC interface.
REQ.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated Citrix ADC interface.

Expression	Definition
REQ.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated Citrix ADC interface.
REQ.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated Citrix ADC interface.
REQ.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
REQ.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.
RES	Flow Type. Operates on outgoing (or response) packets.
RES.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
RES.INTERFACE.ID	Qualifier. Operates on the ID of the designated Citrix ADC interface.
RES.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated Citrix ADC interface.
RES.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated Citrix ADC interface.
RES.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated Citrix ADC interface.
RES.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
RES.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.

Date/time expressions

Expression	Definition
TIME	Qualifier. Operates on the date and time of day, GMT.
DATE	Qualifier. Operates on the date, GMT.

Expression	Definition
DAYOFWEEK	Operates on the specified day in the week, GMT.

File system expressions

You can specify file system expressions in authorization policies for users and groups who access file sharing through the Citrix Gateway file transfer utility (the VPN portal). These expressions work with the Citrix Gateway file transfer authorization feature to control user access to file servers, folders, and files. For example, you can use these expressions in authorization policies to control access based on file type and size.

For more information, refer to the File Name Expression pdf.

Note: File system expressions do not support regular expressions.

Built-in named expressions (General)

Expression	Definition
ns_all_apps_ncomp	Tests for connections with destination ports between 0 and 65535. In other words, tests for all applications.
ns_cachecontrol_nocache	Tests for connections with an HTTP Cache-Control header that contains the value “no-cache”.
ns_cachecontrol_nostore	Tests for connections with an HTTP Cache-Control header that contains the value “no-store”.
ns_cmpclient	Tests the client to determine if it accepts compressed content.
ns_content_type	Tests for connections with an HTTP Content-Type header that contains “text”.
ns_css	Tests for connections with an HTTP Content-Type header that contains “text/css”.
ns_ext_asp	Tests for HTTP connections to any URL that contains the string .asp—in other words, any connection to an active server page (ASP).

Expression	Definition
ns_ext_cfm	Tests for HTTP connections to any URL that contains the string .cfm
ns_ext_cgi	Tests for HTTP connections to any URL that contains the string .cgi—in other words, any connection to a common gateway interface (CGI) script.
ns_ext_ex	Tests for HTTP connections to any URL that contains the string .ex
ns_ext_exe	Tests for HTTP connections to any URL that contains the string .exe—in other words, any connection to a executable file.
ns_ext_htx	Tests for HTTP connections to any URL that contains the string .htx
ns_ext_not_gif	Tests for HTTP connections to any URL that does not contain the string .png—in other words, any connection to a URL that is not a GIF image.
ns_ext_not_jpeg	Tests for HTTP connections to any URL that does not contain the string .jpeg—in other words, any connection to a URL that is not a JPEG image.
ns_ext_shtml	Tests for HTTP connections to any URL that contains the string .shtml—in other words, any connection to a server-parsed HTML page.
ns_false	Always returns a value of FALSE.

Expression	Definition
ns_farclient	Client is in a different geographical region from the Citrix ADC, as determined by the geographical region in the client's IP address. The following regions are predefined: 192.0.0.0 – 193.255.255.255: Multi-regional, 194.0.0.0 – 195.255.255.255: European Union, 196.0.0.0 – 197.255.255.255: Other1, 198.0.0.0 – , 199.255.255.255: North America, 200.0.0.0 – 201.255.255.255: Central and South America, 202.0.0.0 – 203.255.255.255: Pacific Rim, 204.0.0.0 – 205.255.255.255: Other2, and 206.0.0.0 – 207.255.255.255: Other3
ns_header_cookie	Tests for HTTP connections that contain a Cookie header.
ns_header_pragma	Tests for HTTP connections that contain a Pragma: no-cache header.
ns_mozilla_47	Tests for HTTP connections whose User-Agent header contains the string Mozilla/4.7—in other words, any connection from a client using the Mozilla 4.7 Web browser.
ns_msexcel	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msexcel—in other words, any connection transmitting a Microsoft Excel spreadsheet.
ns_msie	Tests for HTTP connections whose User-Agent header contains the string MSIE—in other words, any connection from a client using any version of the Internet Explorer Web browser.
ns_msppt	Tests for HTTP connections whose Content-Type header contains the string application/vnd.ms-powerpoint—in other words, any connection transmitting a Microsoft PowerPoint file.

Expression	Definition
ns_msword	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msword—in other words, any connection transmitting a Microsoft Word file.
ns_non_get	Tests for HTTP connections that use any HTTP method except for GET.
ns_slowclient	Returns TRUE if the average round trip time between the client and the Citrix ADC is more than 80 milliseconds.
ns_true	Returns TRUE for all traffic.
ns_url_path_bin	Tests the URL path to see if it points to the /bin/ directory.
ns_url_path_cgibin	Tests the URL path to see if it points to the CGI-BIN directory.
ns_url_path_exec	Tests the URL path to see if it points to the /exec/directory.
ns_url_tokens	Tests for the presence of URL tokens.
ns_xmldata	Tests for the presence of XML data.

Built-in named expressions (Anti-Virus)

Expression	Definition
McAfee Virus Scan 11	Tests to determine whether the client is running the latest version of McAfee VirusScan.
McAfee Antivirus	Tests to determine whether the client is running any version of McAfee Antivirus.
Symantec AntiVirus 10 (with Updated Definition File)	Tests to determine whether the client is running the most current version of Symantec AntiVirus.
Symantec AntiVirus 6.0	Tests to determine whether the client is running Symantec AntiVirus 6.0.
Symantec AntiVirus 7.5	Tests to determine whether the client is running Symantec AntiVirus 7.5.

Expression	Definition
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
TrendMicro AntiVirus 11.25	Tests to determine whether the client is running Trend Microsystems' AntiVirus, version 11.25.
Sophos Antivirus 4	Tests to determine whether the client is running Sophos Antivirus, version 4.
Sophos Antivirus 5	Tests to determine whether the client is running Sophos Antivirus, version 5.
Sophos Antivirus 6	Tests to determine whether the client is running Sophos Antivirus, version 6.

Built-in named expressions (Personal Firewall)

Expression	Definition
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
Sygate Personal Firewall 5.6	Tests to determine whether the client is running the Sygate Personal Firewall, version 5.6.
ZoneAlarm Personal Firewall 6.5	Tests to determine whether the client is running the ZoneAlarm Personal Firewall, version 6.5.

Built-in named expressions (Client Security)

Expression	Definition
Norton Internet Security	Tests to determine whether the client is running any version of Norton Internet Security.

Summary examples of default syntax expressions and policies

September 14, 2021

The following table provides examples of default syntax expressions that you can use as the basis for your own default syntax expressions.

Table 1. Examples of Default Syntax Expressions

Expression Type	Sample Expressions
Look at the method used in the HTTP request.	`http.req.method.eq(post)http.req.method.eq(get)`

Check the Cache-Control or Pragma header value in an HTTP request (req) or response (res).	`http.req.header("Cache-Control").contains("no-store")`
	`http.req.header("Cache-Control").contains("no-cache")`
	`http.req.header("Pragma").contains("no-cache")`
	`http.res.header("Cache-Control").contains("private")`
	`http.res.header("Cache-Control").contains("public")`
	`http.res.header("Cache-Control").contains("must-revalidate")http.res.header("Cache-Control").contains ("proxy-revalidate")`
	`http.res.header("Cache-Control").contains("max-age")`

Check for the presence of a header in a request (req) or response (res).	`http.req.header("myHeader").exists` `http.res.header("myHeader").exists`

Expression Type	Sample Expressions
Look for a particular file type in an HTTP request based on the file extension.	`http.req.url.contains(".html")http.req.url.contains(".cgi")http.req.url.contains(".asp")http.req.url.contains(".exe")http.req.url.contains(".cfm")http.req.url.contains(".ex")http.req.url.contains(".shtml")http.req.url.contains(".htx")http.req.url.contains("/cgi-bin/")http.req.url.contains("/exec/")http.req.url.contains("/bin/")`
Look for anything that is other than a particular file type in an HTTP request.	`http.req.url.contains(".png").not; http.req.url.contains(".jpeg").not`
Check the type of file that is being sent in an HTTP response based on the Content-Type header.	`http.res.header("Content-Type").contains("text")http.res.header("Content-Type").contains "application/msword")http.res.header("Content-Type").contains("vnd.ms-excel")http.res.header("Content-Type").contains("application/vnd.ms-powerpoint"); http.res.header("Content-Type").contains("text/css"); http.res.header("Content-Type").contains("text/xml"); http.res.header("Content-Type").contains("image/")`
Check whether this response contains an expiration header.	`http.res.header("Expires").exists`
Check for a Set-Cookie header in a response.	`http.res.header("Set-Cookie").exists`
Check the agent that sent the response.	`http.res.header("User-Agent").contains("Mozilla/4.7")http.res.header("User-Agent").contains("MSIE")`

Expression Type	Sample Expressions
Check if the first 1024 bytes of the body of a request starts with the string “some text”.	`http.req.body(1024).contains("some text")`

The following table shows examples of policy configurations and bindings for commonly used functions.

Table 2. Examples of Default Syntax Expressions and Policies

Purpose	Example
Use the rewrite feature to replace occurrences of `http://with https://` in the body of an HTTP response.	`add rewrite action httpRewriteAction replace_all http.res.body(50000)"\"https://\""-pattern http://add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction`
Replace all occurrences of “abcd” with “1234” in the first 1000 bytes of the HTTP body.	`add rewrite action abcdTo1234Action replace_all "http.req.body(1000)""\"1234\""-pattern abcd add rewrite policy abcdTo1234Policy "http.req.body(1000).contains(\"abcd\")" abcdTo1234Action bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE`
Downgrade the HTTP version to 1.0 to prevent the server from chunking HTTP responses.	`add rewrite action downgradeTo1.0Action replace http.req.version.minor "\"0\""add rewrite policy downgradeTo1.0Policy "http.req.version.minor.eq(1)"downgradeTo1.0Action bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST`

Purpose	Example
Remove references to the HTTP or HTTPS protocol in all responses, so that if the user's connection is HTTP, the link is opened by using HTTP, and if the user's connection is HTTPS, the link is opened by using HTTPS.	`add rewrite action remove_http_https replace_all "http.res.body(1000000).set_text_mode(ignorecase)""\"//\""-pattern "re~https?://|HTTPS?://~"add rewrite policy remove_http_https true remove_http_https bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE`
Rewrite instances of http: to https: in all URLs.	`add responder action httpToHttpsAction redirect "\"https://\" + http.req.hostname + http.req.url"-bypassSafetyCheck YES add responder policy httpToHttpsPolicy "!CLIENT.SSL.IS_SSL" httpToHttpsAction bind responder global httpToHttpsPolicy 1 END -type OVERRIDE`
Modify a URL to redirect from URL A to URL B. In this example, "file5.html" is appended to the path.	`add responder action appendFile5Action redirect \"http://\" + http.req.hostname + http.req.url + \"/file5.html\""-bypassSafetyCheck YES add responder policy appendFile5Policy "http.req.url.eq(\"/testsite\")" appendFile5Action bind responder global appendFile5Policy 1 END -type OVERRIDE`

Purpose	Example
Redirect an external URL to an internal URL.	`add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server'"www.my.host.com"'add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")'act_external_to_internal bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE`
Redirect requests to www.example.com that have a query string to www.Webn.example.com. The value n is derived from a server parameter in the query string, for example, server=5.	`add rewrite action act_redirect_query REPLACE q##http.req.header("Host").before_str(".example.com")'"Web"+ http.req.url.query.value("server")## add rewrite policy pol_redirect_query q##http.req.header("Host").eq("www.example.com")&& http.req.url.contains("?")'act_redirect_query##`
Limit the number of requests per second from a URL.	`add ns limitSelector ip_limit_selector http.req.url "client.ip.src"add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector add responder action my_Web_site_redirect_action redirect "\"http://www.mycompany.com/\""add responder policy ip_limit_responder_policy "http.req.url.contains(\"myasp.asp\")&& sys.check_limit (\"ip_limit_identifier\")"my_Web_site_redirect_action bind responder global ip_limit_responder_policy 100 END -type default`

Purpose	Example
Check the client IP address but pass the request without modifying the request.	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER ("x-forwarded-for").EXISTS HTTP.REQ.HEADER ("client-ip").EXISTS'NOREWRITE bind rewrite global check_client_ip_policy 100 END`
Remove old headers from a request and insert an NS-Client header.	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client ' CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END`

Purpose	Example
Remove old headers from a request, insert an NS-Client header, and then modify the “insert header” action so that the value of the inserted header contains the client IP values from the old headers and the Citrix ADC appliance’s connection IP address. Note that this example repeats the previous example, with the exception of the final set rewrite action.	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END set rewrite action insert_ns_client_header -stringBuilderExpr 'HTTP.REQ.HEADER("x-forwarded-for").VALUE(0)+ " " + HTTP.REQ.HEADER("client-ip").VALUE(0)+ " " + CLIENT.IP.SRC'-bypassSafetyCheck YES`

Tutorial examples of default syntax policies for rewrite

September 14, 2021

With the rewrite feature, you can modify any part of an HTTP header, and, for responses, you can modify the HTTP body. You can use this feature to accomplish several useful tasks, such as removing unnecessary HTTP headers, masking internal URLs, redirecting webpages, and redirecting queries or keywords.

In the following examples, you first create a rewrite action and a rewrite policy. Then you bind the policy globally.

This document includes the following details:

- Redirecting an External URL to an Internal URL
- Redirecting a Query
- Rewriting HTTP to HTTPS
- Removing Unwanted Headers
- Reducing Web Server Redirects
- Masking the Server Header
- Converting plain text to URL encoded string and the opposite way

For more information about the commands and syntax descriptions, see Rewrite Command Reference page.

Redirecting an external URL to an internal URL

This example describes how to create a rewrite action and rewrite policy that redirects an external URL to an internal URL. You create an action, called act_external_to_internal, that performs the rewrite. Then you create a policy called pol_external_to_internal.

To redirect an external URL to an internal URL by using the CLI

- To create the rewrite action, at the command prompt, type:

```
add rewrite action act_external_to_internal REPLACE “http.req.hostname.
server” “\” host_name_of_internal_Web_server\” ”
```

- To create the rewrite policy, at the Citrix ADC command prompt, type:

```
add rewrite policy pol_external_to_internal "http.req.hostname.server.eq(\"
host_name_of_external_Web_server\")"act_external_to_internal
```

- Bind the policy globally.

To redirect an external URL to an internal URL by using the configuration utility

1. Navigate to **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Rewrite Action** dialog box, enter the name act_external_to_internal.
4. To replace the HTTP server host name with the internal server name, choose **Replace** from the Type list box.
5. In the header name field, type **Host**.
6. In the string expression for a replacement text field, type the internal host name of your Web server.
7. Click **Create** and then click **Close**.
8. In the navigation pane, click **Policies**.
9. In the details pane, click **Add**.
10. In the Name field, type pol_external_to_internal. This policy detects connections to the Web server.
11. In the **Action** drop-down menu, choose the action act_external_to_internal.
12. In the Expression editor, construct the following expression:

```
HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
<!--NeedCopy-->
```

1. Bind your new policy globally.

Redirecting a query

This example describes how to create a rewrite action and rewrite policy that redirects a query to the proper URL. The example assumes that the request contains a Host header set to **www.example.com** and a GET method with the **string /query.cgi?server=5**. The redirect extracts the domain name from the host header and the number from the query string, and redirects the user's query to the server **Web5.example.com**, where the rest of the user's query is processed.

> **Note:**
>
> Although the following commands appear on multiple lines, you must enter them on a single line without line breaks.

To redirect a query to the appropriate URL using the CLI

- To create a rewrite action named act_redirect_query that replaces the HTTP server host name with the internal server name, type:

```
add rewrite action act_redirect_query REPLACE http.req.header("Host").
before_str(".example.com")'"Web" + http.req.url.query.value("server")'
```


- To create a rewrite policy named pol_redirect_query, type the following commands at the Citrix ADC command prompt. This policy detects connections, to the Web server, that contain a query string. Do not apply this policy to connections that do not contain a query string:

```
add rewrite policy pol_redirect_query 'http.req.header("Host").eq(www.
example.com)&& http.req.url.contains("?")'act_redirect_query
```

- Bind your new policy globally.

Because this rewrite policy is highly specific and must be run before any other rewrite policies, it is advisable to assign it a high priority. If you assign it a priority of 1, it is evaluated first.

Rewriting HTTP to HTTPS

This example describes how to rewrite Web server responses to find all URLs that begin with the string "HTTP" and replace that string with "https." You can use it to avoid having to update webpages after moving a server from HTTP to HTTPS.

To redirect HTTP URLs to HTTPS by using the CLI

- To create a rewrite action named act_replace_http_with_https that replaces all instances of the string "HTTP" with the string "https," enter the following command:

```
add rewrite action act_replace_http_with_https replace_all 'http.res.body
(100)''"https"'-pattern http
```

- To create a rewrite policy named pol_replace_http_with_https that detects connections to the Web server, enter the following command:

```
add rewrite policy pol_replace_http_with_https TRUE act_replace_http_with_https
 NOREWRITE
```

- Bind your new policy globally.

To troubleshoot this rewrite operation, see "Case Study: Rewrite Policy for Converting HTTP Links to HTTPS not Working."

Removing Unwanted Headers

This example explains how to use a Rewrite policy to remove unwanted headers. Specifically, the example shows how to remove the following headers:

- **Accept Encoding header.** Removing the Accept Encoding header from HTTP responses prevents compression of the response.
- **Content Location header.** Removing the Content Location header from HTTP responses prevents your server from providing a hacker with information that might allow a security breach.

To delete headers from HTTP responses, you create a rewrite action and a rewrite policy, and you bind the policy globally.

To create the appropriate Rewrite action by using the CLI

At the command prompt, type one of the following commands to either remove the Accept Encoding header and prevent response compression or remove the Content Location header:

- `add rewrite action "act_remove-ae"delete_http_header "Accept-Encoding"`
- `add rewrite action "act_remove-cl"delete_http_header "Content-Location"`

To create the appropriate Rewrite policy by using the CLI

At the command prompt, type one of the following commands to remove either the Accept Encoding header or the Content Location header:

- `add rewrite policy "pol_remove-ae"true "act_remove-ae"`
- `add rewrite policy "pol_remove-cl"true "act_remove-cl"`

To bind the policy globally by using the CLI

At the command prompt, type one of the following commands, as appropriate, to globally bind the policy that you have created:

- `bind rewrite global pol_remove_ae 100`
- `bind rewrite global pol_remove_cl 200`

Reducing Web Server Redirects

This example explains how to use a Rewrite policy to modify connections to your home page and other URLs that end with a forward slash (/) to the default index page for your server, preventing redirects and reducing load on your server.

To modify directory-level HTTP requests to include the default home page by using the CLI

- To create a Rewrite action named action-default-homepage that modifies URLs that end in a forward slash to include the default home page index.html, type:

```
add rewrite action "action-default-homepage"replace http.req.url.path "\"/index.html\""
```

- To create a Rewrite policy named policy-default-homepage that detects connections to your home page and applies your new action, type:


```
add rewrite policy "policy-default-homepage"q\##http.req.url.path.EQ("/")"
action-default-homepage"\##
```

- Globally bind your new policy to put it into effect.

Masking the Server Header

This example explains how to use a Rewrite policy to mask the information in the Server header in HTTP responses from your Web server. That header contains information that hackers can use to compromise your website. While masking the header will not prevent a skilled hacker from finding out information about your server, it makes hacking your Web server more difficult and encourage hackers to choose less well protected targets.

To mask the Server header in responses from the CLI

1. To create a Rewrite action named act_mask-server that replaces the contents of the Server header with an uninformative string, type:

```
add rewrite action "act_mask-server"replace "http.RES.HEADER(\"Server\")""
\"Web Server 1.0\""
```

1. To create a Rewrite policy named pol_mask-server that detects all connections, type:

```
add rewrite policy "pol_mask-server"true "act_mask-server"
```

1. Globally bind your new policy to put it into effect.

How to convert plain text to URL encoded string and the opposite way

The following expressions convert plain text to URL encoded string and the opposite way:

1. URL_RESERVED_CHARS_SAFE (string to URL ENCODED).

Example:

```
("abc def&123").URL_RESERVED_CHARS_SAFE
Output will be
 “abc%20def%26123”  which is url encoded.
<!--NeedCopy-->
```

1. SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE. (URL ENCODED to string)

Example:

```
("abc%20def%26123").SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE
Output will be
 “abc def&123”
```



```
<!--NeedCopy-->
```

Tutorial examples of classic policies

September 14, 2021

The following examples describe useful examples of classic policy configuration for certain Citrix ADC features, such as Citrix Gateway, application firewall, and SSL.

This document includes the following details:

- Citrix Gateway Policy to Check for a Valid Client Certificate
- Application Firewall Policy to Protect a Shopping Cart Application
- Application Firewall Policy to Protect Scripted Web Pages
- DNS Policy to Drop Packets from Specific IPs
- SSL Policy to Require Valid Client Certificates

Citrix Gateway policy to check for a valid client certificate

The following policies enable the Citrix ADC to ensure that a client presents a valid certificate before establishing a connection to a company's SSL VPN.

To check for a valid client certificate by using the command line interface

- Add an action to perform client certificate authentication.

  ```
  add ssl action act1 -clientAuth DOCLIENTAUTH
  ```

- Create an SSL policy to evaluate the client requests.

  ```
  add ssl policy pol1 -rule "REQ.HTTP.METHOD == GET"-action act1
  ```

- Add a rewrite action to insert the certificate issuer details into the HTTP header of the requests being sent to web server.

  ```
  add rewrite action act2 insert_http_header "CertDN"CLIENT.SSL.CLIENT_CERT
  .SUBJECT
  ```

- Create a rewrite policy to insert the certificate issuer details, if the client certificate exists.

  ```
  add rewrite policy pol2 "CLIENT.SSL.CLIENT_CERT.EXISTS"act2
  ```

  Bind these new policies to the Citrix ADC VIP to put them into effect.

Application firewall policy to protect a shopping cart application

Shopping cart applications handle sensitive customer information, for example, credit card numbers and expiration dates, and they access back-end database servers. Many shopping cart applications also use legacy CGI scripts, which can contain security flaws that were unknown at the time they were written, but are now known to hackers and identity thieves.

A shopping cart application is particularly vulnerable to the following attacks:

- **Cookie tampering.** If a shopping cart application uses cookies, and does not perform the appropriate checks on the cookies that users return to the application, an attacker could modify a cookie and gain access to the shopping cart application under another user's credentials. Once logged on as that user, the attacker could obtain sensitive private information about the legitimate user or place orders using the legitimate user's account.
- **SQL injection.** A shopping cart application normally accesses a back-end database server. Unless the application performs the appropriate safety checks on the data users return in the form fields of its Web forms before it passes that information on to the SQL database, an attacker can use a Web form to inject unauthorized SQL commands into the database server. Attackers normally use this type of attack to obtain sensitive private information from the database or modify information in the database.

The following configuration will protect a shopping cart application against these and other attacks.

To protect a shopping cart application by using the configuration utility

1. Navigate to **Security > Application Firewall > Profiles**, and then click **Add.**
2. In the **Create Application Firewall Profile** dialog box, in the Profile Name field, enter shopping_cart.
3. In the Profile Type drop-down list, select Web Application.
4. In the Configure Select Advanced defaults.
5. Click **Create** and then click **Close.**
6. In the details view, double-click the new profile.
7. In the Configure Web Application Profile dialog box, configure your new profile as described below:
    - Click the Checks tab, double-click the Start URL check, and in the Modify Start URL Check dialog box, click the General tab and disable blocking, and enable learning, logging, statistics, and URL closure. Click OK and then click Close.

      Note that if you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:


```
set appfw profile shopping_cart -startURLAction LEARN LOG STATS -
startURLClosure ON
```

- For the Cookie Consistency check and Form Field Consistency checks, disable blocking, and enable learning, logging, statistics, using a similar method to the Modify Start URL Check configuration.

  If you are using the command line, you configure these settings by typing the following commands:

  ```
  set appfw profile shopping_cart -cookieConsistencyAction LEARN LOG
  STATS
  ```

  ```
  set appfw profile shopping_cart -fieldConsistencyAction LEARN LOG
  STATS
  ```

- For the SQL Injection check, disable blocking, and enable learning, logging, statistics, and transformation of special characters in the Modify SQL Injection Check dialog box, General tab, Check Actions section.

  If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

  ```
  set appfw profile shopping_cart -SQLInjectionAction LEARN LOG STATS
   -SQLInjectionTransformSpecialChars ON
  ```

- For the Credit Card check, disable blocking; enable logging, statistics, and masking of credit card numbers; and enable protection for those credit cards you accept as forms of payment.

  - If you are using the configuration utility, you configure blocking, logging, statistics, and masking (or x-out) in the Modify Credit Card Check dialog box, General tab, Check Actions section. You configure protection for specific credit cards in the Settings tab of the same dialog box.
  - If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

  ```
  set appfw profile shopping_cart -creditCardAction LOG STATS -
  creditCardXOut ON -creditCard <name> [<name>...]
  ```

  For <name> you substitute the name of the credit card you want to protect. For Visa, you substitute VISA. For Master Card, you substitute MasterCard. For American Express, you substitute Amex. For Discover, you substitute Discover. For Diners Club, you substitute DinersClub. For JCB, you substitute JCB.

8. Create a policy named shopping_cart that detects connections to your shopping cart application and applies the shopping_cart profile to those connections.

To detect connections to the shopping cart, you examine the URL of incoming connections. If you host your shopping cart application on a separate host (a wise measure for security and other reasons), you can simply look for the presence of that host in the URL. If you host your shopping cart in a directory on a host that handles other traffic, as well, you must determine that the connection is going to the appropriate directory and/or HTML page.

The process for detecting either of these is the same; you create a policy based on the following expression, and substitute the proper host or URL for <string>.

```
REQ.HTTP.HEADER URL CONTAINS <string>
<!--NeedCopy-->
```

- If you are using the configuration utility, you navigate to the application firewall Policies page, click the Add… button to add a new policy, and follow the policy creation process described in "To create a policy with classic expressions using the configuration utility" beginning on page 201 and following.
- If you are using the command line, you type the following command at the prompt and press Enter:

  ```
  add appfw policy shopping_cart "REQ.HTTP.HEADER URL CONTAINS <
  string>"shopping_cart
  ```

2. Globally bind your new policy to put it into effect.

Because you want to ensure that this policy will match all connections to the shopping cart, and not be preempted by another more general policy, you should assign a high priority to it. If you assign one (1) as the priority, no other policy can preempt this one.

Application firewall policy to protect scripted web pages

Web pages with embedded scripts, especially legacy JavaScripts, often violate the "same origin rule," which does not allow scripts to access or modify content on any server but the server where they are located. This security vulnerability is called cross-site scripting. The application firewall Cross-Site Scripting rule normally filters out requests that contain cross-site scripting.

Unfortunately, this can cause Web pages with older JavaScripts to stop functioning, even when your system administrator has checked those scripts and knows that they are safe. The example below explains how to configure the application firewall to allow cross-site scripting in Web pages from trusted sources without disabling this important filter for the rest of your Web sites.

To protect Web pages with cross-site scripting by using the command line interface

- At the command line, to create an advanced profile, type:


```
add appfw profile pr_xssokay -defaults advanced
```

- To configure the profile, type:

```
set appfw profile pr_xssokay -startURLAction NONE -startURLClosure OFF
-cookieConsistencyAction LEARN LOG STATS -fieldConsistencyAction LEARN
LOG STATS -crossSiteScriptingAction LEARN LOG STATS$"
```

- Create a policy that detects connections to your scripted Web pages and applies the pr_xssokay profile, type:

```
add appfw policy pol_xssokay "REQ.HTTP.HEADER URL CONTAINS ^\\.pl\\?$
|| REQ.HTTP.HEADER URL CONTAINS ^\\.js$"pr_xssokay
```

- Globally bind the policy.

To protect Web pages with cross-site scripting by using the configuration utility

1. Navigate to **Security > Application Firewall > Profiles.**
2. In the details view, click **Add.**
3. In the **Create Application Firewall Profile** dialog box, create a Web Application profile with advanced defaults and name it pr_xssokay. Click **Create** and then click **Close.**
4. In the details view, click the profile, click Open, and in the Configure Web Application Profile dialog box, configure the pr_xssokay profile as shown below.

   Start URL Check: Clear all actions.

   - Cookie Consistency Check: Disable blocking.
   - Form Field Consistency Check: Disable blocking.
   - Cross-Site Scripting Check: Disable blocking.

   This should prevent blocking of legitimate requests involving Web pages with cross-site scripting that you know are nonetheless safe.
5. Click **Policies**, and then click **Add.**
6. In the **Create Application Firewall Policy** dialog box, create a policy that detects connections to your scripted Web pages and applies the pr_xssokay profile:

   - Policy name: pol_xssokay
   - Associated profile: pr_xssokay

Policy expression: “REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$	REQ.HTTP.HEADER URL CONTAINS ^\.js$”

7. Globally bind your new policy to put it into effect.

DNS Policy to drop packets from specific IPs

The following example describes how to create a DNS action and DNS policy that detects connections from unwanted IPs or networks, such as those used in a DDOS attack, and drops all packets from those locations. The example shows networks within the IANA reserved IP block 192.168.0.0/16. A hostile network will normally be on publicly routable IPs.

To drop packets from specific IPs by using the command line interface

- To create a DNS policy named pol_ddos_drop that detects connections from hostile networks and drops those packets, type:

  ```
  add dns policy pol_ddos_drop 'client.ip.src.in_subnet(192.168.253.128/25)
  || client.ip.src.in_subnet(192.168.254.32/27)'-drop YES'
  ```

  For the example networks in the 192.168.0.0/16 range, you substitute the IP and netmask in ###.###.###.###/## format of each network you want to block. You can include as many networks as you want, separating each CLIENT.IP.SRC.IN_SUBNET(###.###.###.###./##) command with the OR operator.

- Globally bind your new policy to put it into effect.

SSL policy to require valid client certificates

The following example shows an SSL policy that checks the user’s client certificate validity before initiating an SSL connection with a client.

To block connections from users with expired client certificates

- Log on to the command line interface.

  If you are using the GUI, navigate to the SSL Policies page, then in the Data area, click the Actions tab.

- Create an SSL action named act_current_client_cert that requires that users have a current client certificate to establish an SSL connection with the Citrix ADC.

  ```
  add ssl action act_current_client_cert-clientAuth DOCLIENTAUTH -clientCert
   ENABLED -certHeader "clientCertificateHeader"-clientCertNotBefore
  ENABLED -certNotBeforeHeader "Mon, 01 Jan 2007 00:00:00 GMT"
  ```

- Create an SSL policy named pol_current_client_cert that detects connections to the Web server that contain a query string.

  ```
  add ssl policy pol_current_ client_cert 'REQ.SSL.CLIENT.CERT.VALIDFROM
  \>= "Mon, 01 Jan 2007 00:00:00 GMT"'act_block_ssl
  ```

- Bind your new policy globally.

  Because this SSL policy should apply to any user's SSL connection unless a more specific SSL policy applies, you may want to assign it a low priority. If you assign it a priority of one thousand (1000), that should ensure that other SSL policies are evaluated first, meaning that this policy will apply only to connections that do not match more specific policy criteria.

Migration of Apache mod_rewrite rules to the default syntax

September 14, 2021

The Apache HTTP Server provides an engine known as mod_rewrite for rewriting HTTP request URLs. If you migrate the mod_rewrite rules from Apache to the Citrix ADC, you boost back-end server performance. In addition, because the Citrix ADC typically load balances multiple (sometimes thousands of) Web servers, after migrating the rules to the Citrix ADC you will have a single point of control for these rules.

Following are examples of mod_rewrite functions, and translations of these functions into Rewrite and Responder policies on the Citrix ADC.

Converting URL variations into canonical URLs

On some Web servers you can have multiple URLs for a resource. Although the canonical URLs should be used and distributed, other URLs can exist as shortcuts or internal URLs. You can make sure that users see the canonical URL regardless of the URL used to make an initial request.

In the following examples, the URL /~user is converted to /u/user.

Apache mod_rewrite solution for converting a URL


```
RewriteRule ^/~([^/]+)/?(.*) /u/$1/$2[R]
<!--NeedCopy-->
```

Citrix ADC solution for converting a URL

```
add responder action act1 redirect '"/u/"+HTTP.REQ.URL.AFTER_STR("/~")'
    -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.STARTSWITH("/~") && HTTP.REQ.
   URL.LENGTH.GT(2)' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Converting Host Name Variations to Canonical Host Names

You can enforce the use of a particular host name for reaching a site. For example, you can enforce the use of www.example.com instead of example.com.

Apache mod_rewrite solution for enforcing a particular host name for sites running on a port other than 80

```
RewriteCond %{
 HTTP_HOST }
    !^www.example.com
RewriteCond %{
 HTTP_HOST }
    !^$
RewriteCond %{
 SERVER_PORT }
  !^80$
RewriteRule ^/(.*)         http://www.example.com:%{
 SERVER_PORT }
 /$1 [L,R]
<!--NeedCopy-->
```

Apache mod_rewrite solution for enforcing a particular host name for sites running on port 80

```
RewriteCond %{
 HTTP_HOST }
```



```
    !^www.example.com
RewriteCond %{
 HTTP_HOST }
    !^$
RewriteRule ^/(.*)         http://www.example.com/$1 [L,R]
<!--NeedCopy-->
```

Citrix ADC solution for enforcing a particular host name for sites running on a port other than 80

```
add responder action act1 redirect '"http://www.example.com:"+CLIENT.
   TCP.DSTPORT+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com
   ")&&!HTTP.REQ.HOSTNAME.EQ("")&&!HTTP.REQ.HOSTNAME.PORT.EQ(80)&&HTTP.
   REQ.HOSTNAME.CONTAINS("example.com")' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Citrix ADC solution for enforcing a particular host name for sites running on port 80

```
add responder action act1 redirect '"http://www.example.com"+HTTP.REQ.
   URL' -bypassSafetyCheck yes
add responder policy pol1  '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.
   com")&&!HTTP.REQ.HOSTNAME.EQ("")&&HTTP.REQ.HOSTNAME.PORT.EQ(80)&&
   HTTP.REQ.HOSTNAME.CONTAINS("example.com")' act1
bind responder global  pol1 100 END
<!--NeedCopy-->
```

Moving a document root

Usually the document root of a Web server is based on the URL “/”. However, the document root can be any directory. You can redirect traffic to the document root if it changes from the top-level “/” directory to another directory.

In the following examples, you change the document root from / to /e/www. The first two examples simply replace one string with another. The third example is more universal because, along with replacing the root directory, it preserves the rest of the URL (the path and query string), for example, redirecting /example/file.html to /e/www/example/file.html.

Apache mod_rewrite solution for moving the document root


```
RewriteEngine on
RewriteRule   ^/$  /e/www/  [R]
<!--NeedCopy-->
```

Citrix ADC solution for moving the document root

```
add responder action act1 redirect '"/e/www/"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.EQ("/")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Citrix ADC solution for moving the document root and appending path information to the request

```
add responder action act1 redirect '"/e/www"+HTTP.REQ.URL' -
   bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.URL.STARTSWITH("/e/www/")' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Moving home directories to a new web server

You may want to redirect requests that are sent to home directories on a Web server to a different Web server. For example, if a new Web server is replacing an old one over time, as you migrate home directories to the new location you need to redirect requests for the migrated home directories to the new Web server.

In the following examples, the host name for the new Web server is newserver.

Apache mod_rewrite solution for redirecting to another Web server

```
RewriteRule   ^/(.+)  http://newserver/$1     [R,L]
<!--NeedCopy-->
```

Citrix ADC solution for redirecting to another Web server (method 1)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -
   bypassSafetyCheck yes
```



```
add responder policy pol1 'HTTP.REQ.URL.REGEX_MATCH(re#^/(.+)#)'   act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Citrix ADC solution for redirecting to another Web server (method 2)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -
    bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.LENGTH.GT(1)' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Working with structured home directories

Typically, a site with thousands of users has a structured home directory layout. For example, each home directory may reside under a subdirectory that is named using the first character of the user name. For example, the home directory for jsmith (/~jsmith/anypath) might be /home/j/smith/.www/anypath, and the home directory for rvalveti (/~rvalveti/anypath) might be /home/r/rvalveti/.www/anypath.

The following examples redirect requests to the home directory.

Apache mod_rewrite solution for structured home directories

```
RewriteRule   ^/~(([a-z])[a-z0-9]+)(.*)  /home/$2/$1/.www$3
<!--NeedCopy-->
```

Citrix ADC solution for structured home directories

Citrix ADC solution for structured home directories

```
add rewrite action act1 replace 'HTTP.REQ.URL'  '"/home/"+ HTTP.REQ.URL
    .AFTER_STR("~").PREFIX(1)+"/"+ HTTP.REQ.URL.AFTER_STR("~").
    BEFORE_STR("/")+"/.www"+HTTP.REQ.URL.SKIP(\'/\',1)'  -
    bypassSafetyCheck yes
add rewrite policy pol1  'HTTP.REQ.URL.PATH.STARTSWITH("/~")' act1
bind rewrite global pol1 100

<!--NeedCopy-->
```


Redirecting invalid URLs to other web servers

If a URL is not valid, it should be redirected to another Web server. For example, you should redirect to another Web server if a file that is named in a URL does not exist on the server that is named in the URL.

On Apache, you can perform this check using mod_rewrite. On the Citrix ADC, an HTTP callout can check for a file on a server by running a script on the server. In the following Citrix ADC examples, a script named file_check.cgi processes the URL and uses this information to check for the presence of the target file on the server. The script returns TRUE or FALSE, and the Citrix ADC uses the value that the script returns to validate the policy.

In addition to performing the redirection, the Citrix ADC can add custom headers or, as in the second Citrix ADC example, it can add text in the response body.

Apache mod_rewrite solution for redirection if a URL is wrong

```
RewriteCond   /your/docroot/%{
 REQUEST_FILENAME }
  !-f
RewriteRule   ^(.+)      http://webserverB.com/$1 [R]

<!--NeedCopy-->
```

Citrix ADC solution for redirection if a URL is wrong (method 1)

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr
     '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).
    CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"'   -
    parameters query=http.req.url.path -headers Name("ddd")
add responder action act1 redirect '"http://webserverB.com"+HTTP.REQ.
    URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.
    HTTP_CALLOUT(call)' act1
bind responder global pol1 100

<!--NeedCopy-->
```

Citrix ADC solution for redirection if a URL is wrong (method 2)

```
add HTTPCallout Call
```



```
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr
     '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).
    CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"'   -
    parameters query=http.req.url.path -headers Name("ddd")
add responder  action act1 respondwith  '"HTTP/1.1 302 Moved
    Temporarily\r\nLocation: http://webserverB.com"+HTTP.REQ.URL+"\r\n\r
    \nHTTPCallout Used"' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.
    HTTP_CALLOUT(call)' act1
bind responder global pol1 100

<!--NeedCopy-->
```

Rewriting a URL based on time

You can rewrite a URL based on the time. The following examples change a request for example.html to example.day.html or example.night.html, depending on the time of day.

Apache mod_rewrite solution for rewriting a URL based on the time

```
RewriteCond   %{
 TIME_HOUR }
 %{
 TIME_MIN }
  >0700
RewriteCond   %{
 TIME_HOUR }
 %{
 TIME_MIN }
  <1900
RewriteRule   ^example\.html$ example.day.html [L]
RewriteRule   ^example\.html$ example.night.html

<!--NeedCopy-->
```

Citrix ADC solution for rewriting a URL based on the time

```
add rewrite action act1 insert_before 'HTTP.REQ.URL.PATH.SUFFIX
    (\'.\',0)' '"day."'
add rewrite action act2  insert_before 'HTTP.REQ.URL.PATH.SUFFIX
    (\'.\',0)' '"night."'
```



```
add rewrite  policy pol1 'SYS.TIME.WITHIN(LOCAL 07h 00m,LOCAL 18h 59m)'
    act1
add rewrite policy pol2 'true'  act2
bind rewrite global pol1 101
bind rewrite global pol2 102

<!--NeedCopy-->
```

Redirecting to a new file name (Invisible to the User)

If you rename a Web page, you can continue to support the old URL for backward compatibility while preventing users from recognizing that the page was renamed.

In the first two of the following examples, the base directory is /~quux/. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for managing a file name change in a fixed location

```
RewriteEngine  on
RewriteBase    /~quux/
RewriteRule    ^foo\.html$  bar.html

<!--NeedCopy-->
```

Citrix ADC solution for managing a file name change in a fixed location

```
add rewrite action act1 replace 'HTTP.REQ.URL.AFTER_STR("/~quux").
   SUBSTR("foo.html")' '"bar.html"'
add rewrite policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/foo.html")' act1
bind rewrite global pol1 100

<!--NeedCopy-->
```

Citrix ADC solution for managing a file name change regardless of the base directory or query strings in the URL

```
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"
   bar.html"'
Add rewrite policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("foo.html")' act1
Bind rewrite global pol1 100
```



```
4
5 <!--NeedCopy-->
```

Redirecting to new file name (user-visible URL)

If you rename a Web page, you may want to continue to support the old URL for backward compatibility and allow users to see that the page was renamed by changing the URL that is displayed in the browser.

In the first two of the following examples, redirection occurs when the base directory is /~quux/. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for changing the file name and the URL displayed in the browser

```
1 RewriteEngine on
2 RewriteBase    /~quux/
3 RewriteRule    ^old\.html$ new.html  [R]
4
5 <!--NeedCopy-->
```

Citrix ADC solution for changing the file name and the URL displayed in the browser

```
1 add responder action act1 redirect 'HTTP.REQ.URL.BEFORE_STR("foo.html")
     +"new.html"' -bypassSafetyCheck yes
2 add responder policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/old.html")'
     act1
3 bind responder global pol1 100
4
5 <!--NeedCopy-->
```

Citrix ADC solution for changing the file name and the URL displayed in the browser regardless of the base directory or query strings in the URL

```
1 add responder action act1 redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("old.
     html")+"new.html"+HTTP.REQ.URL.AFTER_STR("old.html")' -
     bypassSafetyCheck yes
2 add responder policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("old.html")' act1
3 bind responder global pol1 100
4
```



```
<!--NeedCopy-->
```

Accommodating browser dependent content

To accommodate browser-specific limitations—at least for important top-level pages—it is sometimes necessary to set restrictions on the browser type and version. For example, you might want to set a maximum version for the latest Netscape variants, a minimum version for Lynx browsers, and an average feature version for all others.

The following examples act on the HTTP header “User-Agent”, such that if this header begins with “Mozilla/3”, the page MyPage.html is rewritten to MyPage.NS.html. If the browser is “Lynx” or “Mozilla” version 1 or 2, the URL becomes MyPage.20.html. All other browsers receive page MyPage.32.html.

Apache mod_rewrite solution for browser-specific settings

```
RewriteCond %{
 HTTP_USER_AGENT }
   ^Mozilla/3.*
RewriteRule ^MyPage\.html$ MyPage.NS.html [L]
RewriteCond %{
 HTTP_USER_AGENT }
   ^Lynx/.* [OR]
RewriteCond %{
 HTTP_USER_AGENT }
   ^Mozilla/[12].*
RewriteRule ^MyPage\.html$ MyPage.20.html [L]
RewriteRule ^fMyPage\.html$ MyPage.32.html [L]
Citrix ADC solution for browser-specific settings
add patset pat1
bind patset pat1 Mozilla/1
bind Patset pat1 Mozilla/2
bind patset pat1 Lynx
bind Patset pat1 Mozilla/3
add rewrite action act1 insert_before 'HTTP.REQ.URL.SUFFIX' '"NS."'
add rewrite action act2 insert_before 'HTTP.REQ.URL.SUFFIX' '"20."'
add rewrite action act3 insert_before 'HTTP.REQ.URL.SUFFIX' '"32."'
add rewrite policy pol1 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX
   ("pat1").EQ(4)' act1
add rewrite policy pol2 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX
   ("pat1").BETWEEN(1,3)' act2
add rewrite policy pol3 '!HTTP.REQ.HEADER("User-Agent").STARTSWITH_ANY
   ("pat1")' act3
bind rewrite global pol1 101 END
```



```
bind rewrite global pol2 102 END
bind rewrite global pol3 103 END

<!--NeedCopy-->
```

Blocking access by robots

You can block a robot from retrieving pages from a specific directory or a set of directories to ease up the traffic to and from these directories. You can restrict access based on the specific location or you can block requests based on information in User-Agent HTTP headers.

In the following examples, the Web location to be blocked is /~quux/foo/arc/, the IP addresses to be blocked are 123.45.67.8 and 123.45.67.9, and the robot's name is NameOfBadRobot.

Apache mod_rewrite solution for blocking a path and a User-Agent header

```
RewriteCond %{
 HTTP_USER_AGENT }
    ^NameOfBadRobot.*
RewriteCond %{
 REMOTE_ADDR }
        ^123\.45\.67\.[8-9]$
RewriteRule ^/~quux/foo/arc/.+   -   [F]

<!--NeedCopy-->
```

Citrix ADC solution for blocking a path and a User-Agent header

```
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"
    '
add responder policy pol1 'HTTP.REQ.HEADER("User_Agent").STARTSWITH("
    NameOfBadRobot")&&CLIENT.IP.SRC.EQ(123.45.67.8)&&CLIENT.IP.SRC.EQ
    (123.45.67.9) && HTTP.REQ.URL.STARTSWITH("/~quux/foo/arc")' act1
bind responder global pol1 100

<!--NeedCopy-->
```

Blocking access to inline images

If you find people frequently going to your server to copy inline graphics for their own use (and generating unnecessary traffic), you may want to restrict the browser's ability to send an HTTP Referer

header.

In the following example, the graphics are located in Example.

Apache mod_rewrite solution for blocking access to an inline image

```
RewriteCond %{
 HTTP_REFERER }
  !^$
RewriteCond %{
 HTTP_REFERER }
  !^http://www.quux-corp.de/~quux/.*$
RewriteRule .*\.png$ - [F]

<!--NeedCopy-->
```

Citrix ADC solution for blocking access to an inline image

```
add patset pat1
bind patset pat1 .png
bind patset pat1 .jpeg
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"
    '
add responder policy pol1 '!HTTP.REQ.HEADER("Referer").EQ("") && !HTTP.
    REQ.HEADER("Referer").STARTSWITH("http://www.quux-corp.de/~quux/")&&
    HTTP.REQ.URL.ENDSWITH_ANY("pat1")' act1
bind responder global pol1 100

<!--NeedCopy-->
```

Creating extensionless links

To prevent users from knowing application or script details on the server side, you can hide file extensions from users. To do this, you may want to support extensionless links. You can achieve this behavior by using rewrite rules to add an extension to all requests, or to selectively add extensions to requests.

The first two of the following examples show adding an extension to all request URLs. In the last example, one of two file extensions is added. Note that in the last example, the mod_rewrite module can easily find the file extension because this module resides on the Web server. In contrast, the Citrix ADC must invoke an HTTP callout to check the extension of the requested file on the Web server. Based on the callout response, the Citrix ADC adds the .html or .php extension to the request URL.

> **Note**
>
> In the second Citrix ADC example, an HTTP callout is used to query a script named file_check.cgi hosted on the server. This script checks whether the argument that is provided in the callout is a valid file name.

Apache mod_rewrite solution for adding a .php extension to all requests

```
RewriteRule ^/?([a-z]+)$ $1.php [L]

<!--NeedCopy-->
```

Citrix ADC policy for adding a .php extension to all requests

```
add rewrite action act1 insert_after 'HTTP.REQ.URL' '".php"'
add rewrite policy pol1 'HTTP.REQ.URL.PATH.REGEX_MATCH(re#^/([a-z]+)$#)
    ' act1
bind rewrite global pol1 100
<!--NeedCopy-->
```

Apache mod_rewrite solution for adding either .html or .php extensions to requests

```
RewriteCond %{
 REQUEST_FILENAME }
 .php -f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.php [L]
RewriteCond %{
 REQUEST_FILENAME }
 .html  - f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.html [L]
<!--NeedCopy-->
```

Citrix ADC policy for adding either .html or .php extensions to requests

```
add HTTPCallout Call_html
add HTTPCallout Call_php
set policy httpCallout Call_html -IPAddress 10.102.59.101 -port 80 -
    hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.
    BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"
    '   -parameters query=http.req.url+".html"
```

```
set policy httpCallout Call_php -IPAddress 10.102.59.101 -port 80 -
    hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.
    BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"
    ' -parameters query=http.req.url+".php"
add patset pat1
bind patset pat1 .html
bind patset pat1 .php
bind patset pat1 .asp
bind patset pat1 .cgi
add rewrite  action act1 insert_after 'HTTP.REQ.URL.PATH'  '".html"'
add rewrite  action act2 insert_after "HTTP.REQ.URL.PATH"  '".php"'
add rewrite policy pol1 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.
    HTTP_CALLOUT(Call_html)' act1
add rewrite policy pol2 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.
    HTTP_CALLOUT(Call_php)' act2
bind rewrite global pol1 100 END
bind rewrite global pol2 101 END

<!--NeedCopy-->
```

Redirecting a Working URI to a New Format

Suppose that you have a set of working URLs that resemble the following:

```
/index.php?id=nnnn

<!--NeedCopy-->
```

To change these URLs to /nnnn and make sure that search engines update their indexes to the new URI format, you need to do the following:

- Redirect the old URIs to the new ones so that search engines update their indexes.
- Rewrite the new URI back to the old one so that the index.php script runs correctly.

To accomplish this, you can insert marker code into the query string (making sure that the marker code is not seen by visitors), and then removing the marker code for the index.php script.

The following examples redirect from an old link to a new format only if a marker is not present in the query string. The link that uses the new format is re-written back to the old format, and a marker is added to the query string.

Apache mod_rewrite solution

```
RewriteCond %{
```



```
 QUERY_STRING }
  !marker
RewriteCond %{
 QUERY_STRING }
  id=([-a-zA-Z0-9_+]+)
RewriteRule ^/?index\.php$ %1? [R,L]
RewriteRule ^/?([-a-zA-Z0-9_+]+)$  index.php?marker&id=$1 [L]
Citrix ADC solution
add responder action act_redirect redirect 'HTTP.REQ.URL.PATH.
   BEFORE_STR("index.php")+HTTP.REQ.URL.QUERY.VALUE("id")' -
   bypassSafetyCheck yes
add responder policy pol_redirect '!HTTP.REQ.URL.QUERY.CONTAINS("marker
   ")&& HTTP.REQ.URL.QUERY.VALUE("id").REGEX_MATCH(re/[-a-zA-Z0-9_+]+/)
    && HTTP.REQ.URL.PATH.CONTAINS("index.php")' act_redirect
bind responder global pol_redirect 100 END
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"
   index.phpmarker&id="+HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' -
   bypassSafetyCheck yes
add rewrite policy pol1 '!HTTP.REQ.URL.QUERY.CONTAINS("marker")'  act1
bind rewrite global pol1 100 END

<!--NeedCopy-->
```

Ensuring That a Secure Server Is Used for Selected Pages

To make sure that only secure servers are used for selected Web pages, you can use the following Apache mod_rewrite code or Citrix ADC Responder policies.

Apache mod_rewrite solution

```
RewriteCond %{
 SERVER_PORT }
  !^443$
RewriteRule ^/?(page1|page2|page3|page4|page5)$  https://www.example.
   com/%1 [R,L]

<!--NeedCopy-->
```

Citrix ADC solution using regular expressions

```
add responder action res_redirect redirect  '"https://www.example.com"+
   HTTP.REQ.URL' -bypassSafetyCheck yes
```



```
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.
    REQ.URL.REGEX_MATCH(re/page[1-5]/)'  res_redirect
bind responder global pol_redirect 100 END

<!--NeedCopy-->
```

Citrix ADC solution using pattern sets

```
add patset pat1
bind patset pat1 page1
bind patset pat1 page2
bind patset pat1 page3
bind patset pat1 page4
bind patset pat1 page5
add responder action res_redirect redirect  '"https://www.example.com"+
    HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.
    REQ.URL.CONTAINS_ANY("pat1")'  res_redirect
bind responder global pol_redirect 100 END

<!--NeedCopy-->
```

Rewrite and responder policy examples

September 14, 2021

Following are some examples for rewrite and responder policies:

Example 1: To add a local Client-IP header by using the command line interface

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.
    IP.SRC'
add rewrite policy pol_ins_client http.req.is_valid act_ins_client
bind rewrite global pol_ins_client 300 END

namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
```



```
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:06:48 GMT
* Server Apache/2.2.15 (CentOS) is not blacklisted
< Server: Apache/2.2.15 (CentOS)
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
< NS-Client: 10.102.1.98
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$

<!--NeedCopy-->
```

Example 2: Mask the HTTP Server Type

```
add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server") "\"Web Server 1.0\""
add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:15:42 GMT
* Server Web Server 1.0 is not blacklisted
< Server: Web Server 1.0
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
```



```
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$
<!--NeedCopy-->
```

Example 3: Respond by redirecting to different url when a url is received

```
> add responder action act1 redirect "\"www.google.com\""
 Done
> add responder policy pol1 'HTTP.REQ.URL.CONTAINS("file")' act1
 Done
> bind responder global pol1 1
 Done
>

name:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: */*
>
< HTTP/1.1 302 Found : Moved Temporarily
< Location: www.google.com
< Connection: close
< Cache-Control: no-cache
< Pragma: no-cache
<
* Closing connection 0
name@obelix:~$
<!--NeedCopy-->
```

Example 4: Respond with a message which can be any expression or a text

```
add responder action act123 respondwith "\"Please reach out to
   administrator\""
add responder policy pol1 "HTTP.REQ.URL.CONTAINS(\"file\")" act123
bind responder global pol1 100 END

name@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
```



```
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10..Responder Action and Policy:

>add responder action Redirect-Action redirect "\"https://xyz.abc.com/dispatcher/SAML2AuthService?siteurl=wmav\"" -responseStatusCode 302

>add responder policy Redirect-Policy "HTTP.REQ.HOSTNAME.CONTAINS(\"abc\")" Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -gotoPriorityExpression END -type REQUEST.
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: */*
>
* Connection #0 to host 10.10.10.10 left intact
Please reach out to administratort_name@obelix:~$
<!--NeedCopy-->
```

Example 5: Respond with an HTML imported page

```
import responder htmlpage http://10.10.10.10)/testsite/file5.html page112
add responder action act1 respondwithHtmlpage page1
add responder policy pol1 true act1
bind responder global pol1 100

name@obelix:~$ curl -v http://10.10.10.10)/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10)) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.102.58.140
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Length: 5
< Content-Type: text/html
<
```



```
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_name@obelix:~$
<!--NeedCopy-->
```

Example 6: Redirect URL based on HOSTNAME using Responder Policy

```
Responder Action and Policy:

>add responder action Redirect-Action redirect "\"https://xyz.abc.com/dispatcher/SAML2AuthService?siteurl=wmav\"" -responseStatusCode 302

>add responder policy Redirect-Policy "HTTP.REQ.HOSTNAME.CONTAINS(\"abc\")" Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

Rate limiting

September 14, 2021

The rate limiting feature enables you to define the maximum load for a given network entity or virtual entity on the Citrix ADC appliance. The feature enables you to configure the appliance to monitor the rate of traffic associated with the entity and take preventive action, in real time, based on the traffic rate. This feature is particularly useful when the network is under attack from a hostile client that is sending the appliance a flood of requests. You can mitigate the risks that affect the availability of resources to clients, and you can improve the reliability of the network and the resources that the appliance manages.

You can monitor and control the rate of traffic that is associated with virtual and user-defined entities, including virtual servers, URLs, domains, and combinations of URLs and domains. You can throttle the rate of traffic if it is too high, base information caching on the traffic rate, and redirect traffic to a given load balancing virtual server if the traffic rate exceeds a predefined limit. You can apply rate-based monitoring to HTTP, TCP, and DNS requests.

To monitor the rate of traffic for a given scenario, you configure a *rate limit identifier*. A rate limit identifier specifies numeric thresholds such as the maximum number of requests or connections (of a particular type) that are permitted in a specified time period called a *time slice*.

Optionally, you can configure filters, known as *stream selectors*, and associate them with rate limit identifiers when you configure the identifiers. After you configure the optional stream selector and the limit identifier, you must invoke the limit identifier from a default syntax policy. You can invoke identifiers from any feature in which the identifier may be useful, including rewrite, responder, DNS, and integrated caching.

You can globally enable and disable SNMP traps for rate limit identifiers. Each trap contains cumulative data for the rate limit identifier's configured data collection interval (time slice), unless you specified multiple traps to be generated per time slice. For more information about configuring SNMP traps and managers, see SNMP.

Configuring a Stream Selector

September 14, 2021

A traffic stream selector is an optional filter for identifying an entity for which you want to throttle access. The selector is applied to a request or a response and selects data points (keys) that can be analyzed by a rate stream identifier. These data points can be based on almost any characteristic of the traffic, including IP addresses, subnets, domain names, TCP or UDP identifiers, and particular strings or extensions in URLs.

A stream selector consists of individual default syntax expressions called selectlets. Each selectlet is a non-compound default syntax expression. A traffic stream selector can contain up to five non-compound expressions called selectlets. Each selectlet is considered to be in an AND relationship with the other expressions. Following are some examples of selectlets:

```
http.req.url
http.res.body(1000>after_str("car_model").before_str("made_in")"
"client.ip.src.subnet(24)"
<!--NeedCopy-->
```

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the Citrix ADC considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the Citrix ADC, again, can count the same transaction more than once.

Note: If you modify an expression in a stream selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a stream selector named myStreamSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myStreamSelector1, you might receive an error

when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

To configure a traffic stream selector by using the command line interface

At the command prompt, type:

```
add stream selector <name> <rule> ...
<!--NeedCopy-->
```

Example:

```
add stream selector myStreamSel HTTP.REQ.URL CLIENT.IP.SRC
<!--NeedCopy-->
```

To configure a stream selector by using the configuration utility

Navigate to AppExpert > Rate Limiting > Selectors, click Add and specify the relevant details.

Configuring a Traffic Rate Limit Identifier

September 14, 2021

A rate limit identifier checks if the amount of traffic exceeds a specified value, within a particular time interval. The identifier returns a “Boolean TRUE” if the amount of traffic exceeds a limit within a particular time interval. When you include a limit identifier in the compound default syntax expression in a policy rule, you must include a stream selector. If you do not specify, the limit identifier is applied to all requests or responses identified by the compound expressions.

> **Note:**
>
> The maximum length for storing string results (for example, HTTP.REQ.URL) is 60 characters. If the string (for example, URL) is 1000 characters long, out of which 50 characters are long enough to uniquely identify a string, you can use an expression to extract required 50 characters.

To configure a traffic limit identifier from the command line interface

At the command prompt, type:


```
add ns limitIdentifier <limitIdentifier> -threshold <positive_integer>
    -timeSlice <positive_integer> -mode <mode> -limitType ( BURSTY |
    SMOOTH ) -selectorName <string> -maxBandwidth <positive_integer> -
    trapsInTimeSlice <positive_integer>
<!--NeedCopy-->
```

Argument description

limitIdentifier. Name for a rate limit identifier. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Reserved words must not be used. This is a mandatory argument. Maximum Length: 31

threshold. A maximum number of requests that are allowed in the given timeslice when requests (mode is set as REQUEST_RATE) are tracked per timeslice. When connections (mode is set as CONNECTION) are tracked, it is the total number of connections that would be let through. Default value: 1 Minimum value: 1 Maximum Value: 4294967295

timeSlice. Time interval, in milliseconds, specified in multiples of 10, during which requests are tracked to check if they cross the threshold. This argument is needed only when the mode is set to REQUEST_RATE. Default value: 1000 Minimum value: 10 Maximum Value: 4294967295

mode. Defines the type of traffic to be tracked.

1. REQUEST_RATE. Tracks requests/timeslice.
2. CONNECTION. Tracks active transactions.

limitType. Smooth or bursty request type.

selectorName. Name of the rate limit selector. If this argument is NULL, rate limiting will be applied on all traffic received by the virtual server or the Citrix ADC (depending on whether the limit identifier is bound to a virtual server or globally) without any **filtering. Maximum Length: 31**

maxBandwidth. Maximum bandwidth permitted, in kbps. Minimum value: 0 Maximum value: 4294967287

Example:

Configuring traffic rate limit identifier in BURSTY mode:

```
add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000
     -mode REQUEST_RATE -limitType BURSTY -selectorName
    limit_100_requests_selector -trapsInTimeSlice 30
<!--NeedCopy-->
```

Configuring traffic rate limit identifier in SMOOTH mode:


```
add ns limitidentifier limit_req -mode request_rate -limitType smooth -
    timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200
<!--NeedCopy-->
```

To configure a traffic limit identifier by using the configuration utility

Navigate to AppExpert > Rate Limiting > Limit Identifiers, click Add and specify the relevant details.

Configuring and Binding a Traffic Rate Policy

September 14, 2021

You implement rate-based application behavior by configuring a policy in an appropriate Citrix ADC feature. The feature must support default syntax policies. The policy expression must contain the following expression prefix to enable the feature to analyze the traffic rate:

```
sys.check_limit(<limit_identifier>)
<!--NeedCopy-->
```

Where limit_identifier is the name of a limit identifier.

The policy expression must be a compound expression that contains at least two components:

- An expression that identifies traffic to which the rate limit identifier is applied. For example:

```
    http.req.url.contains("my_aspx.aspx").
<!--NeedCopy-->
```

- An expression that identifies a rate limit identifier, for example, sys.check_limit(“my_limit_identifier”). This must be the last expression in the policy expression.

To configure a rate-based policy by using the command line interface

At the command prompt, type the following command to configure a rate-based policy and verify the configuration:

```
add cache|dns|rewrite|responder policy <policy_name> -rule expression
    && sys.check_limit("<LimitIdentifierName>") [<feature-specific
    information>]
<!--NeedCopy-->
```


Following is a complete example of a rate-based policy rule. Note that this example assumes that you have configured the responder action, send_direct_url, that is associated with the policy. Note that the sys.check_limit parameter must be the last element of the policy expression:

```
add responder policy responder_threshold_policy "http.req.url.contains(
    "myindex.html") && sys.check_limit("my_limit_identifier")"
    send_direct_url
<!--NeedCopy-->
```

For information about binding a policy globally or to a virtual server, see "Binding Default Syntax Policies."

To configure a rate-based policy by using the configuration utility

1. In the navigation pane, expand the feature in which you want to configure a policy (for example, Integrated Caching, Rewrite, or Responder), and then click Policies.
2. In the details pane, click Add. In Name, enter a unique name for the policy.
3. Under Expression, enter the policy rule, and make sure that you include the sys.check_limit parameter as the final component of the expression. For example:

   ```
   http.req.url.contains("my_aspx.aspx") && sys.check_limit("
       my_limit_identifier")
   <!--NeedCopy-->
   ```

4. Enter feature-specific information about the policy.

   For example, you may be required to associate the policy with an action or a profile. For more information, see the feature-specific documentation.

5. Click Create, and then click Close.
6. Click Save.

Viewing the Traffic Rate

September 14, 2021

If traffic through one or more virtual servers matches a rate-based policy, you can view the rate of this traffic. The rate statistics are maintained in the limit identifier that you named in the rule for the rate-based policy. If more than one policy uses the same limit identifier, you can view the traffic rate as defined by hits to all of the policies that use the particular limit identifier.

To view the traffic rate by using the command line interface

At the command prompt, type the following command to view the traffic rate:

```
show ns limitSessions <limitIdentifier>
<!--NeedCopy-->
```

Example:

```
sh limitsession myLimitSession
<!--NeedCopy-->
```

To view the traffic rate by using the configuration utility

1. Navigate to AppExpert > Rate Limiting > Limit Identifiers.
2. Select a limit identifier whose traffic rate you want to view.
3. Click the Show Sessions button. If traffic through one or more virtual servers has matched a rate limiting policy that uses this limit identifier (and the hits are within the configured time slice for this identifier), the Session Details dialog box appears. Otherwise, you receive a “No session exists” message.

Testing a Rate-Based Policy

September 14, 2021

To test a rate-based policy, you can send traffic to any virtual server to which a rate-based policy is bound.

Task overview: Testing a rate-based policy

1. Configure a stream selector (optional) and a rate limit identifier (required). For example:

```
add stream selector sel_subnet Q.URL "CLIENT.IP.SRC.SUBNET(24)"
add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode
    REQUEST_RATE  -limittype smooth  -selectorName sel_subnet  -
   trapsInTimeSlice 8
<!--NeedCopy-->
```

2. Configure the action that you want to associate with the policy that uses the rate limit identifier. For example:


```
add responder action resp_redirect redirect "\"http://
   response_site.com/\""
<!--NeedCopy-->
```

3. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier. For example, the policy can apply a rate limit identifier to all requests arriving from a particular subnet, as follows:

```
add responder policy resp_subnet "SYS.CHECK_LIMIT(\"k_subnet\")"
   resp_redirect
<!--NeedCopy-->
```

4. Bind the policy globally or to a virtual server. For example:

```
bind responder global resp_subnet 6 END -type DEFAULT
<!--NeedCopy-->
```

5. In a browser address bar, send a test HTTP query to a virtual server. For example:

```
http://<IP of a vserver>/testsite/test.txt
<!--NeedCopy-->
```

6. At the Citrix ADC command prompt, type:

```
show ns limitSessions \<limitIdentifier\>
<!--NeedCopy-->
```

Example

```
> sh limitsession k_subnet
1)      Time Remaining:       98 secs  Hits: 2
                            Action Taken: 0
        Total Hash:    1718618  Hash String: /test.txt
        IPs gathered:
                1) 10.217.253.0
        Active Transactions: 0
 Done
>
<!--NeedCopy-->
```

7. Repeat the query and check the limit identifier statistics again to verify that the statistics are being updated correctly.

Examples of Rate-Based Policies

September 14, 2021

The following table shows examples of rate-based policies.

Table 1. Examples of Rate-Based Policies

Purpose	Example

Limit the number of requests per second from a URL	`add stream selector ipStreamSelector http.req.url "client.ip.src"`
	`add ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000`
	`-mode request_rate -limitType smooth -selectorName ipStreamSelector`
	
	`add responder action myWebSiteRedirectAction redirect`
	`"\"http://www.mycompany.com/\""`
	
	`add responder policy ipLimitResponderPolicy "http.req.url.contains(\"myasp.asp\")`
	`&& sys.check_limit(\"ipLimitIdentifier\")" myWebSiteRedirectAction`
	
	`bind responder global ipLimitResponderPolicy 100 END -type default`

Cache a response if the request URL rate exceeds 5 per 20000 milliseconds	`add stream selector cacheStreamSelector http.req.url`
	`add ns limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice 2000`
	`-selectorName cacheStreamSelector`
	
	`add cache policy cacheRateLimitPolicy -rule "http.req.method.eq(get)`
	`&& sys.check_limit(\"cacheRateLimitIdentifier\")" -action cache`
	
	`bind cache global cacheRateLimitPolicy -priority 10`

Drop a connection on the basis of cookies received in requests from www.yourcompany.com if the requests exceed the rate limit	`add stream selector reqCookieStreamSelector "http.req.cookie`
	`.value(\"mycookie\")" "client.ip.src.subnet(24)"`
	
	`add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000`
	`-selectorName reqCookieStreamSelector`
	
	`add responder action sendRedirectUrl redirect '\"http://www.mycompany.com\"`
	`+ http.req.url' -bypassSafetyCheck YES`
	
	`add responder policy rateLimitCookiePolicy`
	`"http.req.url.contains(\"www.yourcompany.com\")`
	`&& sys.check_limit(\"myLimitIdentifier\")" sendRedirectUrl`

Drop a DNS packet if the requests from a particular client IP address and DNS domain exceed the rate limit	`add stream selector dropDNSStreamSelector client.udp.dns.domain client.ip.src`
	`add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode request_rate`
	`-selectorName dropDNSStreamSelector -maxBandwidth 1 -trapsintimeslice 20`
	
	`add dns policy dnsDropOnClientRatePolicy "sys.check_limit`
	`(\"dropDNSRateIdentifier\")" -drop yes`

Limit the number of HTTP requests that arrive from the same host (with a subnet mask of 32) and that have the same destination IP address.	`add stream selector ipv6_sel "CLIENT.IPv6.src.subnet(32)" CLIENT.IPv6.dst Q.URL`
	`add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName ipv6_sel`
	`add lb vserver ipv6_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTimeout 180`
	`add responder action redirect_page redirect "\"http://redirectpage.com/\""`
	`add responder policy ipv6_resp_pol "SYS.CHECK_LIMIT(\"ipv6_id\")" redirect_page`
	`bind responder global ipv6_resp_pol 5 END -type DEFAULT`

Sample Use Cases for Rate-Based Policies

September 14, 2021

The following scenarios describe two uses of rate-based policies in global server load balancing (GSLB):

- The first scenario describes the use of a rate-based policy that sends traffic to a new data center if the rate of DNS requests exceed 1000 per second.
- In the second scenario, if more than five DNS requests arrive for a local DNS (LDNS) client within a particular period, the additional requests are dropped.

Redirecting Traffic on the Basis of Traffic Rate

In this scenario, you configure a proximity-based load balancing method, and a rate-limiting policy that identifies DNS requests for a particular region. In the rate-limiting policy, you specify a threshold of 1000 DNS requests per second. A DNS policy applies the rate limiting policy to DNS requests for the region "Europe.GB.17.London.UK-East.ISP-UK." In the DNS policy, DNS requests that exceed the rate limiting threshold, starting with request 1001 and continuing to the end of the one-second interval, are to be forwarded to the IP addresses that are associated with the region "North America.US.TX.Dallas.US-East.ISP-US."

The following configuration demonstrates this scenario:

```
add stream selector DNSSelector1 client.udp.dns.domain

add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000
    -selectorName DNSSelector1

add dns policy DNSLimitPolicy1 "client.ip.src.matches_location(\"Europe
   .GB.17.London.*.*\") &&
sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North
   America.US.TX.Dallas.*.*"

bind dns global DNSLimitPolicy1 5
<!--NeedCopy-->
```

Dropping DNS Requests on the Basis of Traffic Rate

In the following example of global server load balancing, you configure a rate limiting policy that permits a maximum of five DNS requests in a particular interval, per domain, to be directed to an LDNS

client for resolution. Any requests that exceed this rate are dropped. This type of policy can help protect the Citrix ADC from resource exploitation. For example, in this scenario, if the time to live (TTL) for a connection is five seconds, this policy prevents the LDNS from requerying a domain. Instead, it uses data that is cached on the Citrix ADC.

```
add stream selector LDNSSelector1 client.udp.dns.domain client.ip.src

add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice
    1000 -selectorName LDNSSelector1

add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(\".\") &&
    sys.check_limit(\"LDNSLimitIdentifier1\")" -drop YES

bind dns global LDNSPolicy1 6

show gslb vserver gvip

gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1.     site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2.      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3.      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
```



```
4.      www.gslbindia.com        TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
<!--NeedCopy-->
```

Rate Limiting for Traffic Domains

September 14, 2021

You can configure rate limiting for traffic domains. The following expression in the Citrix ADC expressions language for identifies traffic associated with traffic domains.

- client.traffic_domain.id

You can configure rate limiting for traffic associated with a particular traffic domain, a set of traffic domains, or all traffic domains.

For configuring rate limiting for traffic domains, you perform the following steps on a Citrix ADC appliance by using the configuration utility or the Citrix ADC command line:

1. Configure a stream selector that uses the client.traffic_domain.id expression for identifying the traffic, associated with traffic domains, to be rate limited.
2. Configure a rate limit identifier that specifies parameters such as maximum threshold for traffic to be rate limited. You also associate a stream selector to the rate limiter in this step.
3. Configure an action that you want to associate with the policy that uses the rate limit identifier.
4. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier, and associate the action with this policy.
5. Bind the policy globally.

Consider an example in which two traffic domains, with IDs 10 and 20, are configured on Citrix ADC NS1. On traffic domain 10, LB1-TD-1 is configured to load balance servers S1 and S2; LB2-TD1 is configured to load balance servers S3 and S4.

On traffic domain 20, LB1-TD-2 is configured to load balance servers S5 and S6; LB2-TD2 is configured to load balance servers S7 and S8.

The following table lists some examples of rate limiting policies for traffic domains in the example setup.

Purpose	CLI commands
Limit the number of requests to 10 per second for each of the traffic domains.	add stream selector tdratelimit-1 CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier limitidf-1 -threshold 10 -selectorName tdratelimit-1 -trapsInTimeSlice 0 add responder policy ratelimit-pol “sys.check_limit(\“limitidf-1\”)” DROP bind responder global ratelimit-pol 1
Limit the number of requests to 5 per client per second for each of the traffic domains.	add stream selector tdandclientip CLIENT.IP.SRC,CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td_limitidf -threshold 5 -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy tdratelimit-pol “sys.check_limit(\“td_limitidf\”)” DROP bind responder global tdratelimit-pol 2
Limit the number of requests sent for a particular traffic domain (for example traffic domain 10) to 30 requests every 3 seconds.	add stream selector tdratelimit CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td10_limitidf -threshold 30 -timeSlice 3000 -selectorName tdratelimit -trapsInTimeSlice 5 add responder policy td10ratelimit “client.traffic_domain.id==10 && sys.check_limit(\“td10_limitidf\”)” DROP bind responder global td10ratelimit 3
Limit the number of connections to 5 per client per second for a particular traffic domain (for example traffic domain 20).	add stream selector tdandclientip CLIENT.IP.SRC CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td20_limitidf -threshold 5 -mode CONNECTION -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy td20_ratelimit “client.traffic_domain.id==20 && sys.check_limit(\“td20_limitidf\”)” DROP bind responder global td20_ratelimit 4

Configure rate limit at packet level

September 14, 2021

You can configure a stream selector and a responder policy to collect statistics at the packet level flowing through all the connections identified by the selector. If the number of packets per second exceed the configured threshold, the policy applies the configured action (RESET or DROP). You can configure these policies for all types of virtual servers. Packets of all sizes are considered.

To configure rate limiting at packet level, perform the following tasks

1. Enable load balancing
2. Add stream selector
3. Add stream identifier
4. Add responder policy
5. Add load balancing virtual server
6. Bind responder policy

To enable load balancing feature

At the command prompt, type:

```
enable ns feature lb
<!--NeedCopy-->
```

To add a stream selector

At the command prompt, type:

```
add stream selector packetlimitselector client.ip.src client.tcp.
   srcport client.ip.dst client.tcp.dstport
<!--NeedCopy-->
```

To add a stream identifier

At the command prompt, type:

```
add stream identifier packetlimitidentifier packetlimitselector -
   interval 1
<!--NeedCopy-->
```

To enable tracking of ACK only packets

At the command prompt, type:


```
set stream identifier packetlimitidentifier -trackAckOnlyPackets
    ENABLED
<!--NeedCopy-->
```

To add a responder policy

At the command prompt, type:

```
add responder policy packet_rate_sessionpolicy "ANALYTICS.STREAM(\"
    packetlimitidentifier\").COLLECT_STATS(\"PACKET_LIMIT\", <
    max_threshold_PPS>, ACTION, 0/1)" NOOP
<!--NeedCopy-->
```

Where,

- <max_threshold_PPS> is the maximum number of packets allowed through the connection per second.
- ACTION can be DROP or RESET.
- 0 or 1 represents the limit type; 0 represents the BURSTY limit type and 1 represents the SMOOTH limit type.

Example:

```
add responder policy packet_rate_sessionpolicy "ANALYTICS.STREAM(\"
    packetlimitidentifier\").COLLECT_STATS(\"PACKET_LIMIT\", 40, RESET,
    0)" NOOP
<!--NeedCopy-->
```

To add a load balancing virtual server

At the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-lb-1 HTTP 10.102.20.200 80
<!--NeedCopy-->
```

To bind a responder policy

After the selector and the responder policy are configured, the policy can be bound globally or to the specific virtual server.

At the command prompt, type either of the following commands:


```
bind responder global <policyName> <priority> [<gotoPriorityExpression
    >] [-type <type>] [-invoke  (<labelType>  <labelName>) ]
<!--NeedCopy-->
```

OR

```
bind lb vserver <name>@  (-policyName <string>@ [-priority <
    positive_integer>]
<!--NeedCopy-->
```

Examples:

```
bind responder global packet_rate_sessionpolicy 101 END -type
    REQ_DEFAULT

bind responder global packet_rate_sessionpolicy 102 END -type

bind lb vserver v1 -policyname packet_rate_sessionpolicy -priority 10
<!--NeedCopy-->
```

Responder

September 14, 2021

> **Warning**
>
> Filter features using classic policies are deprecated and as an alternative Citrix recommends you to use the rewrite and responder features with advanced policy infrastructure.

Today's complex Web configurations often require different responses to HTTP requests that appear, on the surface, to be similar. When users request a webpage, you may want to provide a different page depending on the user geographical location, browser specification, or languages the browser accepts and the order of preference. You might want to drop the connection if the request is coming from an IP range that has been generating DDoS attacks or initiating hacking attempts.

Responder supports protocols such as TCP, DNS (UDP), and HTTP. With responder enabled on your appliance, server responses can be based on who sends the request, where it is sent from, and other criteria with security and system management implications. The feature is simple and quick to use. By avoiding the invocation of more complex features, it reduces CPU cycles and time spent in handling requests that do not require complex processing.

For handling sensitive data such as financial information, if you want to ensure that the client uses a secure connection to browse a site, you can redirect the request to a secure connection by using `https://` instead of `http://`.

To use a responder, do the following:

- Enable a responder feature on the appliance.
- Configure a responder action. The action can be to generate a custom response, redirect a request to a different webpage, or reset a connection.
- Configure a responder policy. The policy determines the requests (traffic) on which an action has to be taken.
- Bind each policy to a bind point put it into effect. A bind point refers to an entity at which the Citrix ADC appliance examines the traffic to see if it matches a policy. For example, a bind point can be a load balancing virtual server.

You can specify a default action for requests that do not match any policy, and you can bypass the safety check for actions that would otherwise generate error messages.

The Rewrite feature of Citrix ADC helps in rewriting some information in the requests or responses handled by Citrix ADC. The following section shows some differences between the two features.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting an HTTP request to new websites or webpages
- Responding with some custom response
- Dropping or resetting a connection at request level

If there is a responder policy, the Citrix ADC examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

If there is a rewrite policy, the Citrix ADC examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use a responder if you want the appliance to reset or drop a connection based on a request-based parameter. Use a responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

Enabling the Responder Feature

September 14, 2021

To use the Responder feature, you must first enable it.

To enable the responder feature by using the Citrix ADC command line:

At the command prompt, type the following commands to enable the responder feature and verify the configuration:

- `enable ns feature <feature>`
- `show ns feature`

Example:

```
enable ns feature Responder
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
 .
 .
 1)     Responder                       RESPONDER            ON
 2)     HTML Injection                  HTMLInjection        ON
 3)     Citrix ADC Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```

To enable the responder feature by using the GUI:

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes** and **Features**, click **Change advanced features**.
3. In the **Configure Advanced Features** dialog box, select the **Responder** check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)?** dialog box, click **YES**. A message appears in the status bar, stating that the feature has been enabled.

Configure responder action

September 14, 2021

After enabling the responder feature, you must configure one or more actions for handling requests. The responder supports the following types of actions:

- **Respond with**. Sends the response defined by the Target expression without forwarding the request to a web server. (The Citrix ADC appliance substitutes for and acts as a web server.) Use this type of action to manually define a simple HTML-based response. Normally the text for a Respond with action consists of a web server error code and brief HTML page.
- **Respond with SQL OK**. Sends the designated SQL OK response defined by the Target expression. Use this type of action to send an SQL OK response to an SQL query.
- **Respond with SQL Error**. Sends the designated SQL Error response defined by the Target expression. Use this type of action to send an SQL Error response to an SQL query.
- **Respond with HTML page**. Sends the designated HTML page as the response. You can choose from a drop-down list of HTML pages that were previously uploaded, or upload a new HTML page. Use this type of action to send an imported HTML page as the response. The appliance responds with a custom header in the responsewithhtmlpage responder action. You can configure up to eight custom headers.
- **Redirect**. Redirects the request to a different webpage or web server. A Redirect action can redirect requests originally sent to a "dummy" website that exists in DNS, but for which there is no actual web server, to an actual website. It can also redirect search requests to an appropriate URL. Normally, the redirection target for a Redirect action consists of a complete URL.

To configure a responder action by using the Citrix ADC command line:

Displays the current settings for the specified responder action. If no action name is provided, display a list of all responder actions currently configured on the Citrix ADC appliance, with abbreviated settings.

At the command prompt, type the following commands to configure a responder action and verify the configuration:

- `add responder action <name> <type> <target> [-bypassSafetyCheck (YES | NO)]`
- `show responder action`

Parameters:

- **Name**. Name of the responder action. Maximum Length: 127
- **type**. Type of responder action. It can be: (respondwith).

- **target**. An expression specifying what to respond with
- **htmlpage**. Option specifying to respondwith htmlpage
- **bypassSafetyCheck**. The safety check to allow unsafe expressions. **Note:** This attribute is deprecated.
- **hits**. The number of times the action has been taken.
- **referenceCount**. The number of references to the action.
- **undefHits**. The number of times the action resulted in UNDEF.
- **comment**. Any type of information about this responder action.
- **builtin**. Flag to determine whether responder action is built in or not

Example:

```
To create a responder action that displays a “Not Found” error page for URLs that do not exist:

> add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done

> show responder action

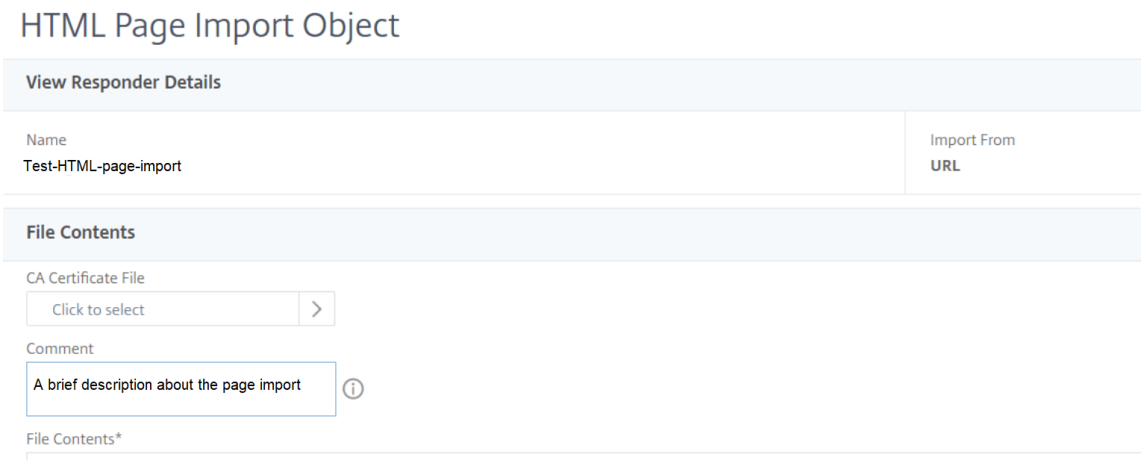
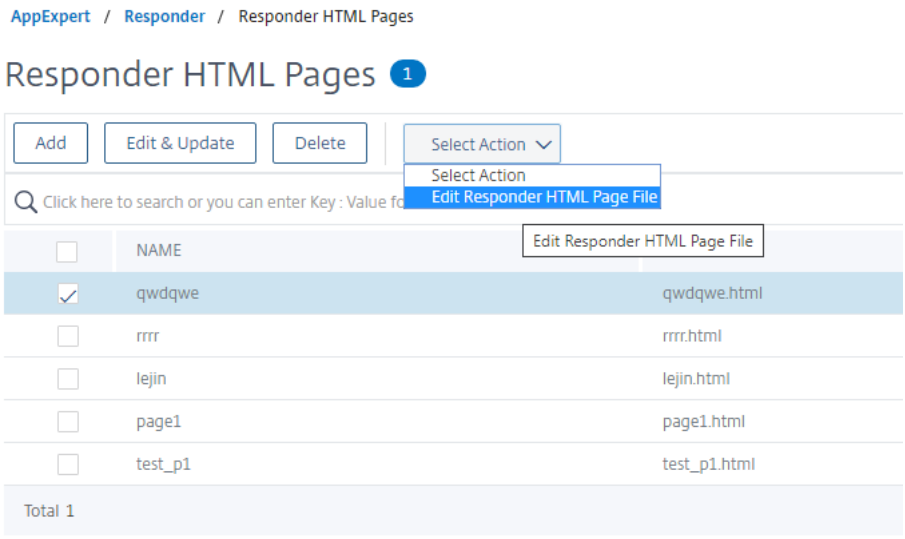
1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

To create a responder action that displays a “Not Found” error page for URLs that do not exist:

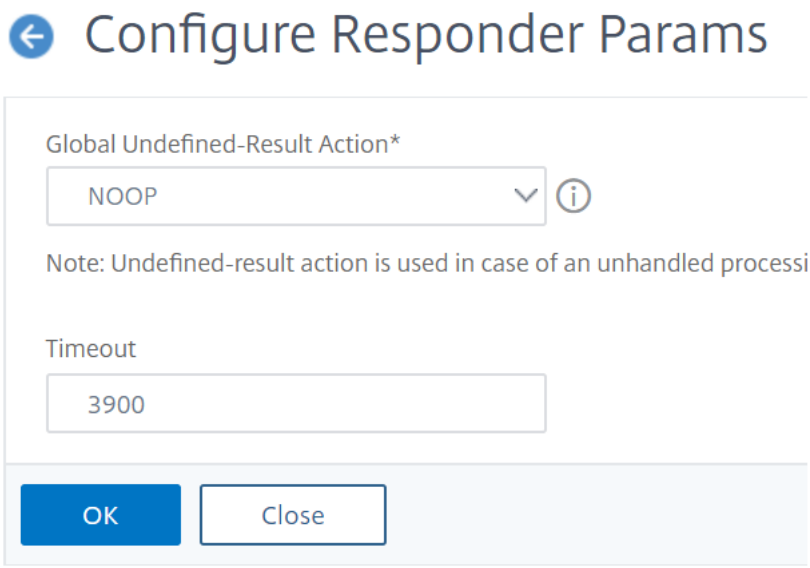
add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done
> show responder action

1) Name: act404Error
```



```
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does
   not exist on the web server."
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
<!--NeedCopy-->
```

To modify an existing responder action by using the Citrix ADC command line:

At the command prompt, type the following command to modify an existing responder action and verify the configuration:

- `set responder action <name> -target <string> [-bypassSafetyCheck ( YES | NO )]`
- `show responder action`

Example:

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\
   r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web
   server."'
 Done
> show responder action

1)      Name: act404Error
        Operation: respondwith
        Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE +
            " does not exist on the web server."
        BypassSafetyCheck : NO
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done
<!--NeedCopy-->
```

To remove a responder action by using the Citrix ADC command line:

At the command prompt, type the following command to remove a responder action and verify the configuration:

- `rm responder action <name>`
- `show responder action`

Example:


```
rm responder action act404Error
 Done

> show responder action
 Done
<!--NeedCopy-->
```

To add custom headers in responsewithhtmlpage responder action by using the Citrix ADC command line:

A Citrix ADC appliance can now respond with custom headers in the responsewithhtmlpage responder action. You can configure up to eight custom headers. Previously, the appliance responded only with `Content-type:text/html` and `Content-Length:<value>` static headers.

> **Note:**
>
> In the custom header configuration, you can also over-write the “Content-Type” header value.

At the command prompt, type the following command:

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment <string
>] [-responseStatusCode <positive_integer>] [-reasonPhrase <expression>] [-
headers <name(value)> ...]
```

Where,

name. Name for the responder action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#),space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the responder policy is added.

Type. Type of responder action. Available settings function as follows:

1. respondwith <target> - Respond to the request with the expression specified as the target.
2. respondwithhtmlpage - Respond to the request with the uploaded HTML page object specified as the target.
3. redirect - Redirect the request to the URL specified as the target.
4. sqlresponse_ok - Send an SQL OK response.
5. sqlresponse_error - Send an SQL ERROR response. This is a mandatory argument. Possible values: noop, respondwith, redirect, respondwithhtmlpage, sqlresponse_ok, sqlresponse_error

Target. Expression specifying what to respond with. Typically a URL for redirect policies or a default-syntax expression. In addition to Citrix ADC default-syntax expressions that refer to information in the request, a stringbuilder expression can contain text and HTML, and simple escape codes that define new lines and paragraphs. Enclose each stringbuilder expression element (either a Citrix ADC default-syntax expression or a string) in double quotation marks. Use the plus (+) character to join the elements.

htmlpage. For respondwithhtmlpage policies, name of the HTML page object to use as the response. You must first import the page object. Maximum Length: 31

Comment. Any type of information about this responder action. Maximum Length: 255

responseStatusCode. HTTP response status code, for example 200, 302, 404, and so forth The default value for the redirect action type is 302 and for respondwithhtmlpage is 200 Minimum value: 100 Maximum value: 599

reasonPhrase. Expression specifying the reason phrase of the HTTP response. The reason phrase may be a string literal with quotes or a PI expression. For example: "Invalid URL: " + HTTP.REQ.URL Maximum Length: 8191

Headers. One or more headers to insert into the HTTP response. Each header is specified as "name(expr)," where expr is an expression that is evaluated at runtime to provide the value for the named header. You can configure a maximum of eight headers for a responder action.

To configure a responder action by using the GUI:

1. Navigate to **AppExpert > Responder > Actions**.
2. In the details pane, do one of the following:
   - To create an action, click **Add**.
   - To modify an existing action, select the action, and then click **Open**.
3. Click **Create** or **OK**, depending on whether you are creating an action or modifying an existing action.
4. Click **Close**. A message appears in the status bar, stating that the feature has been enabled.
5. To delete a responder action, select the action, and then click **Remove**. A message appears in the status bar, stating that the feature has been disabled.

To add an expression by using the **Add Expression** dialog box

1. In the **Create Responder Action** or **Configure Responder Action** dialog box, click **Add**.
2. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.
   - HTTP. The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - SYS. One or more protected websites. Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - CLIENT. The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - ANALYTICS. The analytics data associated with the request. Choose this if you want to examine request metadata.
   - SIP. A SIP request. Choose this if you want to examine some aspect of a SIP request. When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Configuring the Global HTTP Action

You can configure the global HTTP action to invoke a responder action when an HTTP request times out. To configure this feature, you must first create the responder action that you want to invoke. Then, you configure the global HTTP timeout action to respond to a timeout with that responder action.

To configure the global HTTP action by using the Citrix ADC command line:

At the command prompt, type the following command:

- `set ns httpProfile -reqTimeoutAction <responder action name>`
- `save ns config`

For `<responder action name>`, substitute the name of the responder action.

Configure HTML page import

When a Citrix ADC appliance responds with a custom message, we can respond with an HTML file. You can import the file using the **`import`** `responder htmlpage` command and then use this file in `add responder action <act name> respondwithhtmlpage <file name>` command. You can also import the file through the Citrix ADC GUI. You can import a desired HTML page into the appliance folder and upload the page during responder run time.

Import HTML page by using the CLI

At the command prompt, type:

```
import responder htmlpage [<src>] <name> [-comment <string>] [-overwrite][-CAcertFile <string>]
```

Example:

```
import responder htmlpage http://www.example.com/page.html my-responder-page -CAcertFile my_root_ca_cert
```


Where,
CA certificate is used for verifying the client certificate. The certificate should be imported using “import ssl certfile” CLI command or equivalent through API or GUI. If certificate name is not configured, then default root CA certificates are used for the certificate verification.

Import HTML page by using the Citrix ADC GUI

1. Navigate to **AppExpert** > **Responder** > **HTML Page Imports**.
2. In the **Responder HTML Imports** details pane, click **Add**.
3. In the **HTML Page Import Object** page, set the following parameters:
   a) Name. Name of the HTML page.
   b) Import From. Imported from file, text, or text.
   c) URL. Select to enter the URL location of the HTML file.
   d) File. Select the HTML file from the appliance directory.
   e) Text. Select the HTML file as a text.
4. Click **Continue**.
5. Verify responder HTML page details.
6. Click **Done**.



To edit an HTML page, you can select a file and click **Edit Responder HTML Page File** from the **Select Action** drop-down list.



Configuring a responder policy

September 14, 2021

After you configure a responder action, you must next configure a responder policy to select the requests to which the Citrix ADC appliance should respond. A responder policy is based on a rule, which consists of one or more expressions. The rule is associated with an action, which is performed if a request matches the rule.

Note: For creating and managing responder policies, the GUI provides assistance that is not available at the Citrix ADC command prompt.

To configure a responder policy by using the Citrix ADC command line:

At the command prompt, type:

- `add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction <actionName>`
- `show responder policy <name>`

Example:

```
> add responder policy policyThree "CLIENT.IP.SRC.IN_SUBNET
   (222.222.0.0/16)" RESET
 Done
> show responder policy policyThree

        Name: policyThree
        Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
```



```
7          Responder Action: RESET
8          UndefAction: Use Global
9          Hits: 0
10         Undef Hits: 0
11  Done
12 <!--NeedCopy-->
```

To modify an existing responder policy by using the Citrix ADC command line:

At the command prompt, type:

- `set responder policy <name> [-rule <expression>] [-action <string>] [-undefAction <string>]`
- `show responder policy <name>`

To remove a responder policy by using the Citrix ADC command line:

At the command prompt, type:

- `rm responder policy <name>`
- `show responder policy`

Example:

```
1  >rm responder policy pol404Error
2   Done
3
4  > show responder policy
5   Done
6  <!--NeedCopy-->
```

To configure a responder policy by using the GUI:

1. Navigate to **AppExpert > Responder > Policies**.
2. In the details pane, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Open**.
3. Click **Create** or **OK**, depending on whether you are creating a new policy or modifying an existing policy.
4. Click **Close**. A message appears in the status bar, stating that the feature has been configured.

Binding a Responder Policy

September 14, 2021

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the Citrix ADC, or to a specific virtual server, so that the policy applies only to requests whose destination IP address is the VIP of that virtual server.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the Citrix ADC operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The responder feature implements only the first policy that a request matches, not any additional policies that it might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the Citrix ADC, see Policies and Expressions.

> **Note:**
>
> Responder policies are bound to TCP-based virtual servers.

To globally bind a responder policy by using the Citrix ADC command line:

At the command prompt, type the following command to globally bind a responder policy and verify the configuration:

- `bind responder global <policyName> <priority> [<gotoPriorityExpression> [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show responder global`

Example:

```
> bind responder global poliError 100
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
<!--NeedCopy-->
```

To bind responder policy to a specific virtual server by using the Citrix ADC command line:

At the command prompt, type:

- bind lb vserver <name> -policyname <policy_name> -priority <priority>
- sh lb vserver <name>

Example:

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
2)      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
        Time since last state change: 2 days, 00:00:04.260
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

To globally bind a responder policy by using the GUI:

1. Navigate to **AppExpert** > **Responder** > **Policies**.
2. On the **Responder Policies** page, select a responder policy, and then click **Policy Manager**.
3. In the **Responder Policy Manager** dialog box Bind Points menu, select Default Global.

4. Click **Insert Policy** to insert a new row and display a drop-down list of all unbound responder policies.
5. Click one of the policies on the list. That policy is inserted into the list of globally bound responder policies.
6. Click **Apply Changes**.
7. Click **Close**. A message appears in the status bar, stating that the configuration has been successfully completed.

To bind a responder policy to a specific virtual server by using the GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. On the **Load Balancing Virtual Servers** page, select the virtual server to which you want to bind the responder policy, and then click **Open**.
3. In the **Configure Virtual Server** (Load Balancing) dialog box, select the **Policies** tab, which displays a list of all policies configured on your Citrix ADC appliance.
4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click **OK**. A message appears in the status bar, stating that the configuration has been successfully completed.

Setting the Default Action for a Responder Policy

September 14, 2021

The Citrix ADC appliance generates an undefined event (UNDEF event) when a request does not match a responder policy. The appliance then performs the default action assigned to undefined events. By default, the action forwards the request to the next feature such as load balancing, content filtering and so forth. This default behavior ensures the requests do not require any specific responder action to be sent to your Web servers. Also, the clients receive access to the content that they have requested.

If one or more websites your Citrix ADC appliance protects receive a significant number of invalid or malicious requests, however, you might want to change the default action to either reset the client connection or drop the request. In this type of configuration, you would write one or more responder policies that would match any legitimate requests, and simply redirect those requests to their original destinations. Your Citrix ADC appliance would then block any other requests as specified by the default action you configured.

You can assign any one of the following actions to an undefined event:

- **NOOP**. The NOOP action aborts responder processing but does not alter the packet flow. So that the appliance continues to process requests that do not match any responder policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks

or redirects the request. This action is appropriate for normal requests to your Web servers and is the default setting.

- **RESET**. If the undefined action is set to RESET, the appliance resets the client connection, informing the client that it must re-establish its session with the Web server. The action is appropriate for repeat requests for webpages that do not exist, or for connections that might be attempts to hack or probe your protected websites.
- **DROP**. If the undefined action is set to DROP, the appliance silently drops the request without responding to the client in any way. This action is appropriate for requests that appear to be part of a DDoS attack or other sustained attack on your servers.

Note: UNDEF events are triggered only for client requests. No UNDEF events are triggered for responses.

To set the undefined action by using the Citrix ADC command line:

At the command prompt, type the following command to set the undefined action and verify the configuration:

- `set responder param -undefAction (RESET|DROP|NOOP)[-timeout <msecs>]`
- `show responder param`

Where,

timeout - Maximum time in milliseconds to allow for processing all the policies and their selected actions without interruption. If the timeout is reached then the evaluation causes an UNDEF to be raised and no further processing is performed.

Minimum value: 1

Maximum value: 5000

Example:

```
>set responder param -undefAction RESET -timeout 3900
 Done
> show responder param
Action Name: RESET
Timeout: 3900
 Done
>
<!--NeedCopy-->
```

Set the undefined action by using the GUI

1. Navigate to **AppExpert** > **Responder**, and then under **Settings**, click the **Change Responder Settings** link.

2. In the **Set Responder Params** page, set the following parameters:
    a) Global Undefined-Result Action. Undefined-result action is preferred in an unhandled processing exception in the responder policies and actions. Select **NOOP**, **RESET**, or **DROP**.
    b) Timeout. Maximum time in milliseconds to allow for processing all the policies and their selected actions without interruption. If the timeout is reached then the evaluation causes an UNDEF to be raised and no further processing is performed.
3. Click **OK**.



Responder Action and Policy Examples

September 14, 2021

Responder actions and policies are powerful and complex, but you can get started with relatively simple applications.

Example: Blocking Access from Specified IPs

The following procedures block access to your protected Web site(s) by clients originating from the CIDR 222.222.0.0/16. The responder sends an error message stating that the client is not authorized to access the URL requested.

To block access by using the Citrix ADC command line:

At the command prompt, type the following commands to block access:

- add responder action act_unauthorized respond with “HTTP/1.1 403 Forbidden\r\n\r\n” + “Client: “ + CLIENT.IP.SRC + “ is not authorized to access URL:” + “HTTP.REQ.URL.HTTP_URL_SAFE”’
- add responder policy pol_un “CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)” act_unauthorized
- bind responder global pol_un 10

To block access by using the GUI:

1. In the navigation pane, expand **Responder**, and then click **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, do the following:
   a) In the **Name** text box, type act_unauthorized.
   b) Under Type, select Respond with.
   c) In the Target text area, type the following string: “HTTP/1.1 200 OK\r\n\r\n” + “Client: “ + CLIENT.IP.SRC + “ is not authorized to access URL:” + HTTP.REQ.URL.HTTP_URL_SAFE
   d) Click **Create**, and then click **Close**.
   The responder action you configured, named act_unauthorized, now appears in the **Responder Actions** page.
4. In the navigation pane, click **Policies**.
5. In the details pane, click **Add**.
6. In the **Create Responder Policy** dialog box, do the following:
   a) In the Name text box, type pol_unauthorized.
   b) Under **Action**, select act_unauthorized.
   c) In the **Expression** window, type the following rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
   d) Click **Create**, then click **Close**.
   The responder policy you configured, named pol_unauthorized, now appears in the **Responder Policies** page.
7. Globally bind your new policy, pol_unauthorized, as described in Binding a Responder Policy.

Example: Redirecting a client to a new URL

The following procedures redirect clients who access your protected Web site(s) from within the CIDR 222.222.0.0/16 to a specified URL.

To redirect clients by using the Citrix ADC command line:

At the command prompt, type the following commands to redirect clients and verify the configuration:

- add responder action act_redirect redirect "<http://www.example.com/404.html>"
- show responder action act_redirect
- add responder policy pol_redirect “CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)” act_redirect
- show responder policy pol_redirect
- bind responder global pol_redirect 10

Example:


```
> add responder action act_redirect redirect `" http ://www.example.com
    /404.html "`
Done

> add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET
    (222.222.0.0/16)" act_redirect
Done
<!--NeedCopy-->
```

To redirect clients by using the GUI:

1. Navigate to **AppExpert** > **Responder** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, do the following:
   a) In the **Name** text box, type act_redirect.
   b) Under Type, select **Redirect**.
   c) In the **Target** text area, type the following string: `"<http://www.example.com/404.html>"`
   d) Click **Create**, then click **Close**.
   The responder action you configured, named act_redirect, now appears in the **Responder Actions** page.
4. In the navigation pane, click **Policies**.
5. In the details pane, click **Add**.
6. In the **Create Responder Policy** dialog box, do the following:
   a) In the **Name** text box, type pol_redirect.
   b) Under **Action**, select act_redirect.
   c) In the **Expression** window, type the following rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
   d) Click **Create**, then click **Close**.
   The responder policy you configured, named pol_redirect, now appears in the **Responder Policies** page.
7. Globally bind your new policy, pol_redirect, as described in Binding a Responder Policy.

Diameter Support for Responder

September 14, 2021

The Responder feature now supports the Diameter protocol. You can configure Responder to respond to Diameter requests as it does HTTP and TCP requests. For example, you could configure Responder to respond to requests from a specific Diameter origin with a redirect to a web page enhanced for mobile devices. A number of Citrix ADC expressions have been added that support examination of the

Diameter header and the attribute-value pairs (AVPs). These expressions support lookup of specific AVPs by index, ID or name, examine the information in each AVP, and send an appropriate response.

To configure Responder to respond to a Diameter request:

At the command prompt, type the following commands:

- `add responder action <actname> RESPONDWITH "DIAMETER.NEW_REDIRECT(\"aaa://host.example.com\")"`

  For `<actname>`, substitute a name for your new action. The name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For `aaa://host.example.com`, substitute the URL of the diameter host to which you want to redirect connections.

- add responder policy `<polname>` “diameter.req.avp(264).value.eq(”host1.example.net”)” `<actname>`

  For `<polname>`, substitute a name for your new policy. As with `<actname>`, the name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For host1.example.net, substitute the name of the originating host of the requests that you want to redirect. For `<actname>`, substitute the name of the action that you just created.

- `bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST`

  For `<vservername>`, substitute the name of the load balancing virtual server to which you want to bind the policy. For `<polname>`, substitute the name of the policy you just created. For `<priority>`, substitute a priority for the policy.

Example:

To create a Responder action and policy to respond to Diameter requests that originate from “host1.example.net” with a redirect to “host.example.com”, you could add the following action and policy, and bind the policy as shown.

```
1 > add responder action act_resp-dm-redirect RESPONDWITH "DIAMETER.
     NEW_REDIRECT(\"aaa://host.example.com\")"
2 Done
3
4 > add responder pol_resp-dm-redirect "diameter.req.avp(264).value.eq(\"
     host1.example.net\")" act_resp-dm-redirect
5 Done
6
7 > bind lb vserver vs1 -policyName pol_resp-dm-redirect -priority 10 -
     type REQUEST
8 Done
```



```
<!--NeedCopy-->
```

RADIUS Support for Responder

September 14, 2021

The Citrix ADC expressions language contains expressions that can extract information from and manipulate RADIUS requests. These expressions enable you to use the Responder feature to respond to RADIUS requests. Your responder policies and actions can use any expression that is appropriate or relevant to a RADIUS request. The available expressions enable you to identify the RADIUS message type, extract any attribute-value pair (AVP) from the connection, and send different responses on the basis of that information. You can also create policy labels that invoke all responder policies for RADIUS connections.

You can use RADIUS expressions to construct simple responses that do not require communication with the RADIUS server to which the request was sent. When a responder policy matches a connection, the Citrix ADC constructs and sends the appropriate RADIUS response without contacting the RADIUS authentication server. For example, if the source IP address of a RADIUS request is from a subnet that is specified in the responder policy, the Citrix ADC can reply to that request with an access-reject message, or can simply drop the request.

You can also create policy labels to route specific types of RADIUS requests through a series of policies that are appropriate to those requests.

Note: The current RADIUS expressions do not work with RADIUS IPv6 attributes.

The Citrix ADC documentation for expressions that support RADIUS assumes familiarity with the basic structure and purpose of RADIUS communications. If you need more information about RADIUS, see your RADIUS server documentation or search online for an introduction to the RADIUS protocol.

Configuring Responder Policies for RADIUS

The following procedure uses the Citrix ADC command line to configure a responder action and policy, and bind the policy to a RADIUS-specific global bind point.

To configure a Responder action and policy, and bind the policy:

At the command prompt, type the following commands:

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`
  where `<bindPoint>` represents one of the RADIUS-specific global bind points.

RADIUS Expressions for Responder

In a responder configuration, you can use the following Citrix ADC expressions to refer to various portions of a RADIUS request.

Identifying the Type of Connection:

- RADIUS.IS_CLIENT. Returns TRUE if the connection is a RADIUS client (request) message.
- RADIUS.IS_SERVER. Returns TRUE if the connection is a RADIUS server (response) message.

Request Expressions:

- `RADIUS.REQ.CODE`. Returns the number that corresponds to the RADIUS request type. A derivative of the num_at class. For example, a RADIUS access request would return 1 (one). A RADIUS accounting request would return 4.
- `RADIUS.REQ.LENGTH`. Returns the length of the RADIUS request, including the header. A derivative of the num_at class.
- `RADIUS.REQ.IDENTIFIER`. Returns the RADIUS request identifier, a number assigned to each request that allows the request to be matched to the corresponding response. A derivative of the num_at class.
- `RADIUS.REQ.AVP(<AVP Code No>).VALUE`. Returns the value of first occurrence of this AVP as a string of type text_t.
- `RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)`. Returns the specified instance of the AVP as a string of type RAVP_t. A specific RADIUS AVP can occur multiple times in a RADIUS message. INSTANCE (0) returns the first instance, INSTANCE (1) returns second instance, and so on, up to sixteen instances.
- `RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)`. Returns the value of specified instance of the AVP as a string of type text_t.
- `RADIUS.REQ.AVP(<AVP code no>).COUNT`. Returns the number of instances of a specific AVP in a RADIUS connection, as an integer.
- `RADIUS.REQ.AVP(<AVP code no>).EXISTS`. Returns TRUE if the specified type of AVP exists in the message, or FALSE if it does not.

Response Expressions:

RADIUS response expressions are identical to RADIUS request expressions, except that RES replaces REQ.

Typecasts of AVP Values:

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other Citrix ADC typecast expressions.

Example:

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other Citrix ADC typecast expressions.

```
RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
<!--NeedCopy-->
```

AVP Type Expressions:

The Citrix ADC supports expressions to extract RADIUS AVP values by using the assigned integer codes described in RFC2865 and RFC2866. You can also use text aliases to accomplish the same task. Some examples follow.

- RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value. Extracts the RADIUS user-name value.
- RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT_SESSION_ID.value. Extracts the Acct-Session-ID AVP (code 44) from the message.
- RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE. Extracts the vendor-specific value.

The values of most commonly-used RADIUS AVPs can be extracted in the same manner.

RADIUS Bind Points:

Four global bind points are available for policies that contain RADIUS expressions.

- RADIUS_REQ_OVERRIDE. Priority/override request policy queue.
- RADIUS_REQ_DEFAULT. Standard request policy queue.
- RADIUS_RES_OVERRIDE. Priority/override response policy queue.
- RADIUS_RES_DEFAULT. Standard response policy queue.

RADIUS Responder-Specific Expressions:

- RADIUS_RESPONDWITH. Respond with the specified RADIUS response. The response is created with Citrix ADC expressions, both RADIUS expressions and any others that are applicable.
- RADIUS.NEW_ANSWER. Sends a new RADIUS answer to the user.
- RADIUS.NEW_ACCESSREJECT. Rejects the RADIUS request.
- RADIUS.NEW_AVP. Adds the specified new AVP to the response.

Use Cases

Following are use cases for RADIUS with responder.

Blocking RADIUS Requests from a Specific Network

To configure the responder feature to block authentication requests from a specific network, begin by creating a responder action that rejects requests. Use the action in a policy that selects requests from the networks that you want to block. Bind the responder policy to a RADIUS-specific global bind point, specifying:

- The priority
- END as the nextExpr value, to ensure that policy evaluation stops when this policy is matched
- RADIUS_REQ_OVERRIDE as the queue to which you assign the policy, so that it is evaluated before policies assigned to the default queue

To configure Responder to block logons from a specific network**

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder global <polName> <priority> <nextExpr> -type <bindPoint>`

Example:

```
> add responder action rspActRadiusReject respondwith radius.
   new_accessreject
Done

> add responder policy rspPolRadiusReject client.ip.src.in_subnet
   (10.224.85.0/24) rspActRadiusReject
Done

> bind responder global rspPolRadiusReject 1 END -type
   RADIUS_REQ_OVERRIDE
<!--NeedCopy-->
```

DNS Support for the Responder Feature

September 14, 2021

You can configure the responder feature to respond to DNS requests as it does to HTTP and TCP requests. For example, you could configure it to send DNS responses over UDP and ensure that the DNS requests from the client are sent over TCP. A number of Citrix ADC expressions support examination of the DNS header in the request. These expressions examine specific header fields and send an appropriate response.

- **DNS Expressions**. In a responder configuration, you can use the following Citrix ADC expressions to refer to various portions of a DNS request:

Expressions	Descriptions
DNS.NEW_RESPONSE	Creates a new empty DNS response based on the request.
DNS.NEW_RESPONSE <AA, TC, rcode>	Creates a new DNS response based on the specified parameters.

- **DNS Bind Points**. The following global bind points are available for policies that contain DNS expressions.

Bind Points	Descriptions
DNS_REQ_OVERRIDE	Priority/override request policy queue.
DNS_REQ_DEFAULT	Standard request policy queue.

In addition to the default bind points, you can create policy labels of type DNS and bind DNS policies to them.

Configuring Responder Policies for DNS

The following procedure uses the Citrix ADC command line to configure a responder action and policy and bind the policy to a responder-specific global bind point.

To configure Responder to respond to a DNS request:

At the command prompt, type the following commands:

1. `add responder action <actName> <actType>`

   For <actname>, substitute a name for your new action. The name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actType>, substitute a responder action type, *respondWith*.

2. `add responder policy <polName> <rule> <actName>`

   For <polname>, substitute a name for your new policy. For <actname>, the name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actname>, substitute the name of the action that you just created.

3. `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`

   For <bindPoint>, specify one of the responder-specific global bind points. For <polName>, substitute the name of the policy that you just created. For <priority>, specify the priority of the policy.

Sample configuration - Enforce all DNS request over TCP:

To enforce all the DNS requests over TCP, create a responder action that will set the TC bit and rcode as NOERROR.

```
> add responder action resp_act_set_tc_bit respondwith DNS.NEW_RESPONSE
   (true, true, NOERROR)
Done

> add responder policy enforce_tcp  dns.REQ.TRANSPORT.EQ(udp)
   resp_act_set_tc_bit
Done

>bind lb vserver dns_udp  -policyName  enforce_tcp   -type request  -
   priority 100
Done
<!--NeedCopy-->
```

MQTT support for responder

September 14, 2021

The Responder feature supports the MQTT protocol. You can configure responder policies to take an action based on the parameters in the incoming MQTT message.

The action responds with any of the following to a new connection:

- DROP
- RESET
- NOOP
- A responder action to initiate a new MQTT CONNACK response.

Configuring responder policies for MQTT

After enabling the responder feature, you must configure one or more actions for handling MQTT requests. Then, configure a responder policy. You can bind the responder policies globally, or to a specific load balancing virtual server or content switching virtual server.

The following bind points are available to bind the responder policies globally:

- MQTT_REQ_DEFAULT
- MQTT_REQ_OVERRIDE
- MQTT_JUMBO_REQ_DEFAULT

- MQTT_JUMBO_REQ_OVERRIDE

The following bind points are available to bind the responder policies to a content switching or load balancing virtual server:

- REQUEST
- MQTT_JUMBO_REQ (this bind point is used only for Jumbo packets)

To configure the responder to respond to an MQTT request by using CLI

At the command prompt, type the following commands:

Configure a responder action.

```
add responder action <actName> <actType>
<!--NeedCopy-->
```

- For `actname`, substitute a name for your new action. The name can be 1–127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols.
- For `actType`, substitute a responder action type, respondwith.

Example:

```
add responder action mqtt_connack_unsup_ver respondwith MQTT.
   NEW_CONNACK(132)
<!--NeedCopy-->
```

Configure a responder policy. The Citrix ADC appliance responds to the MQTT requests that are selected by this responder policy.

```
add responder policy <polName> <rule> <actname>
<!--NeedCopy-->
```

- For `polname`, substitute a name for your new policy.
- For `actname`, substitute the name of the action that you created.

Example:

```
add responder policy reject_lower_version "MQTT.HEADER.COMMAND.EQ(
   CONNECT) && MQTT.VERSION.LT(3)" mqtt_connack_unsup_ver
<!--NeedCopy-->
```

Bind the responder policy to a specific load balancing virtual server or content switching virtual server. The policy applies only to the MQTT requests whose destination IP address is the VIP of that virtual server.


```
bind lb vserver <name> -policyName <policy_name> -priority <priority>

bind cs vserver <name> -policyName <policy_name> -priority <priority>
<!--NeedCopy-->
```

- For `policy_name`, substitute the name of the policy that you have created.
- For `priority`, specify the priority of the policy.

Example:

```
bind lb vserver lb1 -policyName reject_lower_version -priority 50

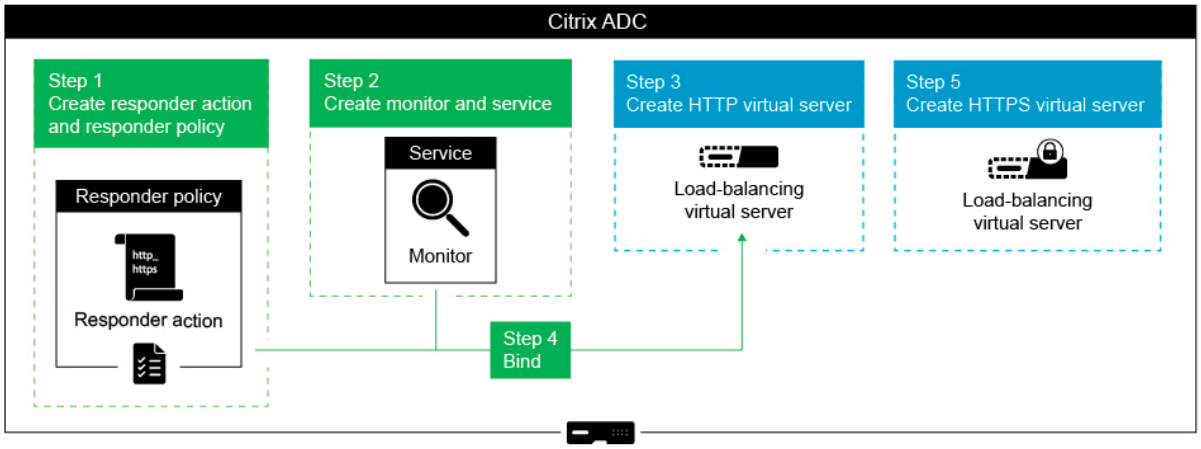
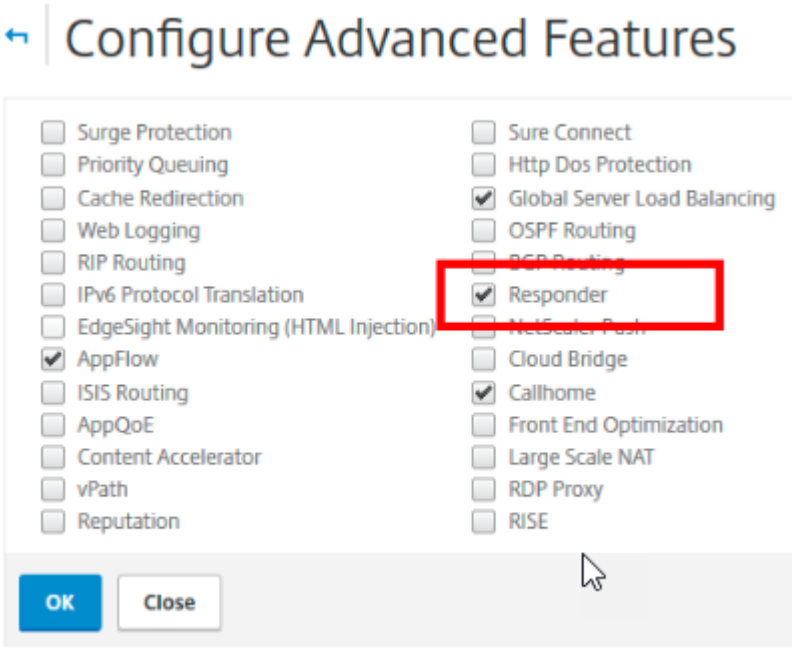
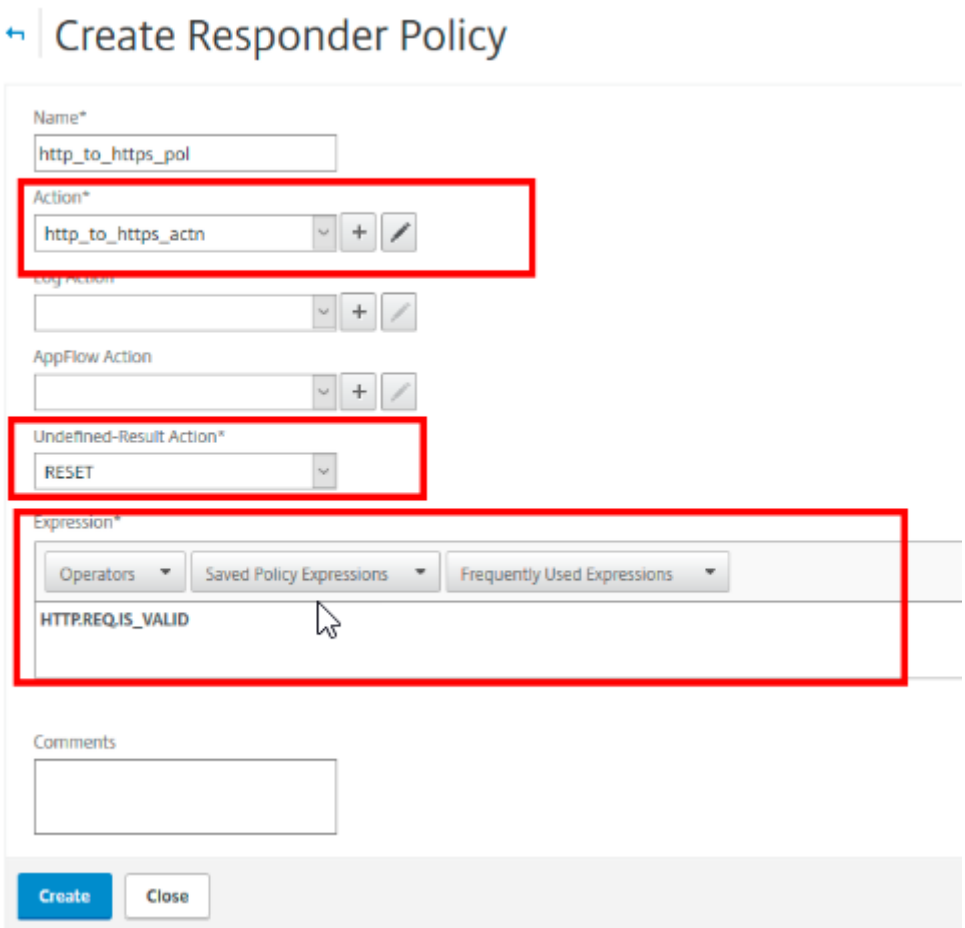
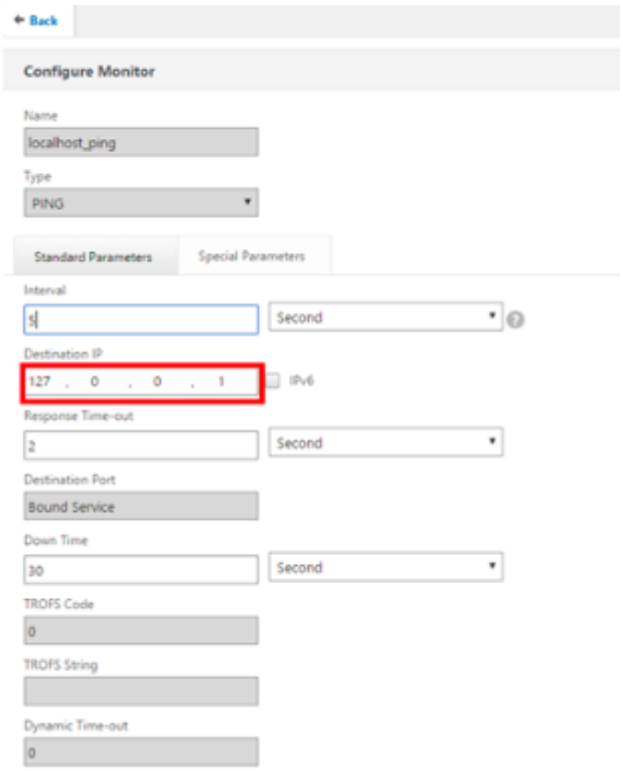
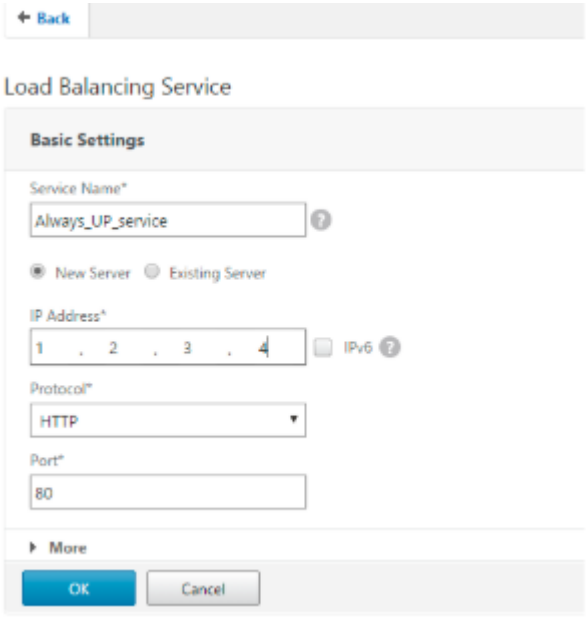
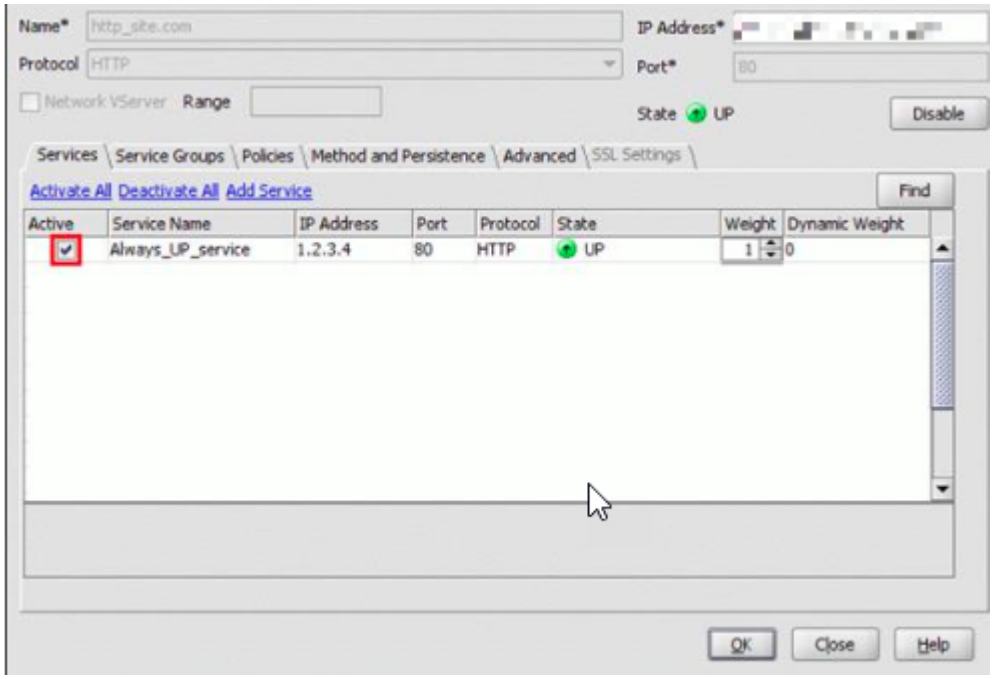
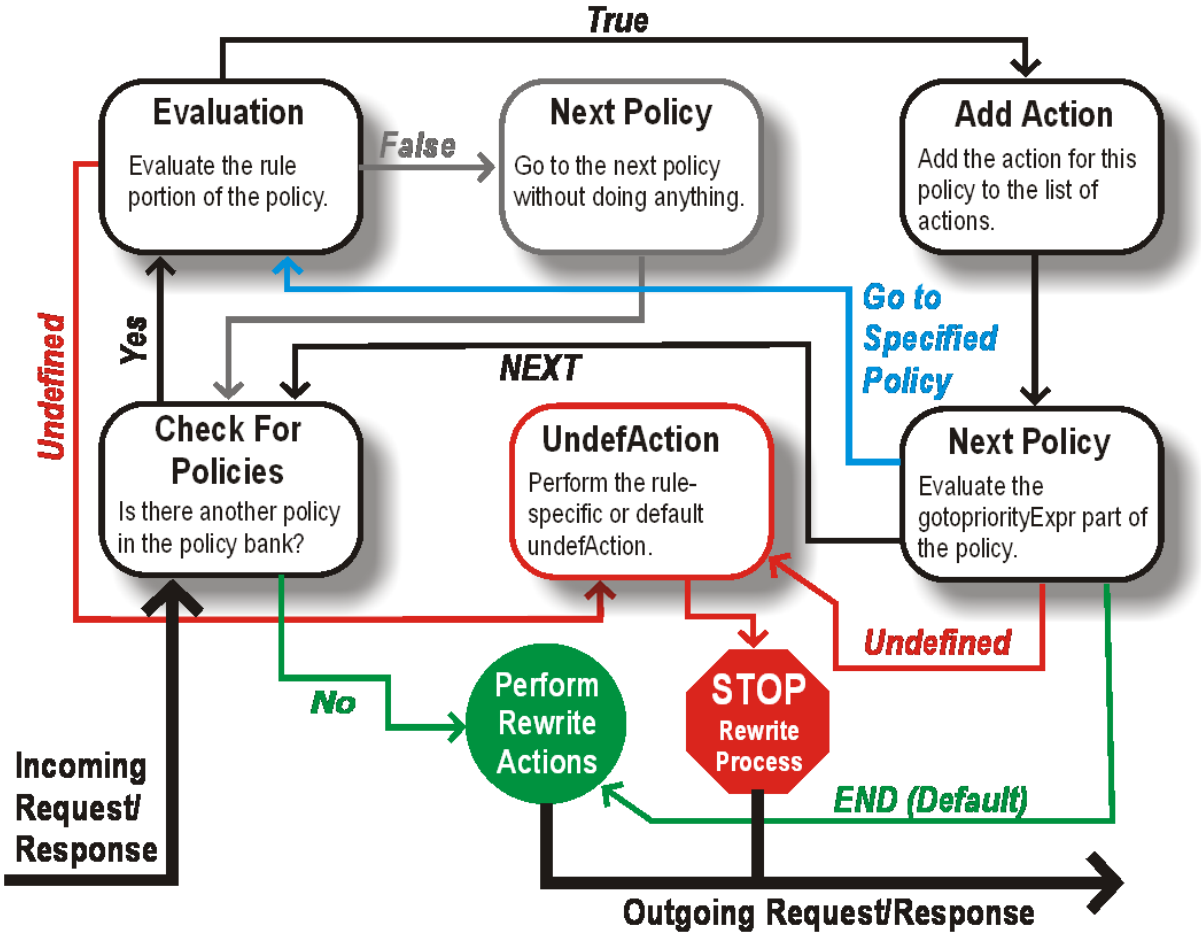
bind cs vserver mqtt_frontend_cs -policyName reject_lower_version -
   priority 5
<!--NeedCopy-->
```

Use case1: Filter clients based on the user name or client ID

The administrator can configure an MQTT responder policy to reject the connection based on the user name or client ID in the MQTT CONNECT message.

Sample configuration for filtering clients based on the client ID

```
add policy patset filter_clients
bind policy patset filter_clients client1

add responder action mqtt_connack_invalid_client respondwith MQTT.
   NEW_CONNACK(2)

add responder policy reject_clients "MQTT.HEADER.COMMAND.EQ(CONNECT) &&
    mqtt.connect.clientid.equals_any(\"filter_clients\")"
   mqtt_connack_invalid_client

bind cs vserver mqtt_frontend_cs -policyName reject_clients -priority 5
<!--NeedCopy-->
```

Use case2: Limit the maximum message length of MQTT messages to handle jumbo packets

The administrator can configure an MQTT responder policy to drop the client connection if the length of the message exceeds a certain threshold, or take necessary action based on the requirement.

To handle jumbo packets, the responder policies with any of the following rule patterns are bound to the jumbo bind point:

- MQTT.MESSAGE_LENGTH
- MQTT.COMMAND
- MQTT.FROM_CLIENT
- MQTT.FROM_SERVER

Policies bound to jumbo bind points are evaluated only for jumbo packets.

Sample configuration for limiting the maximum message length of MQTT messages

```
set lb parameter -dropmqttjumbomessage no

add responder policy drop_large_message MQTT.MESSAGE_LENGTH.GT(100000)
    reset

bind cs vserver mqtt_frontend_cs -policyName drop_large_message -
    priority 10
<!--NeedCopy-->
```

In this example, the `dropmqttjumbomessage` parameter is set to NO. Therefore, the ADC appliance processes the messages with length greater than 64,000 bytes and less than 1,00,000 bytes. The messages with length greater than 1,00,000 bytes are reset.

How to redirect HTTP request to HTTPS using responder

September 14, 2021

This article explains how to configure the responder feature with a load balancing virtual server IP addresses and redirect client requests from HTTP to HTTPS.

Consider a scenario, where a user might attempt to access a secure web site by sending an HTTP request. Instead of dropping the request, you might want to redirect the request to a secure web site. You can use the responder feature to redirect the request to the secure web site without changing the path and the URL query which the user attempts to access.

How Citrix ADC responder redirects a request from HTTP to HTTPS

The following illustration shows a step by step flow of how the appliance redirects a request.



> **Note**: The navigation paths and screen shots are derived from NetScaler 11.0.

To configure the Responder feature along with the Load Balancing VIP addresses of a NetScaler appliance to redirect client requests from HTTP to HTTPS, complete the following procedure.

1. Enable the responder feature on the appliance. Navigate to **System** > **Settings** > **Configure Advanced Features** > **Responder**.



2. Create a responder action and specify an appropriate name, such as, http_to_https_actn, in the Name field.
3. To create a responder action, in the navigation pane, expand **AppExpert** > **Responder**, click **Actions** and then click **Add**.
4. Select Redirect as Type.

5. In the **Expression** field, type the following expression:

   ```
   "https://"+ HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + HTTP.REQ.URL.PATH_AND_QUERY
   .HTTP_URL_SAFE.
   ```

6. In NetScaler version 9.0 and 10.0 ensure that the **Bypass Safety Check** option is cleared.

   Note: This option is not present from NetScaler 11.0 onwards.

7. Create **Responder Policy** and specify an appropriate name, such as http_to_https_pol, in the Name field.
8. To create a Responder Policy, in the navigation pane, expand **AppExpert** > **Responder**, click **Policies** and then click **Add**.
9. From the Action list, select the action name that you have created.
10. From the Undefined Action list, select RESET.
11. Type the **HTTP.REQ.IS_VALID** expression in the **Expression** field as shown in the following screen shot.



1. Create a monitor for which the status is always marked as UP and specify an appropriate name, such as localhost_ping, in the Name field.

2. To create a monitor, in the navigation pane expand **Load Balancing**, click **Monitors** and then click **Add**.
3. In the **Destination IP** field, specify the 127.0.0.1 IP address, as shown in the following screen shot.



4. Create a service and specify an appropriate name, such as Always_UP_service, in the **Name** field.
5. To create a service, in the navigation pane, expand **Load Balancing**, click **Services** and then click **Add**.
6. Specify a non-existent IP address in the **Server** field.



7. Specify 80 in the **Port** field.
8. Add the created monitor from the **Available Monitors** list.
9. Create a Load Balancing Virtual Server and specify an appropriate name in the **Name** field.
10. To create a Load Balancing Virtual Server, in the navigation pane, expand **Load Balancing**, click **Services** and then click **Add**.
11. Specify the IP address of the web site in IP Address field.
12. Select HTTP from the Protocol list.
13. Type 80 in the Port field.
14. On NetScaler version 9.0 and 10.0, select the Active option for the service you have created in the Services tab as shown in the following screen shot. This option is deprecated in NetScaler version 11.0.



15. Click the **Policies** tab.
16. Bind the Responder policy you created to the HTTP Load Balancing VIP address of the web site.
17. Create a secure Load Balancing virtual server that has the IP address of the web site and port as 443.

To create a configuration similar to the preceding procedure from the command line interface of the appliance, run following commands:

```
enable ns feature responder
add responder action http_to_https_actn redirect "\"https://\" + http.
    req.hostname.HTTP_URL_SAFE + http.REQ.URL.PATH_AND_QUERY.
    HTTP_URL_SAFE"
add responder policy http_to_https_pol HTTP.REQ.IS_VALID
    http_to_https_actn RESET
add lb monitor localhost_ping PING -LRTM ENABLED -destIP 127.0.0.1
add service Always_UP_service 1.2.3.4 HTTP 80 -gslb NONE -maxClient 0 -
    maxReq 0 -cip ENABLED dummy -usip NO -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP YES
bind lb monitor localhost_ping Always_UP_service
add lb vserver http_site.com HTTP 10.217.96.238 80 -persistenceType
    COOKIEINSERT -timeout 0 -cltTimeout 180
bind lb vserver http_site.com Always_UP_service
bind lb vserver http_site.com -policyName http_to_https_pol -priority 1
     -gotoPriorityExpression END
<!--NeedCopy-->
```


> **Notes**:
>
> - The status of the port 80 Load Balancing Redirect virtual server must be UP for the redirect to work.
> - Web browsers might not redirect correctly if the HTTPS virtual server is not active.
> - This redirect setup allows for situations where multiple domains are bound to the same IP address.
> - If the client sends an invalid HTTP request to the redirect virtual server, then the appliance sends a RESET message code.

Troubleshooting

September 14, 2021

If the responder feature does not work as expected after you have configured it, you can use some common tools to access Citrix ADC resources and diagnose the problem.

Resources for Troubleshooting

For best results, use the following resources to troubleshoot an integrated cache issue on a Citrix ADC appliance:

- The ns.conf file
- The relevant trace files from the client and the Citrix ADC appliance

In addition to the above resources, the following tools expedite troubleshooting:

- The iehttpheaders or a similar utility
- The Wireshark application customized for the Citrix ADC trace files

Troubleshooting Responder Issues

- **Issue**

  The Responder feature is configured, but the responder action is not working.

  Resolution

  - Verify that the feature is enabled.
  - Check the hit counters of any of the policies to see if the counters are getting incremented.
  - Verify that the policies and actions are configured correctly.
  - Verify that the actions and policies are bound appropriately.

  - Record the packet traces on the client and the Citrix ADC appliance, and analyze the them to get some pointer to the issue.
  - Record the iehttpHeaters packet traces on the client and verify the HTTP requests and responses to get some pointer to the issue.

- **Issue**

  You need to create a maintenance page.

  Resolution

1. Configure the services and virtual Server.
2. Configure a backup virtual server with a service bound to it. This ensures that the status of the Web site is always displayed as UP.
3. Configure the primary virtual server to use the backup virtual server as a backup.
4. Create a responder action with an appropriate target. Following is an example for your reference:

   ```
   add responder action sorry_page respondwith q{ "HTTP/1.0 200 OK"+"\r\n\
   r\n"+ "<html><body>Sorry, this page is not available\</body>\</html>"+
   "\r\n"}
   ```

5. Create a responder policy and bind the action to it.
6. Bind the responder policy to the backup virtual Server.

Rewrite

September 14, 2021

> **Warning**
>
> Filter features using classic policies are deprecated and as an alternative Citrix recommends you to use the rewrite and responder features with advanced policy infrastructure.

Rewrite refers to the rewriting of some information in the requests or responses handled by the Citrix ADC appliance. Rewriting can help in providing access to the requested content without exposing unnecessary details about the Web site's actual configuration. A few situations in which the rewrite feature is useful are described below:

- To improve security, the Citrix ADC can rewrite all the `http://links` to `https://` in the response body.

- In the SSL offload deployment, the insecure links in the response have to be converted into secure links. Using the rewrite option, you can rewrite all the `http://links` to `https://` for making sure that the outgoing responses from Citrix ADC to the client have the secured links.
- If a Web site has to show an error page, you can show a custom error page instead of the default 404 Error page. For example, if you show the home page or site map of the Web site instead of an error page, the visitor remains on the site instead of moving away from the Web site.
- If you want to launch a new Web site, but use the old URL, you can use the Rewrite option.
- When a topic in a site has a complicated URL, you can rewrite it with a simple, easy-to-remember URL (also referred to as 'cool URL').
- You can append the default page name to the URL of a Web site. For example, if the default page of a company's Web site is `http://www.abc.com/index.php`, when the user types 'abc.com' in the address bar of the browser, you can rewrite the URL to 'abc.com/index.php'.

When you enable the rewrite feature, Citrix ADC can modify the headers and body of HTTP requests and responses.

To rewrite HTTP requests and responses, you can use protocol-aware Citrix ADC policy expressions in the rewrite policies you configure. The virtual servers that manage the HTTP requests and responses must be of type
HTTP or
SSL. In HTTP traffic, you can take the following actions:

- Modify the URL of a request
- Add, modify or delete headers
- Add, replace, or delete any specific string within the body or headers.

To rewrite TCP payloads, consider the payload as a raw stream of bytes. Each of the virtual servers that managing the TCP connections must be of type TCP or SSL_TCP. The term TCP rewrite is used to refer to the rewrite of TCP payloads that are not HTTP data. In TCP traffic, you can add, modify, or delete any part of the TCP payload.

For examples to use the rewrite feature, see Rewrite Action and Policy Examples.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a http request to new Web sites or Web pages
- Responding with some custom response
- Dropping or resetting a connection at request level

In case of a responder policy, the Citrix ADC examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

In case of a rewrite policy, the Citrix ADC examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the Citrix ADC to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

How rewrite works

September 14, 2021

A rewrite policy consists of a rule and action. The rule determines the traffic on which rewrite is applied and the action determines the action to be taken by the Citrix ADC. You can define multiple rewrite policies. For each policy, specify the bind point and priority.

A bind point refers to a point in the traffic flow at which the Citrix ADC examines the traffic to verify whether any rewrite policy can be applied to it. You can bind a policy to a specific load balancing or content switching virtual server, or make the policy global if you want the policy to be applied to the entire traffic handled by the Citrix ADC. These policies are referred to as global policies.

In addition to the user-defined policies, the Citrix ADC has some default policies. You cannot modify or delete a default policy.

For evaluating the policies, Citrix ADC follows the order mentioned below:

- Global policies
- Policies bound to specific virtual servers
- Default policies

Note: Citrix ADC can apply a rewrite policy only when it is bound to a point.

Citrix ADC implements the rewrite feature in the following steps:

- The Citrix ADC appliance checks for global policies and then checks for policies at individual bind points.
- If multiple policies are bound to a bind point, the Citrix ADC evaluates the policies in the order of their priority. The policy with the highest priority is evaluated first. After evaluating each policy, if the policy is evaluated to TRUE (the traffic matches the rule), it adds the action associated with the policy to a list of actions to be performed. A match occurs when the characteristics specified in the policy rule match the characteristics of the request or response being evaluated.

- For any policy, in addition to the action, you can specify the policy that should be evaluated after the current policy is evaluated. This policy is referred to as the 'Go to Expression'. For any policy, if a Go to Expression (gotoPriorityExpr) is specified, the Citrix ADC evaluates the Go to Expression policy; it ignores policy with the next highest priority.

  You can specify the priority of the policy to indicate the Go to Expression policy; you cannot use the name of the policy. If you want the Citrix ADC to stop evaluating other policies after evaluating a particular policy, you can set the Go to Expression to 'END'.

- After all the policies are evaluated or when a policy has the Go to Expression set as END, the Citrix ADC starts performing the actions according to the list of actions.

For more information about configuring rewrite policies, see Configuring a Rewrite Policy and about binding rewrite policies, see Binding a Rewrite Policy.

The following figure illustrates how Citrix ADC processes a request or response when the rewrite feature is used.

Figure 1. The Rewrite Process



Policy Evaluation

The policy with the highest priority is evaluated first. Citrix ADC does not stop the evaluation of rewrite policies when it finds a match; it evaluates all the rewrite policies configured on the Citrix ADC.

- If a policy evaluates to TRUE, the Citrix ADC follows the procedure below:
  - If the policy has the Go to Expression set to END, the Citrix ADC stops evaluating all the other policies and starts performing the rewrite.
  - The gotoPriorityExpression can be set to 'NEXT', 'END', some integer or 'INVOCATION_LIST'. The value determines the policy with the next priority. The following table shows the action taken by Citrix ADC for each value of the expression.

Value of the expression	Action
NEXT	Policy with the next priority gets evaluated.
END	Evaluation of policies stops.
<an integer>	Policy with specified priority gets evaluated.
INVOCATION_LIST	Goto NEXT or END is applied based on the result of the invocation list.

- If a policy evaluates to FALSE, the Citrix ADC continues the evaluation in the order of priority.
- If a policy evaluates to UNDEFINED (cannot be evaluated on the received traffic due to an error), the Citrix ADC performs the action assigned to the UNDEFINED condition (referred to as undefAction) and stops further evaluation of polices.

The Citrix ADC starts the actual rewriting only after the evaluation is complete. It refers to the list of actions identified by policies that are evaluated to TRUE, and starts the rewriting. After implementing all the actions in the list, the Citrix ADC forwards the traffic as required.

> **Note**:
>
> Ensure that the policies do not specify conflicting or overlapping actions on the same part of the HTTP header or body, or TCP payload. When such a conflict occurs, the Citrix ADC encounters an undefined situation and aborts the rewrite.

Rewrite Actions

On the Citrix ADC appliance, specify the actions to be taken such as adding, replacing, or deleting text within the body, or adding, modifying or deleting headers, or any changes in the TCP payload as rewrite actions. For more information about rewrite actions, see Configuring a Rewrite Action.

The following table describes the steps the Citrix ADC can take when a policy evaluates to TRUE.

Action	Result
Insert	The rewrite action specified for the policy is carried out.
NOREWRITE	The request or response is not rewritten. Citrix ADC forwards the traffic without rewriting any part of the message.
RESET	The connection is aborted at the TCP level.
DROP	The message is dropped.

> **Note**:
>
> For any policy, you can configure the undefaction (action to be taken when the policy evaluates to UNDEFINED) as NOREWRITE, RESET, or DROP.

To use the
Rewrite feature, take the following steps:

- Enable the feature on the Citrix ADC.
- Define rewrite actions.
- Define rewrite policies.
- Bind the policies to a bind point to bring a policy into effect.

Enabling the rewrite feature

September 14, 2021

Enable the rewrite feature on the Citrix ADC appliance if you want to rewrite the HTTP or TCP requests or responses. If the feature is enabled, Citrix ADC takes rewrite action according to the specified policies. For more information, see How rewrite works.

To enable the rewrite feature by using the command line interface

At the command prompt, type the following commands to enable the rewrite feature and verify the configuration:

- enable ns feature REWRITE
- show ns feature

Example:


```
> enable ns feature REWRITE
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 .
 .
 .
 1)     Rewrite                         REWRITE             ON
 .
 .
 1)     Citrix ADC Push                  push                OFF
 Done
<!--NeedCopy-->
```

To enable the rewrite feature by using the configuration utility

1. In the navigation pane, click **System**, and then click **Settings**.
2. In the details pane, under Modes and Features, click **Configure basic features**.
3. In the **Configure Basic Features** dialog box, select the Rewrite check box, and then click **OK**.
4. In the **Enable/Disable Feature(s)** dialog box, click **Yes**. A message appears in the status bar, stating that the selected feature was enabled.

Configure a Rewrite Action

September 17, 2021

> **Warning**
>
> The Pattern function in a rewrite action is deprecated from Citrix ADC 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use the Search rewrite action parameter.

A rewrite action indicates changes made to a request or response prior to sending it to a server or client.

Expressions define the following:

- Rewrite action type.
- Location of the rewrite action.

- Rewrite action configuration type.

For Example, a DELETE action only uses a target expression. A REPLACE action uses a target expression and an expression to configure the replacement text.

After enabling the rewrite feature, you need to configure one or more actions unless a built-in rewrite action is sufficient. All of the built-in actions have names beginning with the string ns_cvpn, followed by a string of letters and underscore characters. Built-in actions perform useful and complex tasks such as decoding parts of a clientless VPN request or response or modifying JavaScript or XML data. The built-in actions can be viewed, enabled, and disabled, but cannot be modified or deleted.

> Note:
>
> Action types that can be used only for HTTP rewrite are identified in the **Rewrite Action Type** column.

For more information, see **Type parameter**.

Create a rewrite action by using the command line interface

At the command prompt, type the following commands to create a rewrite action and verify the configuration:

- `add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-search <expression>] [refineSearch <expression>] [-comment<string>]`
- `show rewrite action <name>`

For more information, see the Rewrite Action Types and their Arguments table.

The rewrite feature has the following built-in actions:

- NOREWRITE-Sends the request or response to the user without rewriting it.
- RESET - Resets the connection and notifies the user's browser, so that the user can resend the request.
- DROP - Drops the connection without sending a response to the user.

One of the following flow types is implicitly associated with every action:

- Request - Action applies to the request.
- Response - Action applies to the response.
- Neutral - Action applies to both requests and responses.

Name

Name for the user-defined rewrite action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@),

equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added.

Type parameter

The **Type** parameter shows the type of user-defined rewrite action.

Following are the values of the **Type** parameter:

- `REPLACE <target> <string_builder_expr>`. Replaces the string with the string-builder expression.

Example:

```
> add rewrite action replace_http_act replace http.res.body(100) '"
   new_replaced_data"'
Done
> sh rewrite action replace_http_act
Name: replace_http_act
Operation: replace
Target:http.res.body(100)
Value:"new_replaced_data"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- `REPLACE_ALL <target> <string_builder_expr1> -(pattern|search)<string_builder_expr>`. In the request or response specified by `<target>`, replaces all occurrences of the string defined by `<string_builder_expr1>` with the string defined by `<string_builder_expr2>`. You can use a PCRE-format pattern or the search facility to find the strings to be replaced.

Example:

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY
   (100000)" "\"https://\""-search "patset(\"pat_list_2\")" -
   refineSearch "EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done

```



```
> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
Search: patset("pat_list_2")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_HTTP_RES <string_builder_expr>. Replaces the complete HTTP response with the string defined by the string-builder expression.

Example:

```
> add rewrite action replace_http_res_act replace_http_res '"HTTP/1.1
   200 OK\r\n\r\nSending from ADC"'
 Done
> sh rewrite action replace_http_res_act
Name: replace_http_res_act
Operation: replace_http_res
Target:"HTTP/1.1 200 OK
Sending from ADC"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_SIP_RES <target>. Replaces the complete SIP response with the string specified by <target>.

Example:

```
> add rewrite action replace_sip_res_act replace_sip_res '"HTTP/1.1 200
    OK\r\n\r\nSending from ADC"'
Done
> sh rewrite action replace_sip_res_act
Name: replace_sip_res_act
Operation: replace_sip_res
Target:"HTTP/1.1 200 OK
```



```
7  Sending from ADC"
8  Hits: 0
9  Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- INSERT_HTTP_HEADER <header_string_builder_expr> <contents_string_builder_expr>. Inserts the HTTP header specified by and header contents specified by .

Example:

```
1  > add rewrite action ins_cip_header insert_http_header "CIP" "CLIENT.IP
       .SRC"
2  Done
3  > sh rewrite action ins_cip_header
4  Name: ins_cip_header
5  Operation: insert_http_header
6  Target:CIP
7  Value:CLIENT.IP.SRC
8  Hits: 0
9  Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- DELETE_HTTP_HEADER <target>. Deletes the HTTP header specified by <target>

Example:

```
1  > add rewrite action del_true_client_ip_header delete_http_header "True
       -Client-IP"
2  Done
3  > sh rewrite action del_true_client_ip_header
4  Name: del_true_client_ip_header
5  Operation: delete_http_header
6  Target:True-Client-IP
7  Hits: 0
8  Undef Hits: 0
9  Action Reference Count: 0
10 Done
11
12 <!--NeedCopy-->
```


- CORRUPT_HTTP_HEADER <target>. Replaces the header name of all occurrences of the HTTP header specified by <target> with a corrupted name, so that it will not be recognized by the receiver Example: MY_HEADER is changed to MHEY_ADER.

Example:

```
> add rewrite action corrupt_content_length_hdr corrupt_http_header "
   Content-Length"
Done
> sh rewrite action corrupt_content_length_hdr
Name: corrupt_content_length_hdr
Operation: corrupt_http_header
Target:Content-Length
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE <string_builder_expr1> <string_builder_expr1>. Finds the string specified in <string_builder_expr1> and inserts the string in <string_builder_expr2> before it.

```
> add rewrite action insert_before_ex_act insert_before http.res.body
   (100) '"Add this string in the starting"'
Done
> sh rewrite action insert_before_ex_act
Name: insert_before_ex_act
Operation: insert_before
Target:http.res.body(100)
Value:"Add this string in the starting"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE_ALL <target> <string_builder_expr1> -(pattern|search)<string_builder_expr2>. In the request or response specified by <target>, locates all occurrences of the string specified in <string_builder_expr1> and inserts the string specified in <string_builder_expr2> before it. You can use a PCRE-format pattern or the search facility to find the strings.

Example:

```
> add policy patset pat
 Done
> bind policy patset pat abcd
 Done
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
   (10) 'target.prefix(10) + "refineSearch_testing"' -search patset("
   pat") -refineSearch extend(10,10)
 Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
Refine Search:extend(10,10)
Value:target.prefix(10) + "refineSearch_testing"
Search: patset("pat")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- `INSERT_AFTER <string_builder_expr1> <string_builder_expr2>`. Finds the string specified in , and inserts the string specified in after it.
  Example:

```
> add rewrite action insert_after_act insert_after http.req.body(100) '
   "add this string after 100 bytes"'
Done
> sh rewrite action insert_after_act
Name: insert_after_act
Operation: insert_after
Target:http.req.body(100)
Value:"add this string after 100 bytes"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- `INSERT_AFTER_ALL <target> <string_builder_expr1> -(pattern|search)< string_builder_expr>`. In the request or response specified by `<target>`, locates all

occurrences of the string specified by <string_builder_expr1> and inserts the string specified by <string_builder_expr2> after each. You can use a PCRE-format pattern or the search facility to find the strings.

Example:

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
   (100) '"refineSearch_testing"' -search text("abc") -refineSearch
   extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE <target>. Finds and deletes the specified target.

Example:

```
> add rewrite action delete_ex_act delete http.req.header("HDR")
Done
> sh rewrite action delete_ex_act
Name: delete_ex_act
Operation: delete
Target:http.req.header("HDR")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_ALL <target> -(pattern|search)<string_builder_expr>. In the request or response specified by <target>, locates and deletes all occurrences of the string specified by <string_builder_expr>. You can use a PCRE-format pattern or the search facility to find the strings.

Example:

```
>add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
    " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
    REGEX_SELECT(re#\\s`*`<AppData>.`*`\\s`*`<\\/AppData>#)"
Done
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
Operation: delete_all
Target:HTTP.RES.BODY(50000)
Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s`*`<AppData>.`*`\s`*`<\/
    AppData>#)
Search: text("Windows Desktops")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DIAMETER_HEADER_FIELD <target> <field value>. In the request or responses modify the header field specified by <target>. Use Diameter.req.flags.SET (<flag>) or Diameter.req.flags.UNSET<flag> as stringbuilderexpression to set or unset flags.

Example:

```
> add rewrite action replace_diameter_field_ex_act
    replace_diameter_header_field diameter.req.flags diameter.req.flags.
    set(PROXIABLE)
Done
> sh rewrite action replace_diameter_field_ex_act
Name: replace_diameter_field_ex_act
Operation: replace_diameter_header_field
Target:diameter.req.flags
Value:diameter.req.flags.set(PROXIABLE)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DNS_HEADER_FIELD <target>. In the request or response modifies the header field specified by <target>.

Example:

```
> add rewrite action replace_dns_hdr_act replace_dns_header_field dns.
   req.header.flags.set(AA)
Done
> sh rewrite action replace_dns_hdr_act
Name: replace_dns_hdr_act
Operation: replace_dns_header_field
Target:dns.req.header.flags.set(AA)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_DNS_ANSWER_SECTION <target>. Replace the DNS answer section in the response. This is currently applicable for A and AAAA records only. Use DNS.NEW_RRSET_A and NS.NEW_RRSET_AAAA expressions to configure the new answer section.

Example:

```
> add rewrite action replace_dns_ans_act replace_dns_answer_section
   DNS.NEW_RRSET_A("1.1.1.1", 10)
Done
> sh rewrite action replace_dns_ans_act
Name: replace_dns_ans_act
Operation: replace_dns_answer_section
Target:DNS.NEW_RRSET_A("1.1.1.1", 10)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE<target>. Decodes the pattern specified by target In clientless VPN format.

Example:

```
> add rewrite action cvpn_decode_act_1 clientless_vpn_decode http.req.
   body(100)
Done
> sh rewrite action cvpn_decode_act_1
Name: cvpn_decode_act_1
```



```
Operation: clientless_vpn_decode
Target:http.req.body(100)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE_ALL<target>-search<expression>. Decodes ALL the patterns specified by search parameter In clientless VPN format.

Example:

```
> add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
   -search text("abcd")
Done
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE<target>. Encodes the pattern specified by target in clientless VPN format.

Example:

```
> add rewrite action cvpn_encode_act_1 clientless_vpn_encode http.req.
   body(100)
Done
> sh rewrite action cvpn_encode_act_1
Name: cvpn_encode_act_1
Operation: clientless_vpn_encode
Target:http.req.body(100)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

```



```
12 <!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE_ALL<target>-search<expression>. Encodes ALL the patterns specified search parameter in clientless VPN format.

Example:

```
1 > add rewrite action act2 clientless_vpn_encode_all http.req.body(100)
     -search text("abcd")
2 Done
3 > sh rewrite action act2
4 Name: act1
5 Operation: clientless_vpn_encode_all
6 Target:http.req.body(100)
7 Search: text("abcd")
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- CORRUPT_SIP_HEADER<target>. Replaces the header name of all occurrences of the SIP header specified by <target> with a corrupted name, so that the receiver doesn't recognize it.

Example:

```
1 > add rewrite action corrupt_sip_hdr_act corrupt_sip_header SIP_HDR
2 Done
3 > sh rewrite action corrupt_sip_hdr_act
4 Name: corrupt_sip_hdr_act
5 Operation: corrupt_sip_header
6 Target:SIP_HDR
7 Hits: 0
8 Undef Hits: 0
9 Action Reference Count: 0
10 Done
11
12 <!--NeedCopy-->
```

- INSERT_SIP_HEADER <header_string_builder_expr> <contents_string_builder_expr>. Inserts the SIP header specified by <header_string_builder_expr> and header contents specified by <contents_string_builder_expr>.

Example:


```
> add rewrite action insert_sip_hdr_act insert_sip_header SIP_HDR '"
    inserting_sip_header"'
 Done
>sh rewrite action insert_sip_hdr_act
Name: insert_sip_hdr_act
Operation: insert_sip_header
Target:SIP_HDR
Value:"inserting_sip_header"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_SIP_HEADER<target>. Deletes the SIP header specified by <target>

Example:

```
> add rewrite action delete_sip_hdr delete_sip_header  SIP_HDR
Done
> sh rewrite action delete_sip_hdr
Name: delete_sip_hdr
Operation: delete_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

Target parameter

The Target parameter Is an expression that specifies which part of the request or response to rewrite.

StringBuilderExpr

The StringBuilderExpr Is an expression that specifies the content that Is to be Inserted into the request or response at the specified location. This expression replaces a specified string.

Example 1. Inserting an HTTP Header With the Client IP:


```
> add rewrite action insertact INSERT_HTTP_HEADER "client-IP" CLIENT.IP
   .SRC
Done
> show rewrite action insertact
Name: insertact
Operation: insert_http_header
Target:Client-IP
Value:CLIENT.IP.SRC
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

Example 2. Replacing Strings in a TCP Payload (TCP Rewrite):

```
> add rewrite action client_tcp_payload_replace_all REPLACE_ALL
  'client.tcp.payload(1000)' '"new-string"' -search text("old-string")
Done
> show rewrite action client_tcp_payload_replace_all
Name: client_tcp_payload_replace_all
Operation: replace_all
Target:client.tcp.payload(1000)
Value:"new-string"
Search: text("old-string")
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

Search a part of the request or response to rewrite

The Search functionality helps to find all the Instances of the required pattern in the request or response.

The Search functionality is required to be used in the following Action types:

- INSERT_BEFORE_ALL
- INSERT_AFTER_ALL

- REPLACE_ALL
- DELETE_ALL
- CLIENTLESS_VPN_ENCODE_ALL
- CLIENTLESS_VPN_DECODE_ALL

The Search functionality cannot be used with the following Action types:

- INSERT_HTTP_HEADER
- INSERT_BEFORE
- INSERT_AFTER
- REPLACE
- DELETE
- DELETE_HTTP_HEADER
- CORRUPT_HTTP_HEADER
- REPLACE_HTTP_RES
- CLIENTLESS_VPN_ENCODE
- CLIENTLESS_VPN_DECODE
- INSERT_SIP_HEADER
- DELETE_SIP_HEADER
- CORRUPT_SIP_HEADER
- REPLACE_DIAMETER_HEADER_FIELD
- REPLACE_DNS_ANSWER_SECTION
- REPLACE_DNS_HEADER_FIELD
- REPLACE_SIP_RES

The following Search types are supported:

- Text - a literal string
  Example: -search text (”hello”)
- Regular Expression - pattern that is used to match multiple strings in the request or response
  Example: -search regex(re~^hello*~)
- XPATH - An XPATH expression to search XML.
  Example: -search xpath(xp%/a/b%)
- JSON - An XPATH expression to search JSON.
  Example: -search xpath_json(xp%/a/b%)
  HTML - An XPATH expression to search HTML
  Example: -search xpath_html(xp%/html/body%)
  Patset - This searches all the patterns bound to the patset entity.
  Example: -search patset(“patset1”)
- Datset - This searches all the patterns bound to the datset entity.
  Example: -search dataset(“dataset1”)
- AVP - AVP number that is used to match multiple AVPs in a Diameter/Radius Message

Example: -search avp(999)

Refine the search results

You can use the Refine Search functionality to specify the additional criteria to refine the search results. Refine Search functionality can only be used if Search functionality is used.
The Refine search parameter always starts with the “extend(m,n)” operation, where ‘m’ specifies a number of bytes to the left of the search result and ‘n’ specifies a number of bytes to the right of the search result to extend the selected area.

If the configured rewrite action is:

```
> add rewrite action test_refine_search replace_all http.res.body(10) '
    " testing_refine_search" ' -search text("abc") -refineSearch extend
    (1,1)
And the HTTP response body is abcxxxx456.

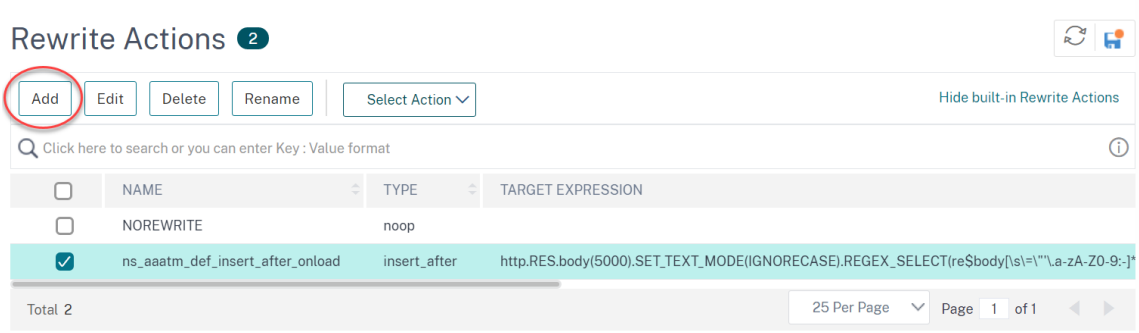
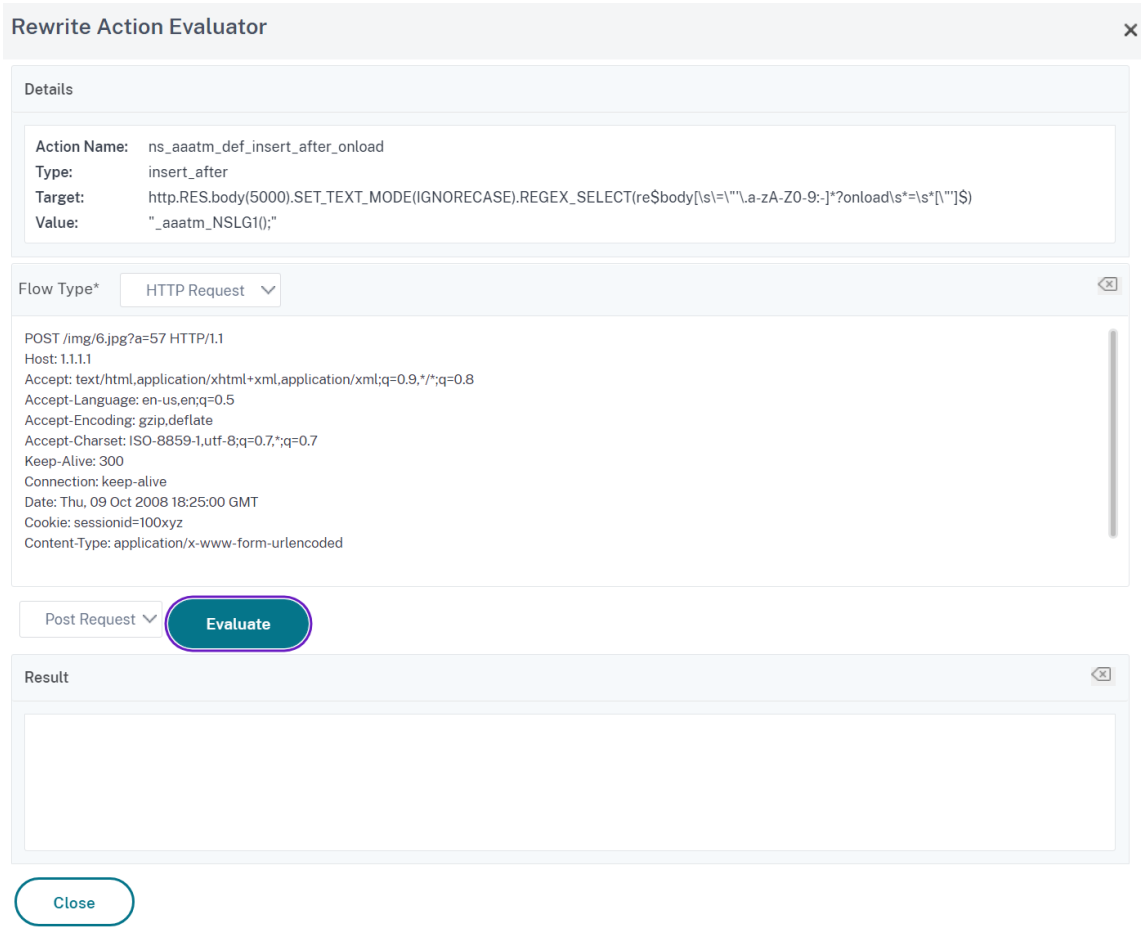
<!--NeedCopy-->
```

Then, the search parameter finds pattern “abc” and since the refineSearch parameter is also configured to check an extra 1 byte to the left and an extra one byte to the right of the matched pattern. The resultant replaced text is: abcx. So, the output of this action is `testing_refine_searchxxx456`.

Example 1:Using the Refine search functionality in INSERT_BEFORE_ALL action type.

```
> add policy patset pat
Done
> bind policy patset pat abcd
Done
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
    (10) 'target.prefix(10) + "refineSearch_testing"' -search patset("
    pat") -refineSearch extend(10,10)
Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
Refine Search:extend(10,10)
Value:target.prefix(10) + "refineSearch_testing"
Search: patset("pat")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

```



```
<!--NeedCopy-->
```

Example 2: Using the Refine search functionality in INSERT_AFTER_ALL action type.

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
   (100) '"refineSearch_testing"' -search text("abc") -refineSearch
   extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

Example 3: Using the Refine search functionality in REPLACE_ALL action type.

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY
   (100000)" "\"https://\"" -search "patset(\"pat_list_2\")" -
   refineSearch "EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done
> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
Search: patset("pat_list_2")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
```



```
20
21 <!--NeedCopy-->
```

Example 4: Using the Refine search functionality in DELETE_ALL action type.

```
1 >add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
     " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
     REGEX_SELECT(re#\\s*<AppData>.*\\s*<\\/AppData>#)"
2 > show REWRITE action refineSearch_act_4
3 Name: refineSearch_act_4
4 Operation: delete_all
5 Target:HTTP.RES.BODY(50000)
6 Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s*<AppData>.*\s*<\/AppData
     >#)
7 Search: text("Windows Desktops")
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12 >
13 <!--NeedCopy-->
```

Example 5: Using the Refine Search functionality in CLIENTLESS_VPN_ENCODE_ALL action type.

”’

add rewrite action act2 clientless_vpn_encode_all http.req.body(100) -search text(“abcd”)
Done
sh rewrite action act2
Name: act1
Operation: clientless_vpn_encode_all
Target:http.req.body(100)
Search: text(“abcd”)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

”’

Example 6: Using the Refine Search functionality in CLIENTLESS_VPN_DECODE_ALL action type.

```
1 > add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
     -search text("abcd")
2 Done
```



```
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

Modify an existing rewrite action by using the command line interface

At the command prompt, type the following commands to modify an existing rewrite action and verify the configuration:

- `set rewrite action <name> [-target<expression>] [-stringBuilderExpr<expression>] [-pattern<expression> | -search <expression>] [-refineSearch <expression>] [-comment<string>]`
- `show rewrite action <name>`

Example:

```
> set rewrite action insertact -target "Client-IP"
 Done
> show rewrite action insertact

Name: insertact
Operation: insert_http_header   Target:Client-IP
Value:CLIENT.IP.SRC
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

Remove a rewrite action by using the command line interface

At the command prompt, type the following commands to remove a rewrite action:

```
rm rewrite action <name>
```

Example:


```
> rm rewrite action insertact
Done

<!--NeedCopy-->
```

Configure a rewrite action by using the configuration utility

1. Navigate to **AppExpert > Rewrite > Actions**.
2. In the details pane, do one of the following:
   - To create an action, click **Add**.
   - To modify an existing action, select the action, and then click **Edit**.
3. Click **Create** or **OK**. A message appears in the status bar, stating that the Action has been configured successfully.
4. Repeat steps 2 through 4 to create or modify as many rewrite actions as you want.
5. Click **Close**.



Add an expression by using the Add Expression dialog box

1. In the **Create Rewrite Action** or **Configure Rewrite Action** dialog box, under the text area for the type argument you want to enter, click **Add**.
2. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.
   - HTTP. The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - SYS. The protected Web sites. Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

- CLIENT. The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

1. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
2. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.
   For more information about the PI expressions language and creating expressions for responder policies, see "Policies and Expressions."

If you want to test the effect of a rewrite action when used on sample HTTP data, you can use the Rewrite Expression Evaluator.

Rewrite TCP payloads

Target expressions in actions for TCP rewrite must begin with one of the following expression prefixes:

- **CLIENT.TCP.PAYLOAD.** For rewriting TCP payloads in client requests. For example, CLIENT.TCP.PAYLOAD(10000).AFTER_STR("string1").
- **SERVER.TCP.PAYLOAD.** For rewriting TCP payloads in server responses. For example, SERVER.TCP.PAYLOAD(1000).B64DECODE.BETWEEN("string1","string2").

Evaluate a rewrite action by using the Rewrite Action Evaluator dialog box

1. In the **Rewrite Actions** details pane, select the rewrite action that you want to evaluate, and then click **Evaluate**.
2. In the Rewrite Expression Evaluator dialog box, specify values for the following parameters. (An asterisk indicates a required parameter.)

   Rewrite Action—If the rewrite action you want to evaluate is not already selected, select it from the drop-down list. After you select a Rewrite action, the Details section displays the details of the selected Rewrite action.
   New—Select New to open the Create Rewrite Action dialog box and create a rewrite action.
   Modify—Select Modify to open the Configure Rewrite Action dialog box and modify the selected rewrite action.
   Flow Type—Specifies whether to test the selected rewrite action with HTTP Request data or

HTTP Response data. The default is Request. If you want to test with Response data, select Response.

HTTP Request/Response Data*—Provides a space for you to provide the HTTP data that the Rewrite Action Evaluator is used for testing. You can paste the data directly into the window, or click Sample to insert some sample HTTP headers.

Show end-of-line—Specifies whether to show UNIX-style end-of-line characters (\n) at the end of each line of sample HTTP data.

Sample—Inserts sample HTTP data into the HTTP Request/Response Data window. You can choose either GET or POST data.

Browse—Opens a local browse window so that you can choose a file containing sample HTTP data from a local or network location.

Clear—Clears the current sample HTTP data from the HTTP Request/Response Data window.

3. Click Evaluate. The **Rewrite Action Evaluator** evaluates the effect of the Rewrite action on the sample data that you chose, and displays the results as modified by the selected **Rewrite** action in the **Results** window. Additions and deletions are highlighted as indicated in the legend in the lower left-hand corner of the dialog box.
4. Continue evaluating Rewrite actions until you have determined that all of your actions have the effect that you wanted.
    - You can modify the selected rewrite action and test the modified version by clicking **Modify** to open the **Configure Rewrite Action** dialog box, making and saving your changes, and then clicking Evaluate again.
    - You can evaluate a different rewrite action using the same request or response data by selecting it from the **Rewrite Action** drop-down list, and then clicking **Evaluate** again.
5. Click **Close** to close the **Rewrite Expression Evaluator** and return to the **Rewrite Actions** pane.
6. To delete a rewrite action, select the rewrite action you want to delete, then click **Remove** and, when prompted, confirm your choice by clicking **OK**.



Configuring a Rewrite Policy

September 14, 2021

After you create any needed rewrite action, you must create at least one rewrite policy to select the requests that you want the Citrix ADC appliance to rewrite.

A rewrite policy consists of a rule, which itself consists of one or more expressions. And an associated action that is done if a request or response matches the rule. Policy rules for evaluating HTTP requests and responses can be based on almost any part of a request or response.

You can't use TCP rewrite actions to rewrite data other than the TCP payload. You can base the policy rules for TCP rewrite policies on the information in the transport layer. And the layers below the transport layer.

If a configured rule matches a request or response, the corresponding policy is triggered and the action associated is carried out.

> **Note:**
> You can use either the command line interface or the configuration utility to create and configure rewrite policies. Users who are not thoroughly familiar with the command line interface and the Citrix ADC Policy expression language will usually find using the configuration utility much easier.

To add a new rewrite policy by using the command line interface

At the command prompt, type the following commands to add a new rewrite policy and verify the configuration:

- <add rewrite policy <name> <expression> <action> [<undefaction>]
- <show rewrite policy <name>

Example 1. Rewriting HTTP Content:

```
> add rewrite policy policyNew "HTTP.RES.IS_VALID" insertact NOREWRITE
 Done
> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertact
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

Example 2. Rewriting a TCP Payload (TCP Rewrite):

```
> add rewrite policy client_tcp_payload_policy CLIENT.IP.SRC.EQ
   (172.168.12.232) client_tcp_payload_replace_all
 Done
> show rewrite policy client_tcp_payload_policy
        Name: client_tcp_payload_policy
        Rule: CLIENT.IP.SRC.EQ(172.168.12.232)
        RewriteAction: client_tcp_payload_replace_all
        UndefAction: Use Global
        LogAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
>
<!--NeedCopy-->
```


To modify an existing rewrite policy by using the command line interface

At the command prompt, type the following commands to modify an existing rewrite policy and verify the configuration:

- <set rewrite policy <name>-rule <expression-action <action> [<undefaction>]
- <show rewrite policy <name>

Example:

```
> set rewrite policy policyNew -rule "HTTP.RES.IS_VALID" -action
   insertaction
 Done

> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertaction
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

To remove a rewrite policy by using the command line interface

At the command prompt, type the following command to remove a rewrite policy:

rm rewrite policy <name>

Example:

```
> rm rewrite policy policyNew
Done
<!--NeedCopy-->
```

To configure a rewrite policy by using the configuration utility

1. Navigate to **AppExpert > Rewrite > Policies**.
2. In the details pane, do one of the following:
   - To create a policy, click Add.

- To modify an existing policy, select the policy, and then click Open.

3. Click **Create** or **OK**. A message appears in the status bar, stating that the Policy has been configured successfully.
4. Repeat steps 2 through 4 to create or modify as many rewrite actions as you want.
5. Click **Close**. To delete a rewrite policy, select the rewrite policy you want to delete, then click **Remove** and, when prompted, confirm your choice by clicking **OK**.

Create rewrite policy for content security headers, XSS protection, HSTS, X-Content-Type-Options, and Content-Security-Policy

At the command prompt, type the following rewrite action commands to add Security header to webpages served through NetScaler using rewrites.

```
add rewrite action insert_STS_header insert_http_header Strict-Transport-Security "\"max-age=157680000\""
add rewrite action rw_act_insert_XSS_header insert_http_header X-Xss-Protection "\"1; mode=block\""
add rewrite action rw_act_insert_Xcontent_header insert_http_header X-Content-Type-Options "\"nosniff\""
add rewrite action rw_act_insert_Content_security_policy insert_http_header Content-Security-Policy "\"default-src \'self\' ; script-src \'self\' \'unsafe-inline\' \'unsafe-eval\' ; style-src \'self\' \'unsafe-inline\' \'unsafe-eval\'; img-src \'self\' data:\""
<!--NeedCopy-->
```

At the command prompt, type the following rewrite policy commands to add Security header to webpages served through NetScaler using rewrites.

```
add rewrite policy enforce_STS true insert_STS_header
add rewrite policy rw_pol_insert_XSS_header "HTTP.RES.HEADER("X-Xss-Protection").EXISTS.NOT" rw_act_insert_XSS_header
add rewrite policy rw_pol_insert_XContent TRUE rw_act_insert_Xcontent_header
add rewrite policy rw_pol_insert_Content_security_policy TRUE rw_act_insert_Content_security_policy
<!--NeedCopy-->
```

At the command prompt, type the following commands to bind policies to virtual server on Response using Goto Expression NEXT.

```
bind vpn vserver access -policy enforce_STS -priority 100 -gotoPriorityExpression NEXT -type RESPONSE
```



```
bind vpn vserver "VSERVERNAME" -policy rw_pol_insert_XSS_header -
    priority 110 -gotoPriorityExpression NEXT -type RESPONSE
bind vpn vserver access -policy rw_pol_insert_XContent -priority 120 -
    gotoPriorityExpression NEXT -type RESPONSE
bind vpn vserver access -policy rw_pol_insert_Content_security_policy -
    priority 130 -gotoPriorityExpression NEXT -type RESPONSE
<!--NeedCopy-->
```

Configure rewrite policy for content security headers, XSS protection, HSTS, X-Content-Type-Options, and Content-Security-Policy using configuration utility

1. Navigate to **AppExpert** > **Rewrite** > **Actions**
2. Click **Add** to create rewrite actions for each one of the headers.
3. Navigate to **AppExpert** > **Rewrite** > **Policies**
4. Click **Add** to create rewrite policies and link them to actions.
5. Bind policies to virtual server on Response using the Goto Expression **NEXT**.

> **Note:**
> In SSLVPN, we need to use the below Content-Security Action:

```
add rewrite action Rewrite_Insert_Content-Security-Policy
    insert_http_header Content-Security-Policy "\"default-src \'self\' ;
     script-src \'self\' \'unsafe-inline\' \'unsafe-eval\' ; style-src
    \'self\' \'unsafe-inline\' \'unsafe-eval\'; img-src \'self\' http://
    localhost:* data:;\"
<!--NeedCopy-->
```

The localhost exception is required because the browser passes the cookie/GW information to the plug-in using localhost HTTP call. Since the CSP had only “self”, only calls to the virtual server would be allowed.

Binding a Rewrite Policy

September 14, 2021

After creating a rewrite policy, you must bind it to put it into effect. You can bind your policy to Global if you want to apply it to all traffic that passes through your Citrix ADC, or you can bind your policy to a specific virtual server or bind point to direct only that virtual server or bind point’s incoming traffic to that policy. If an incoming request matches a rewrite policy, the action associated with that policy is carried out.

Rewrite policies for evaluating HTTP requests and responses can be bound to virtual servers of type HTTP or SSL, or they can be bound to the REQ_OVERRIDE,
REQ_DEFAULT, RES_OVERRIDE, and RES_DEFAULT bind points. Rewrite policies for TCP rewrite can be bound only to virtual servers of type TCP or SSL_TCP, or to the OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, OTHERTCP_RES_OVERRIDE, and
OTHERTCP_RES_DEFAULT bind points.

Note: The term OTHERTCP is used in the context of the Citrix ADC appliance to refer to all TCP or SSL_TCP requests and responses that you want to treat as a raw stream of bytes regardless of the protocols that the TCP packets encapsulate.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the Citrix ADC operating system, policy priorities work in reverse order - the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is applied first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000.

Unlike most other features in the Citrix ADC operating system, the rewrite feature continues to evaluate and implement policies after a request matches a policy. However, the effect of a particular action policy on a request or response will often be different depending on whether it is performed before or after another action. Priority is important to get the results you intended.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind it. If you do this, you can add additional policies at any time without having to reassign the priority of an existing policy.

When binding a rewrite policy, you also have the option of assigning a goto expression (gotoPriorityExpression) to the policy. A goto expression can be any positive integer that matches the priority assigned to a different policy that has a higher priority than the policy that contains the goto expression. If you assign a goto expression to a policy, and a request or response matches the policy, the Citrix ADC will immediately go to the policy whose priority matches the goto expression. It will skip over any policies with priority numbers that are lower than that of the current policy, but higher than the priority number of the goto expression, and not evaluate those policies.

To globally bind a rewrite policy by using the command line interface

At the command prompt, type the following commands to globally bind a rewrite policy and verify the configuration:

- `bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]`

- show rewrite global

Example:

```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)      Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

 Done
<!--NeedCopy-->
```

To bind rewrite policy to a specific virtual server by using the command line interface

At the command prompt, type the following commands to bind rewrite policy to a specific virtual server and verify the configuration:

- bind lb vserver <name>@ (<serviceName>@ [-weight <positive_integer>])| <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type (REQUEST | RESPONSE)] [-invoke (<labelType> <labelName>)])
- show lb vserver <name>

Example:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
        Time since last state change: 28 days, 01:57:26.350
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
```



```
16          Mode: IP
17          Persistence: NONE
18          Vserver IP and Port insertion: OFF
19          Push: DISABLED  Push VServer:
20          Push Multi Clients: NO
21          Push Label Rule: none
22
23  1)      Policy : ns_cmp_msapp Priority:50
24  2)      Policy : cf-pol Priority:1      Inherited
25   Done
26  <!--NeedCopy-->
```

To bind a rewrite policy to a bind point by using the configuration utility

1. Navigate to **AppExpert** >**Rewrite** > **Policies**.
2. In the details pane, select the rewrite policy you want to globally bind, and then click **Policy Manager**.
3. In the **Rewrite Policy Manager** dialog box, in the **Bind Points** menu, do one of the following:
   a) If you want to configure bindings for HTTP rewrite policies, click **HTTP**, and then click either **Request** or **Response**, depending on whether you want to configure request-based rewrite policies or response-based rewrite policies.
   b) If you want to configure bindings for TCP rewrite policies, click **TCP**, and then click either **Client** or **Server**, depending on whether you want to configure client-side TCP rewrite policies or server-side TCP rewrite policies.
4. Click the bind point to which you want to bind the rewrite policy. The **Rewrite Policy Manager** dialog box displays all the rewrite policies that are bound to the selected bind point.
5. Click **Insert Policy** to insert a new row and display a drop-down list with all available, unbound rewrite policies.
6. Click the policy you want to bind to the bind point. The policy is inserted into the list of rewrite policies bound to the bind point.
7. In the **Priority** column, you can change the priority to any positive integer. For more information about this parameter, see priority in “Parameters for binding a rewrite policy.”
8. If you want to skip over policies and go directly to a specific policy in the event that the current policy is matched, change the value in the Goto Expression column to equal the priority of the next policy to be applied.. For more information about this parameter, see gotoPriorityExpression in “Parameters for binding a rewrite policy.”
9. To modify a policy, click the policy, and then click **Modify Policy**.
10. To unbind a policy, click the policy, and then click **Unbind Policy**.
11. To modify an action, in the Action column, click the action you want to modify, and then click **Modify Action**.

12. To modify an invoke label, in the **Invoke** column, click the invoke label you want to modify, and then click **Modify Invoke Label**.
13. To regenerate the priorities of all the policies that are bound to the bind point you are currently configuring, click **Regenerate Priorities**. The policies retain their existing priorities relative to the other policies, but the priorities are renumbered in multiples of ten.
14. Click **Apply Changes**.
15. Click **Close**. A message appears in the status bar, stating that the Policy has been configured successfully.

To bind a rewrite policy to a specific virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the details pane list of virtual servers, select the virtual server to which you want to bind the rewrite policy, and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, select the **Policies** tab. All policies configured on your Citrix ADC appear on the list.
4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click **OK**. A message appears in the status bar, stating that the Policy has been configured successfully.

Configuring Rewrite Policy Labels

September 14, 2021

If you want to build a more complex policy structure than is supported by single policies, you can create policy labels and then bind them as you would policies. A policy label is a user-defined point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority you configured. A policy label can include one or multiple policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and the return of control to the policy that invoked the policy label.

A rewrite policy label consists of a name, a transform name that describes the type of policy included in the policy label, and a list of policies bound to the policy label. Each policy that is bound to the policy label contains all of the elements described in Configuring a Rewrite Policy.

Note: You can use either the command line interface or the configuration utility to create and configure rewrite policy labels. Users who are not thoroughly familiar with the command line interface and the Citrix ADC Policy Infrastructure (PI) language will usually find using the configuration utility much easier.

To configure a rewrite policy label by using the command line interface

To add a new rewrite policy label, at the command prompt, type the following command:

add rewrite policylabel <labelName> <transform>

For example, to add a rewrite policy label named polLabelHTTPResponses to group all policies that work on HTTP responses, you would type the following:

add rewrite policylabel polLabelHTTPResponses http_res

To modify an existing rewrite policy label, at the Citrix ADC command prompt, type the following command:

set rewrite policy <name> <transform>

Note: The set rewrite policy command takes the same options as the add rewrite policy command.

To remove a rewrite policy label, at the Citrix ADC command prompt, type the following command:

rm rewrite policy<name>

For example, to remove a rewrite policy label named polLabelHTTPResponses, you would type the following:

rm rewrite policy polLabelHTTPResponses

To configure a rewrite policy label by using the configuration utility

1. Navigate to **AppExpert** > **Rewrite** > **Policy Labels**.
2. In the details pane, do one of the following:
   - To create a new policy label, click **Add**.
   - To modify an existing policy label, select the policy, and then click **Open**.
3. Add or remove policies from the list that is bound to the policy label.
   - To add a policy to the list, click **Insert Policy**, and choose a policy from the drop-down list. You can create a new policy and add it to the list by choosing New Policy in the list, and following the instructions in Configuring a Rewrite Policy.
   - To remove a policy from the list, select that policy, and then click Unbind Policy.
4. Modify the priority of each policy by editing the number in the Priority column.

   You can also automatically renumber policies by clicking Regenerate Priorities.
5. Click **Create** or **OK**, and then click **Close**.

   To remove a policy label, select it, and then click **Remove**. To rename a policy label, select it and then click **Rename**. Edit the name of the policy, and then click **OK** to save your changes.

Configuring the Default Rewrite Action

September 14, 2021

An undefined event is triggered when the Citrix ADC cannot evaluate a policy, usually because it detects a logical or other error in the policy or an error condition on the Citrix ADC. When the rewrite policy evaluation results in an error, the specified undefined action is carried out. Undefined actions configured at the rewrite policy level are carried out before a globally configured undefined action.

The Citrix ADC supports following three types of undefined actions:

- undefAction NOREWRITE

  Aborts rewrite processing, but does not alter the packet flow. This means that the Citrix ADC continues to process requests and responses that do not match any rewrite policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers, and is the default setting.

- undefAction RESET

  Resets the client connection. This means that the Citrix ADC tells the client that it must reestablish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

- undefAction DROP

  Silently drops the request without responding to the client in any way. This means that the Citrix ADC simply discards the connection without responding to the client. This action is appropriate for requests that appear to be part of a DDoS attack or another sustained attack on your servers.

Note: Undefined events can be triggered for both request and response flow specific policies.

To configure the default action by using the command line interface

At the command prompt, type the following commands to configure the default action and verify the configuration:

```
<set rewrite param                RESET                        DROP )
-undefAction ( NOREWRITE
```

-

- <show rewrite param

Example:

```
> set rewrite param -undefAction NOREWRITE
 Done
> show rewrite param
        Action Name: NOREWRITE
 Done
<!--NeedCopy-->
```

To configure the default action by using the configuration utility

1. Navigate to AppExpert > Rewrite.
2. In the details pane, under Rewrite Overview, click the Change Rewrite Settings link. The Set Rewrite Params dialog box appears.
3. Under Global Undefined-Result Action, select an option as follows:
   - NoRewrite—NOREWRITE
   - Reset—RESET
   - Drop—DROP
4. Click OK. The global undefined action is set to the value you chose.

Bypassing the Safety Check

September 16, 2021

> **Warning:**
>
> The Bypass Safety Check parameter is deprecated from NetScaler 12.0 build 56.20 onwards. However, Bypass Safety Check parameter is removed and no longer available on the Citrix ADC appliance release 13.1 onwards.

When you create a rewrite action, the Citrix ADC verifies that the expression you used to create the action is safe. Expressions created by the Citrix ADC from run-time data, such as URLs contained in HTTP requests, can cause unexpected errors. The Citrix ADC reports expressions that cause such errors as unsafe expressions.

In some cases, the expressions may be safe. For example, the Citrix ADC cannot validate an expression that contains a URL that does not resolve, even if the URL does not resolve because the Web server is temporarily unavailable. You can manually bypass the Safety Check to allow these expressions.

To bypass the safety check by using the command line interface

At the command prompt, type the following commands to bypass the safety check and verify the configuration:

- <set rewrite action <name> -bypassSafetyCheck YES
- <show rewrite action <name>

Example:

```
> set rewrite action insertact -bypassSafetyCheck YES
 Done
> show rewrite action insertact

        Name: insertact
        Operation: insert_http_header   Target:Client-IP
        Value:CLIENT.IP.SRC
        BypassSafetyCheck : YES
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 2
 Done
<!--NeedCopy-->
```

To bypass safety check by using the configuration utility

1. Navigate to **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, select the rewrite action to be exempted from the safety check, and then click **Open**.
3. In the **Configure Rewrite Action** dialog box, select the **Bypass Safety Check** check box.
4. Click **OK**.

Rewrite Action and Policy Examples

September 14, 2021

The examples in this section demonstrate how to configure rewrite to perform various useful tasks. The examples occur in the server room of Example Manufacturing Inc., a mid-sized manufacturing company that uses its Web site to manage a considerable portion of its sales, deliveries, and customer support.

Example Manufacturing has two domains: example.com for its Web site and email to customers, and example.net for its intranet. Customers use the Example Web site to place orders, request quotes,

research products, and contact customer service and technical support.

As an important part of Example's revenue stream, the Web site must respond quickly and keep customer data confidential. Example therefore has several Web servers and uses Citrix ADC appliances to balance the Web site load and manage traffic to and from its Web servers.

The Example system administrators use the rewrite features to perform the following tasks:

Example 1: Delete old X-Forwarded-For and Client-IP Headers

Example Inc. removes old X-Forwarded-For and Client-IP HTTP headers from incoming requests.

Example 2: Adding a Local Client-IP Header

Example Inc. adds a new, local Client-IP header to incoming requests.

Example 3: Tagging Secure and Insecure Connections

Example Inc. tags incoming requests with a header that indicates whether the connection is a secure connection.

Example 4: Mask the HTTP Server Type

Example Inc. modifies the HTTP Server: header so that unauthorized users and malicious code cannot use that header to determine the HTTP server software it uses.

Example 5: Redirect an External URL to an Internal URL

Example Inc. hides information about the actual names of its Web servers and the configuration of its server room from users, to make URLs on its Web site shorter and easier to remember and to improve security on its site.

Example 6: Migrating Apache Rewrite Module Rules

Example Inc. moved its Apache rewrite rules to a Citrix ADC appliance, translating the Apache PERL-based script syntax to the Citrix ADC rewrite rule syntax.

Example 7: Marketing Keyword Redirection

The marketing department at Example Inc. sets up simplified URLs for certain predefined keyword searches on the company's Web site.

Example 8: Redirect Queries to the Queried Server.

Example Inc. redirects certain query requests to the appropriate server.

Example 9: Home Page Redirection

Example Inc. recently acquired a smaller competitor, and it now redirects requests to the acquired company's home page to a page on its own Web site.

Example 10: Policy-based RSA Encryption

Example Inc. encrypt HTTP predefined and user-defined header or body content by using PEM RSA public key.

Each of these tasks requires that the system administrators create rewrite actions and policies and bind them to a valid bind point on the Citrix ADC.

Example 1: Delete Old X-Forwarded-For and Client-IP Headers

September 14, 2021

Example Inc. wants to remove old X-Forwarded-For and Client-IP HTTP headers from incoming requests, so that the only X-Forwarded-For headers that appear are the ones added by the local server. This configuration can be done through the Citrix ADC command line or the configuration utility. The Example Inc. system administrator is an old-school networking engineer and prefers to use a CLI where possible, but wants to be sure he understands the configuration utility interface so that he can show new system administrators on the team how to use it.

The examples below demonstrate how to perform each configuration with both the CLI and the configuration utility. The procedures are abbreviated on the assumption that users will already know the basics of creating rewrite actions, creating rewrite policies, and binding policies.

- For more detailed information about creating rewrite actions, see Configuring a Rewrite Action.
- For more detailed information about creating rewrite policies, see Configuring a Rewrite Policy.
- For more detailed information about binding rewrite policies, see Binding a Rewrite Policy.

To delete old X-Forwarded and Client-IP headers from a request by using the command line interface

At the command prompt, type the following commands in the order shown:

```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER("x-forwarded-for").
    EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER("client-ip").EXISTS'
    act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
<!--NeedCopy-->
```


To delete old X-Forwarded and Client-IP headers from a request by using the configuration utility

In the Create Rewrite Action dialog box, create two rewrite actions with the following descriptions.

Name	Type	Argument(s)
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

In the Create Rewrite Policy dialog box, create two rewrite policies with the following descriptions.

Name	Expression	Action
pol_check_xfor	'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'	act_del_xfor
pol_check_cip	'HTTP.REQ.HEADER("client-ip").EXISTS'	act_del_cip

Bind both policies to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_check_xfor	100	200
pol_check_cip	200	300

All old X-Forwarded-For and Client-IP HTTP headers are now deleted from incoming requests.

Example 2: Adding a Local Client-IP Header

September 14, 2021

Example Inc. wants to add a local Client-IP HTTP header to incoming requests. This example contains two slightly different versions of the same basic task.

To add a local Client-IP header by using the command line interface

At the command prompt, type the following commands in the order shown:


```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' act_ins_client
bind rewrite global pol_ins_client 300 END
<!--NeedCopy-->
```

To add a local Client-IP header by using the configuration utility

In the Create Rewrite Action dialog box, create a rewrite action with the following description.

Name	Type	Argument(s)
act_ins_client	insert_http_header	NS-Client 'CLIENT.IP.SRC'

In the Create Rewrite Policy dialog box, create a rewrite policy with the following description.

Name	Expression	Action
pol_ins_client	'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS'	act_ins_client

Bind policy to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_ins_client	100	Next

Example 3: Tagging Secure and Insecure Connections

September 14, 2021

Example Inc. wants to tag incoming requests with a header that indicates whether or not the connection is a secure connection. This helps the server keep track of secure connections after the Citrix ADC has decrypted the connections.

To implement this configuration, you would begin by creating rewrite actions with the values shown in the following tables. These actions label connections to port 80 as insecure connections, and connections to port 443 as secure connections.

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	YES

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	NO

You would then create a rewrite policy with the values shown in the following tables. These policies check incoming requests to determine which requests are directed to port 80 and which are directed to port 443. The policies then add the correct SSL header.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

Finally, you would bind the rewrite policies to Citrix ADC, assigning the first policy a priority of 200, and the second a priority of 300, and setting the goto expression of both policies to END.

Each incoming connection to port 80 now has an SSL:NO HTTP header added to it and each incoming connection to port 443 has an SSL:YES HTTP header added to it.

Example 4: Mask the HTTP Server Type

September 14, 2021

Example Inc. wants to modify the HTTP Server: header so that unauthorized users and malicious code cannot use the header to identify the software that the HTTP server uses.

To modify the HTTP Server: header, you would create a rewrite action and a rewrite policy with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Server_Mask	REPLACE	HTTP.RES.HEADER("Se	"Web Server 1.0"

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

Example commands:

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server")"\"Web Server 1.0\""
```

```
> add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
```

You would then globally bind the rewrite policy, assigning a priority of 100 and setting the Goto Priority Expression of the policy to END.

The HTTP Server: header is now modified to read "Web Server 1.0," masking the actual HTTP server software used by the Example Inc. Web site.

Example 5: Redirect an external URL to an internal URL

September 14, 2021

Example Inc. wants to hide its actual server room configuration from users to improve security on its Web servers.

To do this, you would create a rewrite action with the values as shown in the following tables. For request headers, the action in the table modifies `www.example.com` to web.hq.example.net. For response headers, the action does the opposite, translating `web.hq.example.net` to `www.example.com`.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Request_Server_Replac	REPLACE	HTTP.REQ.HOSTNAME.S	“Web.hq.example.net”
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER(“Server”)	“www.example.com”

The first policy checks incoming requests to see if they are valid, and if they are, it performs the Action-Rewrite-Request_Server_Replace action. The second policy checks responses to see if they originate at the server `web.hq.example.net`. If they do, it performs the Action-Rewrite-Response_Server_Replace action.

Examples of rewrite action and policy for redirecting an external URL.

```
add rewrite action Action-Rewrite-Request_Server_Replace REPLACE HTTP.REQ.
HOSTNAME.SERVER '"Web.hq.example.net"
```

```
add rewrite action Action-Rewrite-Response_Server_Replace REPLACE HTTP.RES.
HEADER("Server")'"www.example.com"'
```

```
add rewrite policy Policy-Rewrite-Request_Server_Replace HTTP.REQ.HOSTNAME.
SERVER.EQ("www.example.com")Action-Rewrite-Request_Server_Replace NOREWRITE
```

```
add rewrite policy Policy-Rewrite-Response_Server_Replace HTTP.REQ.HEADER("
Server").EQ("Web.hq.example.net")Action-Rewrite-Response_Server_Replace
```

Finally, you would bind the rewrite policies, assigning each a priority of 500 because they are in different policy banks and therefore will not conflict. You should set the goto expression to NEXT for both bindings.

```
bind rewrite global Policy-Rewrite-Request_Server_Replace 500 END -type
REQ_DEFAULT
```

```
bind rewrite global Policy-Rewrite-Response_Server_Replace 500 END -type
RES_DEFAULT
```

All instances of `www.example.com` in the request headers are now changed to `web.hq.example.net`, and all instances of `web.hq.example.net` in response headers are now changed to `www.example.com`.

Example 6: Migrating Apache Rewrite Module Rules

September 14, 2021

Example Inc., is currently using the Apache rewrite module to process search requests sent to its Web servers and redirect those requests to the appropriate server on the basis of information in the request URL. Example Inc. wants to simplify its setup by migrating these rules onto the Citrix ADC platform.

Several Apache rewrite rules that Example currently uses are shown below. These rules redirect search requests to a special results page if they do not have a SiteID string or if they have a SiteID string equal to zero (0), or to the standard results page if these conditions do not apply.

The following are the current Apache rewrite rules:

- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ results2.html [P,L]
- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results.html [P,L]

To implement these Apache rewrite rules on the Citrix ADC, you would create rewrite actions with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Display_Results_NulSit	REPLACE	HTTP.REQ.URL	“/results2.html”
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	“/results2.html”

You would then create rewrite policies with the values as shown in the tables below.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Display_Results_NulSit	Action-Rewrite-Display_Results_NulSit	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MOD && (!HTTP.REQ.URL.QUERY.CONTAINS("S || HTTP.REQ.URL.QUERY.CONTAINS("Si || HTTP.REQ.URL.QUERY.SET_TEXT_MO
Policy-Rewrite-Display_Results	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MOD || HTTP.REQ.URL.QUERY.SET_TEXT_MO

Finally, you would bind the rewrite policies, assigning the first a priority of 600 and the second a priority of 700, and then set the goto expression to NEXT for both bindings.

The Citrix ADC now handles these search requests exactly as the Web server did before the Apache rewrite module rules were migrated.

Example 7: Marketing Keyword Redirection

September 14, 2021

The marketing department at Example Inc. wants to set up simplified URLs for certain predefined keyword searches on the company's Web site. For these keywords, it wants to redefine the URL as shown below.

- External URL:

```
http://www.example.com/\<marketingkeyword\>
```

- Internal URL:

```
http://www.example.com/go/kwsearch.asp?keyword=\<marketingkeyword\>
```

To set up redirection for marketing keywords, you would create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target location	String expression for replacement text
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GE	””go/kwsearch.aspkeyword=”l”

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“v

Finally, you would bind the rewrite policy, assigning it a priority of 800. Unlike the previous rewrite policies, this policy should be the last to be applied to a request that matches its criteria. For this reason, Citrix ADC administrator sets its Goto Priority Expression to END.

Any request using a marketing keyword is redirected to the keyword search CGI page, whereupon a search is performed and all remaining policies are skipped.

Example 8: Redirect Queries to the Queried Server

September 14, 2021

Example Inc. wants to redirect query requests to the appropriate server, as shown here.

- <Request: GET /query.cgi?server=5HOST: www.example.com
- <Redirect URL: <http://web-5.example.com/>

To implement this redirection, you would first create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_Hostheader	REPLACE	HTTP.REQ.HEADER(“Hc	“server-“ + ıple.com”) HTTP.REQ.URL.QUERY.VALUE(“web”)

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	HTTP.REQ.HEADER("Host").EQ("www.example.com")

Example commands:

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server")"\"Web Server 1.0\""
Done

> add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
Done
```

Finally, you would bind the rewrite policy, assigning it a priority of 900. Because this policy should be the last policy applied to a request that matches its criteria, you set the goto expression to END.

Incoming requests to any URL that begins with <http://www.example.com/query.cgi?server>= are redirected to the server number in the query.

Example 9: Home Page Redirection

September 14, 2021

New Company, Inc. recently acquired a smaller competitor, Purchased Company, and wants to redirect the home page for Purchased Company to a new page on its own Web site, as shown here.

- Old URL: http://www.purchasedcompany.com/*
- New URL: http://www.newcompany.com/products/page.htm

To redirect requests to the Purchased Company home page, you would create rewrite actions with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_A	“/products/page.htm”
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	“www.newcompany.com”

You would then create rewrite policies with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ(“
Policy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“v
Policy-Rewrite-Replace-URL	Action-Rewrite-Replace_URL	NOREWRITE	HTTP.REQ.IS_VALID

Finally, you would bind the rewrite policies globally, assigning the first a priority of 100, the second a priority of 200, and the third a priority of 300. These policies should be the last policies applied to a request that matches the criteria. For this reason, set the goto expression to END for the first and third policies, and to 300 for the second policy. This ensures that all remaining requests are processed correctly.

Requests to the acquired company’s old Web site are now redirected to the correct page on the New Company home page.

Example 10: Policy-based RSA Encryption

September 14, 2021

The RSA algorithm uses the PKEY_ENCRYPT_PEM() function to encrypt HTTP predefined and user-defined header or body content. The function accepts only RSA public keys (not private keys) and the encrypted data cannot be longer than the length of the public key. When the data being encrypted is shorter than the key length, the algorithm uses RSA_PKCS1 padding method.

In a sample scenario, the function can be used with B64ENCODE() function in a rewrite action to replace an HTTP header value with a value encrypted by an RSA public key. The data being encrypted is then decrypted by the recipient using the RSA private key.

You can implement the feature by using a rewrite policy. To do this, you must complete the following tasks:

1. Add RSA public key as a policy expression.
2. Create rewrite action.
3. Create rewrite policy.
4. Bind rewrite policy as global.
5. Verify RSA encryption

Policy-based RSA encryption by using Citrix ADC command interface

Complete the following tasks to configure policy-based RSA encryption by using the Citrix ADC command interface.

To add RSA public key as a policy expression by using the Citrix ADC command interface:

```
add policy expression pubkey '"-----BEGIN RSA PUBLIC KEY-----
    MIGJAoGBAKl5vgQEj73Kxp+9
    yn1v5gPR1pnc4oLM2a0kaWwBOsB6rzCIy6znwnvwCY1xRvQhRlJSAyJbloL7wZFIJ2FOR8Cz
    +8ZQWXU2syG+udi4EnWqLgFYowF9zK+o79az597eNPAjsHZ/C2oL/+6qY5a/
    f1z8bQPrHC4GpFfAEJhh/+NnAgMBAAE=-----END RSA PUBLIC KEY-----"'
<!--NeedCopy-->
```

To add rewrite an action to encrypt an HTTP header request by using the Citrix ADC command interface:

```
add rewrite action encrypt_act insert_http_header encrypted_data

HTTP.REQ.HEADER("data_to_encrypt").PKEY_ENCRYPT_PEM(pubkey).B64ENCODE
```

To add rewrite policy by using the Citrix ADC command interface:

```
add rewrite policy encrypt_pol 'HTTP.REQ.HEADER("data_to_encrypt").
    EXISTS' encrypt_act
<!--NeedCopy-->
```

To bind rewrite policy global by using the Citrix ADC command interface:

```
bind rewrite global encrypt_pol 10 -type RES_DEFAULT
```

To verify RSA encryption by using the Citrix ADC command interface:

```
>curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
    http://10.217.24.7/`
```



```
2
3  * About to connect() to 10.217.24.7 port 80 (#0)
4
5  *  Trying 10.217.24.7...
6
7  *  connected
8
9  *  Connected to 10.217.24.7 (10.217.24.7) port 80 (#0)
10
11 > GET / HTTP/1.1
12 > User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
      OpenSSL/0.9.8y zlib/1.2.3
13 > Host: 10.217.24.7
14 > Accept: */*
15 > data_to_encrypt: Now is the time that tries men's souls
16 >
17 < HTTP/1.1 200 OK
18 < Date: Mon, 09 Oct 2017 05:22:37 GMT
19 < Server: Apache/2.2.24 (FreeBSD) mod_ssl/2.2.24 OpenSSL/0.9.8y DAV/2
20 < Last-Modified: Thu, 20 Feb 2014 20:29:06 GMT
21 < ETag: "6bd9f2-2c-4f2dc5b570880"
22 < Accept-Ranges: bytes
23 < Content-Length: 44
24 < Content-Type: text/html
25 < encrypted_data: UliegKBJqZd7JdaC49XMLEK1+eQN2rEfevypW91gKvBVlaKM9N9/
      C2BKuztS99SE0xQaisidzN5IgeIcpQMn+
      CiKYVlLzPG1RuhGaqHYzIt6C8A842da7xE4OlV5SHwScqkqZ5aVrXc3EwtUksna7jOLr40aLeXnnB
      /DB11pUAE=
26 <
27 * Connection #0 to host 10.217.24.7 left intact
28 <html><body><h1>It works!</h1></body></html>* Closing connection #0
29
30 <!--NeedCopy-->
```

Subsequent execution of this curl command with the same data to encrypt shows that the encrypted data is different each execution. This is because the padding inserts random bytes at the beginning of the data to encrypt, causing the encrypted data to be different each time.

```
1  >curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
      http://10.217.24.7/`
2
3  < encrypted_data:
      DaOjtl1Pl4DlQKf58MMeL4cFwFvZwhjMqv5aUYM5Iyzk4UpwIYhpRvgTNu2lXEVc1H0tcR1EGC
      /ViQncLc4EbTurCWLbzjce3+fknnMmzF0lRT6ZZXWbMvsNFOxDA1SnuAgwxWXy/
      ooe9Wy6SYsL2oi1sr5wTG+RihDd9zP+P14=
```



```
4
5 >curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
     http://10.217.24.7/
6
7 . . .
8
9 < encrypted_data: eej6YbGP68yHn48qFUvi+fkG+OiO8j3yYLScrRBU+
     TPQ8WeDVaWnDNAVLvL0ZYHHAU1W2YDRYb+8
     cdKHLpW36QbI6Q5FfBuWKZSI2hSyUvypTpCoAYcHXFv0ns+tRtg0EPNNj+
     lyGjKQWtFi6K8IXXISoDy42FblKIlaA7gEriY=
10 <!--NeedCopy-->
```

Policy-based RSA encryption by using the GUI

The GUI enables you to complete the following tasks:

To add RSA public key as a policy expression by using the GUI:

1. Sign into the Citrix ADC appliance and navigate to **Configurations** > **AppExpert** > **Advanced Expressions**.
2. In the details pane, click **Add** to define an RSA public key as an advanced policy expression.
3. In Create Expression page, set the following parameters:
   a) Expression name. Name of the advanced expression.
   b) Expression. Define RSA public key as an advanced expression using the Expression Editor.
   c) Comments. A brief description of the expression.
4. Click **Create**.

To add rewrite an action to encrypt an HTTP header request by using the GUI:

1. Sign into the Citrix ADC appliance and navigate to **Configurations** > **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, click **Add** to add a rewrite action.
3. In the **Create Rewrite Action** screen, set the following parameters:
   a) Name. Name of the rewrite action.
   b) Type. Select action type as INSERT_HTTP_HEADER.
   c) Use the action type to insert a header. Enter the name of the HTTP header that needs to be rewritten.
   d) Expression. Name of the advanced policy expression associated to the action.
   e) Comments. A brief description of the rewrite action.
4. Click **Create**.

To add rewrite advanced policy by using the GUI:

1. Sign into the Citrix ADC appliance and navigate to **Configurations** > **AppExpert** > **Rewrite** > **Policies**.
2. In the **Rewrite Policies** page, click **Add** to add a rewrite policy.
3. In the **Create Rewrite Policy** page, set the following parameters:
   a) Name. Name of the rewrite policy.
   b) Action. Name of the rewrite action to perform if the request or response matches this rewrite policy.
   c) Log Action. Name of message log action to use when a request matches this policy.
   d) Undefined-Result Action. Action to perform if the result of policy evaluation is undefined.
   e) Expression. Name of the advanced policy expression that triggers the action.
   f) Comments. A brief description of the rewrite action.
4. Click **Create**.

To bind rewrite policy global by using the GUI:

1. Sign into the Citrix ADC appliance and navigate to **Configurations** > **AppExpert** > **Rewrite** > **Policies**.
2. In the **Rewrite Policies** screen, select a rewrite policy that you want to bind and click **Policy Manager**.
3. In the Rewrite Policy Manager page, in the Bind Points section, set the following parameters:
   a) Bind Point. Select the binding point as Default Global.
   b) Protocol. Select the protocol type as HTTP.
   c) Connection Type. Select the connection type as Request.
   d) Click **Continue** to view the **Policy Binding** section.
   e) In the **Policy Binding** section, select the rewrite policy and set the bind parameters.
4. Click **Bind**.

Example 11: Policy-based RSA encryption with no padding operation

September 14, 2021

The PKEY_ENCRYPT_PEM_NO_PADDING() policy function uses the RSA algorithm with no padding operation before performing RSA encryption. The policy function works just like the PKEY_ENCRYPT_PEM() function, except it uses RSA_NO_PADDING method instead of RSA_PKCS1_PADDING. The pkey parameter is a text string with a PEM-encoded RSA public key. Similar to PKEY_ENCRYPT_PEM(), you can use a policy expression for the key.

You can implement the feature by using a rewrite policy. To do this, you must complete the following tasks:

1. Add RSA public key as a policy expression.

2. Create rewrite action.

Policy-based RSA encryption by using Citrix ADC command interface

Complete the following tasks to configure policy-based RSA encryption by using the Citrix ADC command interface.

To add RSA public key with no padding policy expression by using the Citrix ADC command interface:

```
add expression rsa_pub_key_4096 '"-----BEGIN RSA PUBLIC KEY-----" + "
    MIICCgKCAgEArrwBldKd48xrpOSRPMrg+eNAO0ODU6t5b/WYQLdElqNv7WpefBrA" +
    \"nwI2s619gEU1r4zoLqL7l5ALtt5Z+F0JBYfOzBzOky0GtEJ5iX5GP4QxT65J3nHH"
    + "4MTF3acmjvXxclmaKXEFlaVIzW7FTr3Luw/CnOjflAB403Q6F9VBVvQmOVYWnqoI"
     + \"+0q1VIg6Q1pAcvdKBiOf85BBofE5EIbZ/1Jt0CdbsV568l+8
    ve7BnSUncFHoRR3O" + "/
    VfSsDuNWZf7n3RNMzxEuIA72UGPzNYFQzvcPOdzd0aN7jAXw0mgC/NSvKzGKHlo" +
    \"mUYYBzlVQdDMZWnd6jSzsBRXSXxsNEy/RuXwplrA5epo7JdCoMkfeI4vUXm6MNr8"
    + "TQdFqIc1pdnOsbRf9ec62XbcfR7P8CDTsmLSaagx3rjenPdB+LTWKw2VUF+YONIg"
     + \"jM3fyFef9ovVhLhS5HvMqFGs8P75W+
    d7BOIbIu3EngACiEJOpYSsETD4WgPK6Iyv" + "
    j6cxsLeYMtElTb0fBIIqysCHdmjF3M1lqdqp4dKs3+W798GJZYM5MxZKUzrBi0Xu" +
    \"e7GtSh2aimsfQureUD+0z0RN2umeDsYcA1ghXMclDP+jLS1lnrv0Yvo+TKcm9b8G"
    + "uR/drbCrCsGyWFW+bsAu3AWz9S6TePurP5unRmNNvXpH5DRgsYl3d50CAwEAAQ=="
     + \"-----END RSA PUBLIC KEY-----\"
<!--NeedCopy-->
```

To add rewrite action for no padding policy expression by using the Citrix ADC command interface:

```
add rewrite action rsa_encrypt_act insertHttpHeader encrypted 'HTTP.REQ.
HEADER("plaintext").PKEY_ENCRYPT_PEM_NO_PADDING(rsa_pub_key_4096)
```

Policy-based RSA encryption with no padding option by using the GUI

The GUI enables you to complete the following tasks:

To add RSA public key for no padding operation as a policy expression by using the GUI:

1. Sign into the Citrix ADC appliance and navigate to **Configurations** > **AppExpert** > **Advanced Expressions**.
2. In the details pane, click **Add** to define an RSA public key as an advanced policy expression.
3. In Create Expression page, set the following parameters:
    a) Expression name. Name of the advanced expression.

b) Expression. Define RSA public key as an advanced expression using the Expression Editor. **Note**: The maximum string length is of 255 characters in a policy expression. For any key longer than 1024-bits, you have to break the key into smaller chunks and concatenate the chunks together as “chunk1” + “chunk2” + …
c) Comments. A brief description of the expression.

4. Click **Create**.

To add rewrite an action by using the GUI:

1. Sign into the Citrix ADC appliance and navigate to **Configurations** > **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, click **Add** to add a rewrite action.
3. In the **Create Rewrite Action** screen, set the following parameters:
   a) Name. Name of the rewrite action.
   b) Type. Select action type as INSERT_HTTP_HEADER.
   c) Use the action type to insert a header. Enter the name of the HTTP header that needs to be rewritten.
   d) Expression. Name of the advanced policy expression associated to the action.
   e) Comments. A brief description of the rewrite action.
4. Click **Create**.

Example 12: Configure rewrite to change the host name and URL in client request on Citrix ADC appliance

September 14, 2021

Rewrite feature on a Citrix ADC appliance is used to convert the URL available in the client request to another URL that the back end server can understand. You can achieve the following benefits by using the rewrite feature:

- Enhances the security by hiding the actual URL to the resource, which is requested by the client.
- Prevents the unauthorized user access from gaining access to the network resources.

Consider an example where your current organization is acquired by another organization. It becomes a difficult job for admins to inform about the new web address to every user of the acquired organization. In this scenario, using rewrite feature becomes convenient to change the host name and URL in the client requests for the website of the acquired organization. You can use rewrite to change the URLs in the client request temporarily when the website is under maintenance.

The following section describes the procedure to change the host name and URL in a client request using rewrite feature.

Consider an example where the user enters a `http://www.example.com` URL in the web browser. The website administrator wants the Citrix ADC appliance to convert the preceding URL in the client request as `http://myexample.example.net.in/resource/inventory/s?t=112`.

In the preceding example, the website administrator wants the Citrix ADC appliance to replace the "example.com" domain name with "myexample.example.net.in" and the URL with "resource/inventory/s?t=112".

Perform the following by using the CLI

1. Log on to the Citrix ADC appliance using SSH.
2. Add rewrite actions.
   - `add rewrite action rewrite_host_hdr_act replace "HTTP.REQ.HEADER(\"Host\")""\"myexample.example.net.in\""`
   - `add rewrite action rewrite_url_act replace HTTP.REQ.URL.PATH_AND_QUERY "\"/resource/inventory/s?t=112\""`
3. Add rewrite policies for the rewrite actions.
   - `add rewrite policy rewrite_host_hdr_pol "HTTP.REQ.HEADER(\"Host\").CONTAINS(\"www.example.com\")"rewrite_host_hdr_act`
   - `add rewrite policy rewrite_url_pol "HTTP.REQ.HEADER(\"Host\").CONTAINS(\"www.example.com\")"rewrite_url_act`
4. Bind the rewrite policies to a virtual server.
   - `bind lb vserver rewrite_LB -policyName rewrite_host_hdr_pol -priority 10 -gotoPriorityExpression 20 -type REQUEST`
   - `bind lb vserver rewrite_LB -policyName rewrite_url_pol -priority 20 -gotoPriorityExpression END -type REQUEST`

URL Transformation

September 14, 2021

The URL transformation feature provides a method for modifying all URLs in designated requests from an external version seen by outside users to an internal URL seen only by your Web servers and IT staff. You can redirect user requests seamlessly, without exposing your network structure to users. You can also modify complex internal URLs that users may find difficult to remember into simpler, more easily remembered external URLs.

> **Note**
>
> Before you can use the URL transformation feature, you must enable the Rewrite feature. To enable the Rewrite feature, see Enabling the Rewrite Feature.
>
> URL transform feature rewrites URLs in HTML response body and is not applied to JavaScript and other variables.

To begin configuring URL transformation, you create profiles, each describing a specific transformation. Within each profile, you create one or more actions that describe the transformation in detail. Next, you create policies, each of which identifies a type of HTTP request to transform, and you associate each policy with an appropriate profile. Finally, you globally bind each policy to put it into effect.

Configuring URL Transformation Profiles

September 14, 2021

A profile describes a specific URL transformation as a series of actions. The profile functions primarily as a container for the actions, determining the order in which the actions are performed. Most transformations transform an external hostname and optional path into a different, internal hostname and path. Most useful transformations are simple and require only a single action, but you can use multiple actions to perform complex transformations.

You cannot create actions and then add them to a profile. You must create the profile first, and then add actions to it. In the CLI, creating an action and configuring the action are separate steps. Creating a profile and configuring the profile are separate steps in both the CLI and the configuration utility.

To create a URL transformation profile by using the Citrix ADC command line

At the Citrix ADC command prompt, type the following commands, in the order shown, to create a URL transformation profile and verify the configuration. You can then repeat the second and third commands to configure additional actions:

- `add transform profile <profileName> -type URL [-onlyTransformAbsURLinBody (ON|OFF)] \[-comment <comment>]`
- `add transform action <name> <profileName> <priority>`
- `set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainFrom <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment "<string>"]`
- `show transform profile <name>`

Example:

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping
   .example.com' -reqUrlInto 'www.example.net/shopping' -resUrlFrom '
   www.example.net/shopping' -resUrlInto 'shopping.example.com' -
   cookieDomainFrom 'example.com' -cookieDomainInto 'example.net' -
   state ENABLED -comment 'URL transformation for shopping cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

To modify an existing URL transformation profile or action by using the Citrix ADC command line

At the Citrix ADC command prompt, type the following commands to modify an existing URL transformation profile or action and verify the configuration:

Note: Use a set transform profile or set transform action command, respectively. The set transform profile command takes the same arguments as does the add transform profile command, and set transform action is the same command that was used for initial configuration.

- `set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment "<string>"]`
- `show transform profile <name>`

Example:

```
> set transform action actshopping -priority 1000 -reqUrlFrom '
   searching.example.net' -reqUrlInto 'www.example.net/searching' -
   resUrlFrom 'www.example.net/searching' -resUrlInto 'searching.
   example.com' -cookieDomainInto 'example.net' -state ENABLED -comment
    'URL transformation for searching cart.'
```



```
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

To remove a URL transformation profile and actions by using the Citrix ADC command line

First remove all actions associated with that profile by typing the following command once for each action:

- rm transform action <name> After you have removed all actions associated with a profile, remove the profile as shown below.
- rm transform profile <name>

To create a URL transformation profile by using the configuration utility

1. In the navigation pane, expand **Rewrite**, expand URL Transformation, and then click **Profiles**.
2. In the details pane, click **Add**.
3. In the **Create URL Transformation Profile** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
   - Name*—name
   - Comment—comment
   - Only transform absolute URLs in response body—onlyTransformAbsURLinBody
4. Click **Create**, and then click **Close**. A message appears in the status bar, stating that the Profile has been configured successfully.

To configure a URL transformation profile and actions by using the configuration utility

1. In the navigation pane, expand **Rewrite**, expand URL Transformation, and then click **Profiles**.
2. In the details pane, select the profile you want to configure, and then click **Open**.
3. In the **Configure URL Transformation Profile** dialog box, do one of the following.
   - To create a new action, click **Add**.

- To modify an existing action, select the action, and then click **Open**.

4. Fill in the **Create URL Transformation Action** or **Modify URL Transformation Action** dialog box by typing or selecting values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
   - Action Name*—name
   - Comments—comment
   - Priority*—priority
   - Request URL from—reqUrlFrom
   - Request URL into—reqUrlInto
   - Response URL from—resUrlFrom
   - Response URL into—resUrlInto
   - Cookie Domain from—cookieDomainFrom
   - Cookie Domain into—cookieDomainInto
   - Enabled—state
5. Save your changes.
   - If you are creating a new action, click **Create**, and then **Close**.
   - If you are modifying an existing action, click **OK**.

     A message appears in the status bar, stating that the Profile has been configured successfully.
6. Repeat step 3 through step 5 to create or modify any additional actions.
7. To delete an action, select the action, and then click Remove. When prompted, click OK to confirm the deletion.
8. Click **OK** to save your changes and close the Modify URL Transformation Profile dialog box.
9. To delete a profile, in the details pane select the profile, and then click **Remove**. When prompted, click **OK** to confirm the deletion.

Configuring URL Transformation Policies

September 14, 2021

After you create a URL transformation profile, you next create a URL transformation policy to select the requests and responses that the Citrix ADC should transform by using the profile. URL transformation considers each request and the response to it as a single unit, so URL transformation policies are evaluated only when a request is received. If a policy matches, the Citrix ADC transforms both the request and the response.

Note: The URL transformation and rewrite features cannot both operate on the same HTTP header during request processing. Because of this, if you want to apply a URL transformation to a request,

you must make sure that none of the HTTP headers it will modify are manipulated by any rewrite action.

To configure a URL transformation policy by using the Citrix ADC command line

You must create a new policy. On the command line, an existing policy can only be removed. At the Citrix ADC command prompt, type the following commands to configure a URL transformation policy and verify the configuration:

- `<add transform policy <name> <rule> <profileName>`
- `<show transform policy <name>`

Example:

```
1  > add transform policy polsearch  HTTP.REQ.URL.SUFFIX.EQ("Searching")
       prosearching
2   Done
3  > show transform policy polsearch
4  1)      Name: polsearch
5          Rule: HTTP.REQ.URL.SUFFIX.EQ("Searching")
6          Profile: prosearching
7          Priority: 0
8          Hits: 0
9   Done
10 <!--NeedCopy-->
```

To remove a URL transformation policy by using the Citrix ADC command line

At the Citrix ADC command prompt, type the following command to remove a URL transformation policy:

`rm transform policy <name>`

Example:

```
1  > rm transform policy polsearch
2  Done
3  <!--NeedCopy-->
```

To configure a URL transformation policy by using the configuration utility

1. In the navigation pane, expand **Rewrite**, expand URL Transformation, and then click **Policies**.
2. In the details pane, do one of the following:

- To create a new policy, click **Add**.
- To modify an existing policy, select the policy, and then click **Open**.

3. In the **Create URL Transformation Policy** or **Configure URL Transformation Policy** dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation policies" as follows (asterisk indicates a required parameter):

   - Name*—name (Cannot be changed for a previously configured policy.)
   - Profile*—profileName
   - Expression—rule

   If you want help with creating an expression for a new policy, you can either hold down the Control key and press the space bar while your cursor is in the Expression text box. To create the expression, you can type it directly as described below, or you can use the Add Expression dialog box.

4. Click **Prefix**, and choose the prefix for your expression.

   Your choices are:

   - HTTP—The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - SYS—The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - CLIENT—The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - SERVER—The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
   - URL—The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.
   - TEXT—Any text string in the request. Choose this if you want to examine a text string in the request.
   - TARGET—The target of the request. Choose this if you want to examine some aspect of the request target.

   After you choose a prefix, the Citrix ADC displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom. The choices depend on which prefix you chose.

5. Select your next term.

If you chose HTTP as your prefix, your choices are REQ, which specifies HTTP requests, and RES, which specifies HTTP responses. If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you are certain which choice you want, double-click it to insert it into the Expression window.

1. Type a period, and then continue selecting terms from the list boxes that appear to the right of the previous list box. You type the appropriate text strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.
2. Click **Create** or **OK**, depending on whether you are creating a new policy or modifying an existing policy.
3. Click **Close**. A message appears in the status bar, stating that the Policy has been configured successfully.

To add an expression by using the Add Expression dialog box

1. In the **Create Responder Action** or **Configure Responder Action** dialog box, click **Add**.
2. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.
   - HTTP. The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - SYS. The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - CLIENT. The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - SERVER. The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
   - URL. The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.
   - TEXT. Any text string in the request. Choose this if you want to examine a text string in the request.
   - TARGET. The target of the request. Choose this if you want to examine some aspect of the request target.

     When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.

4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Globally Binding URL Transformation Policies

September 14, 2021

After you have configured your URL transformation policies, you bind them to Global or a bind point to put them into effect. After binding, any a request or response that matches a URL transformation policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the Citrix ADC OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the URL transformation feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you tell the Citrix ADC to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you tell the Citrix ADC to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you globally bind your policies.

Note: URL transformation policies cannot be bound to TCP-based virtual servers.

To bind a URL transformation policy by using the Citrix ADC command line

At the Citrix ADC command prompt, type the following commands to globally bind a URL transformation policy and verify the configuration:

- `bind transform global <policyName> <priority>`
- `show transform global`

Example:

```
1 > bind transform global polisearching 100
2  Done
3 > show transform global
4 1)      Policy Name: polisearching
5         Priority: 100
6
```



```
7   Done
8 <!--NeedCopy-->
```

To bind a URL transformation policy by using the configuration utility

1. In the navigation pane, expand Rewrite, then expand URL Transformation, and then click **Policies.
2. In the details pane, click **Policy Manager**.
3. In the **Transform Policy Manager** dialog box, choose the bind point to which you want to bind the policy**. The choices are:
   - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the Citrix ADC appliance, and are applied before any other policies.
   - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
   - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
   - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the Citrix ADC appliance.
   - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
4. Select Insert Policy to insert a new row and display a drop-down list with all available, unbound URL transformation policies.
5. Select the policy you want to bind, or select New Policy to create a new policy. The policy that you selected or created is inserted into the list of globally bound URL transformation policies.
6. Make any additional adjustments to the binding.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Transform Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression.
7. Repeat steps 3 through 6 to add any additional URL transformation policies you want to globally

bind.

8. Click **OK** to save your changes. A message appears in the status bar, stating that the Policy has been configured successfully.

RADIUS support for the rewrite feature

September 14, 2021

The Citrix ADC expressions language includes expressions that can extract information from and manipulate RADIUS messages in requests and responses. These expressions enable you to use the rewrite feature to modify portions of a RADIUS message before sending it to its destination. Your rewrite policies and actions can use any expression that is appropriate or relevant to a RADIUS message. The available expressions enable you to identify the RADIUS message type, extract any attribute-value pair (AVP) from the connection, and modify RADIUS AVPs. You can also create policy labels for RADIUS connections.

You can use the new RADIUS expressions in Rewrite rules for a number of purposes. For example, you could:

- Remove the domain\ portion of the RADIUS user-name AVP to simplify single sign-on (SSO).
- Insert a vendor-specific AVP, such as the MSISDN field used in telephone company operations to contain subscriber information.

You can also create policy labels to route specific types of RADIUS requests through a series of policies that are appropriate to those requests.

> **Note**:
>
> RADIUS for Rewrite has the following limitations:

- The Citrix ADC does not re-sign rewritten RADIUS requests or responses. If the RADIUS authentication server requires signed RADIUS messages, authentication will fail.
- The currently available RADIUS expressions do not work with RADIUS IPv6 attributes.

The Citrix ADC documentation for expressions that support RADIUS assumes familiarity with the basic structure and purpose of RADIUS communications. If you need more information about RADIUS, see your RADIUS server documentation or search online for an introduction to the RADIUS protocol.

Configuring Rewrite Policies for RADIUS

The following procedure uses the Citrix ADC command line to configure a rewrite action and policy and bind the policy to a rewrite-specific global bind point.

To configure a Rewrite action and policy, and bind the policy:

At the command prompt, type the following commands:

- `add rewrite action <actName> <actType>`
- `add rewrite policy <polName> <rule> <actName>`
- `bind rewrite policy <polName> <priority> <nextExpr> -type <bindPoint>`
  where `<bindPoint>` represents one of the rewrite-specific global bind points.

RADIUS Expressions for Rewrite

In a rewrite configuration, you can use the following Citrix ADC expressions to refer to various portions of a RADIUS request or response.

Identifying the Type of Connection:

- `RADIUS.IS_CLIENT`

  Returns TRUE if the connection is a RADIUS client (request) message.

- `RADIUS.IS_SERVER`

  Returns TRUE if the connection is a RADIUS server (response) message.

Request Expressions:

- `RADIUS.REQ.CODE`

  Returns the number that corresponds to the RADIUS request type. A derivative of the num_at class. For example, a RADIUS access request would return 1 (one). A RADIUS accounting request would return 4.

- `RADIUS.REQ.LENGTH`

  Returns the length of the RADIUS request, including the header. A derivative of the num_at class.

- `RADIUS.REQ.IDENTIFIER`

  Returns the RADIUS request identifier, a number assigned to each request that allows the request to be matched to the corresponding response. A derivative of the num_at class.

- `RADIUS.REQ.AVP(<AVP Code No>).VALUE`

  Returns the value of first occurrence of this AVP as a string of type text_t.

- `RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)`

  Returns the specified instance of the AVP as a string of type RAVP_t. A specific RADIUS AVP can occur multiple times in a RADIUS message. INSTANCE (0) returns the first instance, INSTANCE (1) returns second instance, and so on, up to sixteen instances.

- `RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)`

  Returns the value of specified instance of the AVP as a string of type text_t.

- `RADIUS.REQ.AVP(<AVP code no>).COUNT`

  Returns the number of instances of a specific AVP in a RADIUS connection, as an integer.

- `RADIUS.REQ.AVP(<AVP code no>).EXISTS`

  Returns TRUE if the specified type of AVP exists in the message, or FALSE if it does not.

Response Expressions:

RADIUS response expressions are identical to RADIUS request expressions, except that RES replaces REQ.

Typecasts of AVP Values:

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other Citrix ADC typecast expressions.

Example:

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other Citrix ADC typecast expressions.

```
1 RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
2 <!--NeedCopy-->
```

AVP Type Expressions:

The Citrix ADC supports expressions to extract RADIUS AVP values by using the assigned integer codes described in RFC2865 and RFC2866. You can also use text aliases to accomplish the same task. Some examples follow.

- `RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value`

  Extracts the RADIUS user-name value.

- `RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT\\_SESSION\\_ID.value`

  Extracts the Acct-Session-ID AVP (code 44) from the message.

- `RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR\\_SPECIFIC.VALUE`

  Extracts the vendor-specific value.

The values of most commonly-used RADIUS AVPs can be extracted in the same manner.

RADIUS Bind Points:

Four global bind points are available for policies that contain RADIUS expressions.

- `RADIUS_REQ_OVERRIDE`

  Priority/override request policy queue.

- `RADIUS_REQ_DEFAULT`

  Standard request policy queue.

- `RADIUS_RES_OVERRIDE`

  Priority/override response policy queue.

- `RADIUS_RES_DEFAULT`

  Standard response policy queue.

RADIUS Rewrite-Specific Expressions:

- `RADIUS.NEW_AVP`

  Returns the specified RADIUS AVP as a string.

- `RADIUS.NEW_AVP_INTEGER32`

  Returns the specified RADIUS AVP as an integer.

- `RADIUS.NEW_AVP_UNSIGNED32`

  Returns the specified RADIUS AVP as an unsigned integer.

- `RADIUS.NEW_VENDOR_SPEC_AVP(<ID>, <definition>)`

  Adds the specified extended vendor specific AVPs to the connection. For `<ID>`, substitute a long number. For `<definition>`, substitute a string that contains the data for the AVP.

- `RADIUS.REQ.AVP_START`

  Returns the location between the end of the RADIUS header and the start of the AVPs. Used in rewrite actions.

  Example:

```
1     add rewrite action insert1 insert_after radius.req.avp_start radius
          .new_avp(33, "NEW AVP")
2  <!--NeedCopy-->
```

- `RADIUS.REQ.AVP_END`

  Returns the location at the end of radius message (or in other words end of all AVPs) in radius message. Used when performing rewrite actions.

Example:

```
1     add rewrite action insert2 insert_before radius.req.avp_end "radius
         .new_avp(33, \"NEW AVP\")"
2  <!--NeedCopy-->
```

- `RADIUS.REQ.AVP_LIST`

  Returns the location at the start of the AVPs in a RADIUS message, and the length of the RADIUS message, excluding the header. In other words, returns all AVPs in a RADIUS message. Used to perform Rewrite actions.

  Example:

```
1     add rewrite action insert3 insert_before_all radius.req.avp_list "
         radius.new_avp(33, \"NEW AVP\")" -search "avp(33)"
2  <!--NeedCopy-->
```

Valid Rewrite-Action Types for RADIUS:

The Rewrite action types that can be used with RADIUS expressions are:

- INSERT_AFTER
- INSERT_BEFORE
- INSERT_AFTER_ALL
- INSERT_BEFORE_ALL
- DELETE
- DELETE_ALL
- REPLACE
- REPLACE_ALL

All `INSERT_ actions` can be used to insert a RADIUS AVP into a RADIUS connection.

Use Cases

Following are use cases for RADIUS with rewrite.

Rewriting the User-Name AVP

To configure the rewrite feature to remove the Domain\ string from the RADIUS user-name AVP, begin by creating a rewrite REPLACE action as shown in the example below. Use the action in a Rewrite policy that selects all RADIUS requests. Bind the policy to a global bind point. When you do so, set the priority the appropriate level to allow any block or reject policies to take effect first, but ensure that all requests that are not blocked or rejected are rewritten. Set the Goto Expression (gotoPriorityExpr) to NEXT to continue policy evaluation, and attach the policy to the RADIUS_REQ_DEFAULT queue.

Example:

```
add rewrite action rwActRadiusDomainDel replace radius.req.user_name q/
   RADIUS.NEW_AVP(1,RADIUS.REQ.USER_NAME.VALUE.AFTER_STR(" "))/
add rewrite policy RadiusRemoveDomainPol true rwActRadiusDomainDel
<!--NeedCopy-->
```

> **Note**:
>
> The rewrite policy for RADIUS is not applicable to a gateway virtual server. If a gateway virtual server is used a load balancing then RADIUS needs to be configured and the rewrite policy needs to be bound to a RADIUS load balancing virtual server.

Inserting a Vendor-Specific AVP

To configure Rewrite action to insert a Vendor-Specific AVP containing the contents of the MSISDN field, begin by creating a rewrite INSERT action that inserts the MSISDN field into the request. Use the action in a Rewrite policy that selects all RADIUS requests. bind the policy to global, setting the priority to an appropriate level and the other parameters as shown in the following example.

Example:

```
add rewrite action rwActRadiusInsMSISDN insert_after radius.req.
   avp_start RADIUS.NEW_VENDOR_SPEC_AVP(<VENDOR ID>, "RADIUS.NEW_AVP(<
   Attribute Code>, <MSISDN>)")
add rewrite policy rwPolRadiusInsMSISDN true rwActRadiusInsMSISDN
bind rewrite global rwPolRadiusInsMSISDN 100 NEXT -type
   RADIUS_REQ_DEFAULT
<!--NeedCopy-->
```

Diameter Support for Rewrite

September 14, 2021

The Rewrite feature now supports the Diameter protocol. You can configure Rewrite to modify Diameter requests and response as you would HTTP or TCP requests and responses, allowing you to use Rewrite to manage the flow of Diameter requests and make necessary modifications. For example, if the “Origin-Host” value in a Diameter request is inappropriate, you can use Rewrite to replace it with a value that is acceptable to the Diameter server.

To configure Rewrite to modify a Diameter request

To configure the Rewrite feature to replace the Origin-Host in a diameter request with a different value, at the command prompt, type the following commands:

- <add rewrite action <actname> replace “DIAMETER.REQ.AVP(264,\“Citrix ADC.example.net\”)”
  For <actname>, substitute a name for your new action. The name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For Citrix ADC.example.net, substitute the Host-Origin that you want to use instead of the original Host-Name.
- add rewrite policy <polname> “diameter.req.avp(264).value.eq(\“host.example.com\”)” <actname>
  For <polname>, substitute a name for your new policy. As with <actname>, the name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For host.example.com, substitute the name of the Host-Origin that you want to change. For <actname>, substitute the name of the action that you just created.
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  For <vservername>, substitute the name of the load balancing virtual server to which you want to bind the policy. For <polname>, substitute the name of the policy you just created. For <priority>, substitute a priority for the policy.

Example:

To create a Rewrite action and policy to modify all Diameter Host-Origins of “host.example.com” to “Citrix ADC.example.net”, you could add the following action and policy, and bind the policy as shown.

```
1 > add rewrite action rw_act_replace_avp replace "diameter.req.avp(264)"
     "diameter.new.avp(264,\"Citrix ADC.example.net\")"
2 > add rewrite policy rw_diam_pol "diameter.req.avp(264).value.eq(\"
    client.realm2.net\")" rw_act_replace_avp
3 > bind lb vserver vs1 -policyName rw_diam_pol -priority 10 -type
    REQUEST
4
5 Done
6 <!--NeedCopy-->
```

DNS Support for the Rewrite Feature

September 14, 2021

You can configure the rewrite feature to modify DNS requests and responses, as you would for HTTP

or TCP requests and responses. You can use rewrite to manage the flow of DNS requests, and make necessary modifications in the header, or in the answer section. For example, if the DNS response does not have the AA bit set in the header flag, you can use rewrite to set the AA bit in the DNS response and send it to the client.

DNS Expressions

In a rewrite configuration, you can use the following Citrix ADC expressions to refer to various portions of a DNS request or response:

See Expressions and Descriptions

DNS Bind Points

The following global bind points are available for policies that contain DNS expressions.

Bind Points	Description
DNS_REQ_OVERRIDE	Override request policy queue.
DNS_REQ_DEFAULT	Standard request policy queue.
DNS_RES_OVERRIDE	Override response policy queue.
DNS_RES_DEFAULT	Standard response policy queue.

In addition to the default bind points, you can create policy labels of type DNS_REQ or DNS_RES and bind DNS policies to them.

Rewrite Action Types for DNS

- **replace_dns_answer_section**—This action replaces the DNS answers section with the defined expression in the DNS policy.
- **replace_dns_header_field**—Checks the opcode type in the DNS request. Returns True or False, indicating whether the opcode type in the DNS request matches the specified opcode type. This action replaces the DNS header section with the defined expression in the DNS policy.

Configuring Rewrite Policies for DNS

The following procedure uses the Citrix ADC command line to configure a rewrite action and policy and bind the policy to a rewrite-specific global bind point.

Configure Rewrite action and policy, and bind the policy for DNS

At the command prompt, type the following commands:

1. `add rewrite action <actName> <actType>`

   For `<actname>`, substitute a name for your new action. The name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For `<actType>`, specify the rewrite action types provided for DNS expressions.

2. `add rewrite policy <polName> <rule> <actName>`

   For `<polname>`, substitute a name for your new policy. For `<actname>`, the name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For `<actname>`, substitute the name of the action that you just created.

3. `bind rewrite global <polName> <priority> <gotoPriorityExpression> -type <bindPoint>`

   For `<polName>`, substitute the name of the policy that you just created. For `<priority>`, specify the priority of the policy. For `<bindPoint>`, substitute one of the rewrite -specific global bind points.

Example:

Set the AA bit of DNS request to load balance virtual server.

The following commands configure the Citrix ADC appliance to act as an authoritative DNS server for all the queries that it serves.

```
1 add rewrite action set_aa replace_dns_header_field dns.req.header.flags
     .set(aa)
2 add rewrite policy pol !dns.req.header.flags.is_set(aa)  set_aa
3 bind rewrite global  pol  100  -type dns_res_override
4 <!--NeedCopy-->
```

Modify the response answer and header section.

If the server responds with an NX domain, you can set the rewrite action to replace the response with specified IP address. A NOPOLICY-REWRITE enables you to invoke an external bank without processing an expression (a rule). This entry is a dummy policy that does not contain a rule but directs the entry to a policy label or virtual server specific policy banks.

```
1 add rewrite action set_aa_res replace_dns_header_field "dns.res.header.
     flags.set(aa)"
2 add rewrite action modify_nxdomain_res replace_dns_answer_section "dns.
     new_rrset_a(\"10.102.218.160\",300)"
3 add rewrite policy set_res_aa true set_aa_res
```



```
4 add add rewrite policy modify_answer "dns.RES.HEADER.RCODE.EQ(nxdomain)
      && dns.RES.QUESTION.TYPE.EQ(A)"
5 modify_nxdomain_res
6 add rewrite policylabel MODIFY_NODATA dns_res
7 bind rewrite policylabel MODIFY_NODATA modify_answer 10 END
8 bind rewrite policylabel MODIFY_NODATA set_res_aa 11 END
9 bind lb vserver v1 -policyName NOPOLICY-REWRITE -priority 11 -
     gotoPriorityExpression END -type
10 RESPONSE -invoke policylabel MODIFY_NODATA
11 <!--NeedCopy-->
```

Limitations:

- Rewrite policies are evaluated only if the Citrix ADC appliance is configured as a DNS proxy server and there is a cache miss.
- If the Recursion Available (RA) flag in the header is set to YES, the RA flag will not be modified in the rewrites.
- If the RA flag in the header is set to YES, the CD flag in the header is modified regardless of any rewrite action.

String maps

September 14, 2021

You can use string maps to perform pattern matching in all Citrix ADC features that use the default policy syntax. A string map is a Citrix ADC entity that consists of key-value pairs. The keys and values are strings in either ASCII or UTF-8 format. String comparison uses two new functions, `MAP_STRING (<string_map_name>)` and `IS_STRINGMAP_KEY(<string_map_name>)`.

A policy configuration that uses string maps performs better than one that does string matching through policy expressions, and you need fewer policies to perform string matching with a large number of key-value pairs. String maps are also intuitive, simple to configure, and result in a smaller configuration.

How String Maps Work

String maps are similar in structure to pattern sets (a pattern set defines a mapping of index values to strings; a string map defines a mapping of strings to strings) and the configuration commands for string maps (commands such as add, bind, unbind, remove, and show) are syntactically similar to configuration commands for pattern sets. Also, as with index values in a pattern set, each key

in a string map must be unique across the map. The following table illustrates a string map called url_string_map, which contains URLs as keys and values.

Key	Value
/url_1.html	http://www.redirect_url_1.com/url_1.html
/url_2.html	http://www.redirect_url_2.com/url_2.html
/url_3.html	http://www.redirect_url_1.com/url_1.html

Table 1. String Map “url_string_map”

The following table describes the two functions that have been introduced to enable string matching with keys in a string map. String matching is always performed with the keys. Additionally, the following functions perform a comparison between the keys in the string map and the complete string that is returned by the expression prefix. The examples in the descriptions refer to the preceding example.

For completed information about the two functions introduced for enabling string matching with keys in a string map, see String Map Function table pdf.

Configuring a String Map

You first create a string map and then bind key-value pairs to it. You can create a string map from the command line interface (CLI) or the configuration utility.

To configure a string map by using the command line interface

At the command prompt, do the following:

1. Create a string map.

```
add policy stringmap <name> -comment <string>
```

1. Bind a key-value pair to the string map.

```
bind policy stringmap <name> <key> <value> [-comment <string>]
```

Example:

```
1 bind policy stringmap url_string_map1 "/url_1.html" "http://www.
     redirect_url_1.com/url_1.html"
2 <!--NeedCopy-->
```


To configure a string map by using the Citrix ADC GUI

Navigate to **AppExpert** > **String Maps**, click **Add** and specify the relevant details.

Example: responder policy with a redirect action

The following use case involves a responder policy with a redirect action. In the example below, the first four commands create the string map url_string_map and bind the three key-value pairs used in the earlier example. After creating the map and binding the key-value pairs, you create a responder action (act_url_redirects) that redirects the client to the corresponding URL in the string map or to www.default.com. You also configure a responder policy (pol_url_redirects) that checks whether requested URLs match any of the keys in url_string_map and then performs the configured action. Finally, you bind the responder policy to the content switching virtual server that receives the client requests that are to be evaluated.

```
add stringmap url_string_map

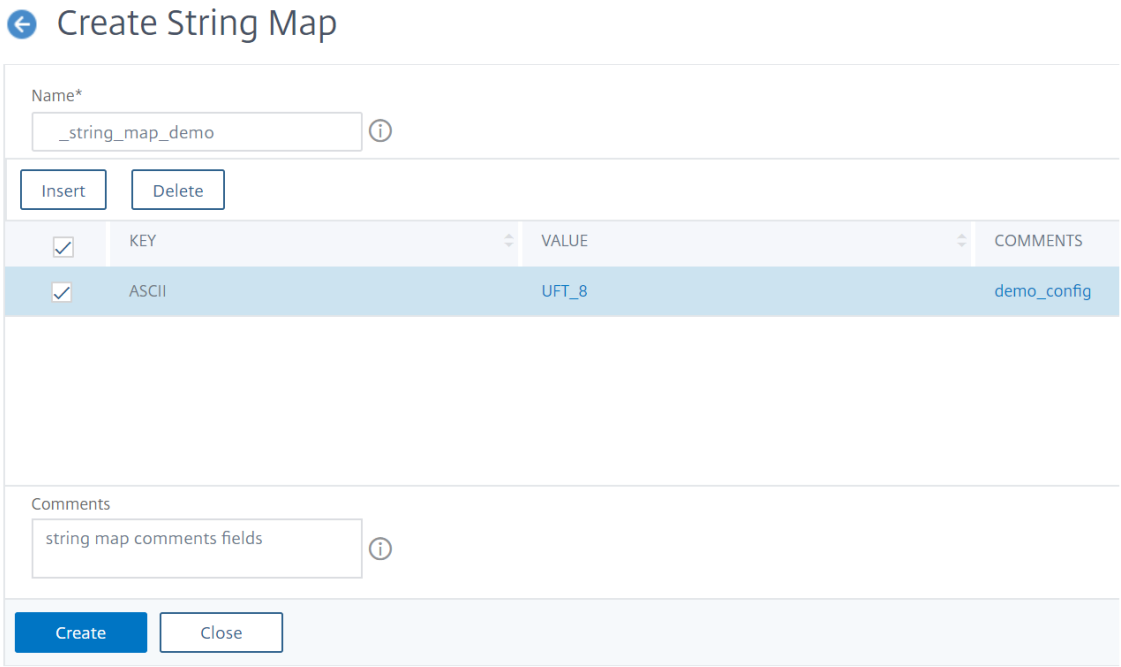
bind stringmap url_string_map /url_1.html http://www.redirect_url_1.com/
url_1.html

bind stringmap url_string_map /url_2.html http://www.redirect_url_2.com/
url_2.html

bind stringmap url_string_map /url_3.html http://www.redirect_url_1.com/
url_1.html

add responder action act_url_redirects redirect 'HTTP.REQ.URL.MAP_STRING("
url_string_map")ALT "www.default.com"'-bypassSafetyCheck yes

add responder policy pol_url_redirects TRUE act_url_redirects

bind cs vserver csw_redirect -policyname pol_url_redirects -priority 1 -
type request
```

To configure a string map by using the Citrix ADC GUI

Follow the procedure given below to configure a string map.

1. On the navigation pane, expand **AppExpert** and click **String Maps**.
2. On the details pane, click **Add**.
3. In the **Create String Map** page, set the following parameters:
    - Name. Name of the string map.
    - Configure key value. ASCII based key value entry bound to the string map
    - Comments. A short description about the key values bound to the string map.

4. Click **Create** and **Close**.



URL Sets

September 14, 2021

This feature enables you to blacklist one million URLs. The section includes the following topics:

- Getting Started
- Using Advanced Policy Expressions for URL Evaluation
- Configuring a URL Set
- URL Pattern Semantics
- Blacklisted URL Categories

Getting Started

September 14, 2021

To prevent access to restricted websites, a Citrix ADC appliance uses a specialized URL matching algorithm. The algorithm uses a URL set that can contain a list of URLs up to 1 million (1,000,000) blacklisted entries. Each entry can include metadata that defines URL categories and category groups as indexed patterns. The appliance can also periodically download URLs of highly sensitive URL sets

managed by internet enforcement agencies (with government websites) or internet organizations. Once the URL set is downloaded from a website and imported into the appliance, the appliance encrypts the URL sets (as required by these agencies) and they are kept confidential and the entries are not tampered.

The Citrix ADC appliance uses advanced policies to determine whether an incoming URL must be blocked, allowed, or redirected. These policies use advanced expressions to evaluate incoming URLs against blacklisted entries. An entry can include metadata. For entries that have no metadata, you might want to use an expression that evaluates the URL based on an exact string match. For other URLs, you might want to use an expression that evaluates the URL's metadata, in addition to an expression that checks for an exact string match.

Use Case for Safe Internet Access Policies for ISPs/Telcos

A URL set enables an ISP (ISP) or a Telco customer to enforce government mandated safe internet access policies such as:

1. Block access to illegal internet sites (child abuse, drugs, and so on)
2. Safe browsing for children

A Citrix ADC appliance enables you to periodically download URL sets managed by internet enforcement agencies or independent internet organizations. The appliance periodically downloads the list and updates it securely. The list is stored as confidential URL sets so that it is not tampered or human readable. The periodically downloaded URL set functions as a blacklisted set for URL evaluation purposes.

If you have a private URL set and the contents of the list are kept confidential and the network administrator does not know about the blacklisted URLs present in the list. To make sure the policy is configured correctly and the correct list is referenced, you must configure the Canary URL and add it to the URL set. Using the Canary URL, the administrator can request through the appliance uses the private URL set to ensure it is looked up for every URL request.

Advanced Policy Expressions for URL Evaluation

September 14, 2021

The following table describes the expressions you can use to evaluate incoming URLs with entries in an URL set.

Note: HTTP.REQ.URL is generalized to be used as <URL expression>

Expression	Operation
<URL expression>.URLSET_MATCHES_ANY	Evaluates to TRUE if the URL exactly matches any entry in the URL set.
<URL expression> .GET_URLSET_METADATA(<URLSET>)	The GET_URLSET_METADATA() expression returns the associated metadata if the URL exactly matches any pattern within the URL set. An empty string is returned if there was no match.
<URL expression> .GET_URLSET_METADATA(<URLSET>).EQ(<METADATA>	Evaluates to TRUE if the matched metadata is equal to <METADATA>.
<URL expression> .GET_URLSET_METADATA(<URLSET>).TYPECAST_LIST_T(‘,’).GET(0).EQ(<CATEGORY>)	Evaluates to TRUE if the matched metadata is at the beginning of the category. This pattern can be used to encode separate fields within metadata, but match only the 1st field.
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	Joins the host and URL parameters, which can then be used as a <URL expression> for matching.

Configuring URL Set

September 14, 2021

You can perform the following tasks to configure a URL set and restrict URLs on a Citrix ADC platform:

1. Import a URL set (download and encrypt it). Importing a URL set in a Citrix ADC appliance allows you:

- To download the URL file.
- To add the file to the appliance.
- To encrypt the file.
  Until you add the URL set to the system, it is not visible to the user.

You can download a set in the following ways:

- Download a URL set once from a remote server and specify it as `http://myserver.com/file_with_urlset.csv`
- add a file under the `/var/tmp/` path inside ADC and use the command, as in the example:


```
> shell cat /var/tmp/test_urlset.csv
example.com
google.com
> import policy urlset top10
k -url local:test_urlset.csv -delimiter "," -rowSeparator "n" -interval
    10 -privateSet -canaryUrl http://www.in.gr
Done

<!--NeedCopy-->
```

The imported URL set is further categorized into different categories and category groups in the database. This is valid only if categories exist in the metadata of the URL set file.

Note: There can be a chance that you might have URL patterns without metadata.

Once you have imported the file, you can update, delete, or display file properties. After the file is pushed into the appliance, you can modify the entries by more adding rows.

The imported set is then stored in an encrypted file format on the Citrix ADC directory. The imported list contains millions of URL entries. To the following 'The imported list can contain up to 1 million URL entries. Otherwise, the appliance returns an error message saying that the value exceeds the limit. If the imported URL set has blacklisted entries with metadata, the metadata it is detected by the appliance when it is imported.

Once you import a URL set and add it into the appliance, the URL set is available for advanced policies to identify the correct URL set during incoming URL evaluation. `HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URLSET_MATCHES_ANY(<URL set name>)`

1. Updating a URL set on the Citrix ADC appliance. Once you have pushed the file into the appliance, at this interval you can manually update a URL file by using command line interface.
2. Exporting a URL set. If you prefer a backup of the URL set, you can export the list of URL patterns and save a copy of it to a destination URL. Before you export, check whether the URL set is marked as private. If is marked private, the URL set cannot be exported. Export functionality does not work with private set. So a new url set `myurl` would be imported without private set defined, and then it would be exported to another file in a local path, as below:

```
> shell touch /var/tmp/test_urlset_export.csv
Done
> shell cat /var/tmp/test_urlset_export.csv
Done
> shell cat /var/tmp/test_urlset.csv
example.com
google.com
Done
```



```
> export urlset myurl -url local:test_urlset_export.csv

> import urlset myurl -url local:test_urlset.csv
Done
(a non-private urlset is imported)

<!--NeedCopy-->
```

1. Removing a URL set. If you want to delete a URL set of blacklisted entries, you can use the remove command to delete the URL set from the Citrix ADC appliance.
2. Displaying a URL set. You can display the properties of a URL set by using the show command.

> **Note:** URLs with query part are removed during import.

Example:

```
show urlset
Name: top100 PatternCount: 100 Delimiter:  RowSeparator: Interval: 0
Done
<!--NeedCopy-->
```

Import a URL set with meta by using the command line interface

At command prompt, type:

```
import urlset <name> [-overwrite] [ - delimiter <character>] [-
    rowSeparator <character>] [-url] <url> [-interval <seconds>] [-
    privateSet] [-canaryUrl <URL>]
<!--NeedCopy-->
```

Where,

Delimiter is a CSV file record with default value set as 44.

rowSeparator is a CSV file row separator with default value set as 10.

Interval is the time interval in secs, rounded to the nearest 15 minutes at which the update of the url set occurs.

CanaryURL is a URL used for testing when the contents of the url set is kept confidential.

Example

```
import policy urlset -url local:test_urlset.csv -delimiter ","-rowSeparator
 "n"-interval 10 -privateSet -canaryUrl http://www.in.gr
```


Perform explicit subdomain match for an imported URL set

You can now perform an explicit subdomain match for an imported URL set. A new parameter, "subdomainExactMatch" is added to the "import policy URLset" command. When you enable the parameter, the URL Filtering algorithm performs an explicit subdomain match. For example, if the incoming URL is "news.example.com" and if the entry in the URL set is "example.com," the algorithm does not match the URLs.

At the command prompt, type:

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-rowSeparator
 <character>] -url [-interval <secs>] [-privateSet][-subdomainExactMatch]
[-canaryUrl <URL>]
```

Example:

```
import policy urlset forth_urlset -url local:test_urlset.csv -interval 3600
 -subdomainExactMatch
```

To show the URL set by using the command line interface

At the command prompt, type:

```
show urlset <name>
```

Example:

At the command prompt, type:

```
1          URLset      Count
2          ------      -----
3  1)      top1k       100
4   Done
5
6  > show urlset top1k
7          Count       Delimiter   Interval    RowSeparator
8          -----       ---------   --------    ------------
9          100               ,         0          0x0a
10  Done
11 >
12
13 <!--NeedCopy-->
```

To show the URL set imported by using the command line interface

At the command prompt, type:


```
show urlset -imported
```

Example:

At the command prompt, type:

```
1          URLset
2          ------
3  1)      top1k
4   Done
5  <!--NeedCopy-->
```

To show URL set by using the command line interface

At the command prompt, type:

```
show urlset <name>
```

To export a URL set by using the command line interface

At the command prompt, type:

```
export urlset <name> <url>
```

To add a URL set by using the command line interface

At the command prompt, type:

```
add urlset <urlset_name>
```

To update a URL set by using the command line interface

At the command prompt, type:

```
update urlset <name>
```

To remove a URL set command by using the command line interface

At the command prompt, type:

```
remove urlset <name>
```

Example:

> Note:
>
> Before you import or export a URLset, you must make sure the `test_urlset_export.csv` and `test_urlset.csv` files are created and available under the `/var/tmp` directory.

```
import policy urlset -url local:test_urlset.csv -delimiter "," -
   rowSeparator "n" -interval 10 -privateSet -overwrite  - canaryUrl
   http://www.in.gr

add policy urlset top10k

update policy urlset top10k

sh policy urlset

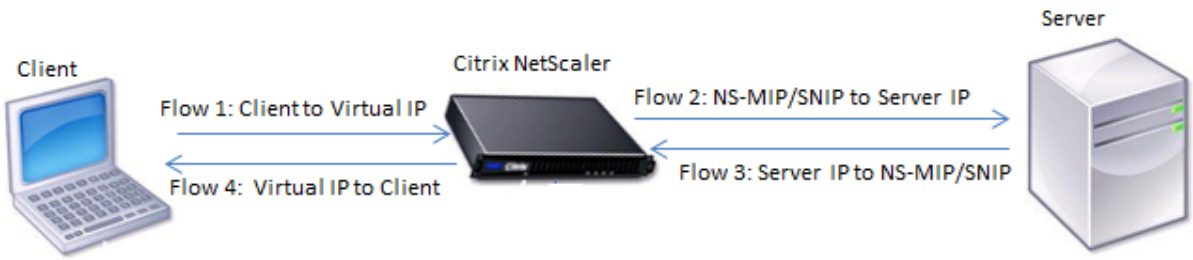
sh policy urlset top10k

export policy urlset urlset1 -url local:test_urlset_export.csv

import policy urlset top10k -url local:test_urlset.csv  - privateSet

add policy urlset top10k

update policy urlset top10k

show policy urlset top10k
<!--NeedCopy-->
```

Display imported URL sets

You can now display imported URL sets in addition to added URL sets. To do this, a new parameter "imported" is added to the "show url set" command. If you enable this option, the appliance displays all imported URL sets and distinguishes the imported URL sets from the added URL sets.

At the command prompt, type:

```
show policy urlset [<name>] [-imported]
```

Example:

```
show policy urlset -imported
```

To import a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets**, click **Import** to download the URL set.

To add a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets**, click **Add** to create a URL set file for the downloaded URL set.

To edit a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets**, select a URL set and click **Edit** to modify.

To Update a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets,** select a URL set and click **Update URL Set** to update the URL set with the latest modifications made to the file.

To Export a URL set by using the GUI

Navigate to **AppExpert > URL Sets**, select a URL set, and click **Export URL Set** to export the URL patterns in a set to a destination URL and save it in that location.

URL Pattern Semantics

September 14, 2021

The following table shows the URL patterns used for specifying the list of pages you to want to filter. For example, the URL pattern, `http://www.example.com/bar` matches a single page `http://www.example.com/bar`. To cover all the pages where the URL starts with www.example.com/bar, you must explicitly add a '*' at the end.

For more information, see URL pattern metadata mapping table.

URL Categories

September 14, 2021

Following is a list of blacklisted categories.

S.no	Blacklisted Categories
1	Illegal Activities
2	Illegal Drugs

S.no	Blacklisted Categories
3	Medication
4	Marijuana
5	Terrorism/Extremists
6	Weapons
7	Hate/Slander
8	Violence/Suicide
9	Advocacy in general
10	Adult/Porn
11	Nudity
12	Sexual Services
13	Adult Search/Links
14	Hacking/Cracking
15	Malware
16	Remote Proxies
17	Search Engine Caches
18	Translators
19	Dating
20	Weddings/Matrimony
21	Market Rates
22	Online Trading
23	Insurance
24	Financial Products
25	Gambling in general
26	Lottery
27	Online games
28	Games
29	Auctions
30	Shopping/Retail
31	Real Estate

S.no	Blacklisted Categories
32	IT Online Shopping
33	Web based Chat
34	Instant Messages
35	Web based Mail
36	E-Mail Subscriptions
37	Bulletin Boards
38	IT Bulletin Boards
39	Personal Web Pages/Blogs
40	Downloads
41	Program Downloads
42	Storage Services
43	Streaming Media
44	Employment
45	Career Advancement
46	Side Business
47	Grotesque
48	Special Events
49	Popular Topics
50	Adult Magazine/News
51	Smoking
52	Drinking
53	Alcoholic Products
54	Fetish
55	Sexual Expression(text)
56	Costume Play/Enjoyment
57	Occult
58	Home & Family
59	Professional Sports
60	Sports in general

S.no	Blacklisted Categories
61	Life Events
62	Travel & Tourism
63	Public Agency Tourism
64	Public Transit
65	Accommodations
66	Music
67	Horoscope/Astrology/Fortune Telling
68	Entertainer/Celebrity
69	Dining/Gourmet
70	Entertainment/Venues/Activities
71	Traditional Religions
72	Religions
73	Politics
74	Advertisements/Banners
75	Sweepstakes/Prizes
76	SPAM
77	News
78	Automotive
79	Business & Commercial
80	Computing & Internet
81	Education
82	Government
83	Health
84	Internet Telephony
85	Military
86	Peer to Peer/Torrents
87	Recreation & Hobbies
88	Reference
89	Search Engines & Portals

S.no	Blacklisted Categories
90	Sex Education
91	SMS & Mobile Telephony Services
92	Mobile Apps & Publishers
93	Spyware
94	Content Delivery Networks & Infrastructure
95	Kids Sites
96	Swimsuits & Lingerie
97	Arts & Cultural Events
98	Hosting Sites
99	Philanthropy & Non-Profit Organizations
100	Photo Search & Photo Sharing Sites
101	Ringtones
102	Fashion & Beauty
103	Mobile App Stores
104	Parked Domains
105	Emoticons
106	Mobile Operators
107	Botnets
108	Infected Sites
109	Phishing Sites
110	Keyloggers
111	Mobile Malware
112	No Content
113	Agriculture
114	Architecture
115	Associations/Trade Groups/Unions
116	Books/eBooks
117	BOT Phone Home
118	DDNS

S.no	Blacklisted Categories
119	Unsupported URL
120	Law
121	Local Communities
122	Miscellaneous
123	Online Magazines
124	Pets/Veterinarian
125	Piracy & Copyright Theft
126	Private IP Addresses
127	Recycling/Environment
128	Science
129	Society & Culture
130	Transport Services & Freight
131	Photography & Film
132	Museums & History
133	eLearning
134	Social Networks in General
135	Facebook
136	Facebook: Posting
137	Facebook: Commenting
138	Facebook: Friends
139	Facebook: Photo Upload
140	Facebook: Events
141	Facebook: Apps
142	Facebook: Chat
143	Facebook: Questions
144	Facebook: Video Upload
145	Facebook: Groups
146	Facebook: Games
147	LinkedIn

S.no	Blacklisted Categories
148	LinkedIn: Updates
149	LinkedIn: Mail
150	LinkedIn: Connections
151	LinkedIn: Jobs
152	Twitter
153	Twitter: Posting
154	Twitter: Mail
155	Twitter: Follow
156	YouTube
157	YouTube: Commenting
158	YouTube: Video Upload
159	YouTube: Sharing
160	Instagram
161	Instagram: Upload
162	Instagram: Commenting
163	Instagram: Private Message
164	Tumblr
165	Tumblr: Posting
166	Tumblr: Commenting
167	Tumblr: Photo or Video Upload
168	Google+
169	Google+: Posting
170	Google+: Commenting
171	Google+: Photo Upload
172	Google+: Video Upload
173	Google+: Video Chat
174	Pinterest
175	Pinterest: Pin
176	Vine: Upload

S.no	Blacklisted Categories
177	Vine: Commenting
178	Vine: Message
179	Ask.fm
180	Ask.fm: Ask
181	Ask.fm: Answer
182	YikYak
183	YikYak: Posting
184	YikYak: Commenting
185	Wordpress
186	Wordpress: Posting
187	Wordpress: Upload

AppFlow

September 14, 2021

The Citrix ADC appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. It also collects webpage performance data and database information. AppFlow transmits the information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information, webpage performance data, and database information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, SSL_TCP flows, and HDX Insight flows. You can sample and filter the flow types that you want to monitor.

> **Note**
>
> For more information on HDX Insight, see HDX Insight.

AppFlow use actions and policies to send records for a selected flow to specific set of collectors. An AppFlow action specifies which set of collectors receive the AppFlow records. Policies, which are based on Advanced expressions can be configured to select flows for which flow records are sent to the collectors specified by the associated AppFlow action.

To limit the types of flows, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

Note: This feature is supported only on Citrix ADC nCore builds.

How AppFlow Works

In the most common deployment scenario, inbound traffic flows to a Virtual IP address (VIP) on the Citrix ADC appliance and is load balanced to a server. Outbound traffic flows from the server to a mapped or subnet IP address on the Citrix ADC and from the VIP to the client. A flow is a unidirectional collection of IP packets identified by the following five tuples: sourceIP, sourcePort, destIP, destPort, and protocol.

The following figure describes how the AppFlow feature works.

Figure 1. Citrix ADC Flow Sequence



As shown in the figure, the network flow identifiers for each leg of a transaction depend on the direction of the traffic.

The different flows that form a flow record are:

Flow1: <Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

Flow2: <NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

Flow3: <Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

Flow4: <VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

To help the collector link all four flows in a transaction, AppFlow adds a custom transactionID element to each flow. For application-level content switching, such as HTTP, it is possible for a single client

TCP connection to be load balanced to different back end TCP connections for each request. AppFlow provides a set of records for each transaction.

Flow Records

AppFlow records contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. AppFlow records also contain application-level information (such as HTTP URLs, HTTP request methods and response status codes, server response time, and latency). Webpage performance data (such as page load time, page render time, and time spent on the page). And database information (such as database protocol, database response status, and database response size). IPFIX flow records are based on templates that need to be sent before sending flow records.

Templates

AppFlow defines a set of templates, one for each type of flow. Each template contains a set of standard Information Elements (IEs) and Enterprise-specific Information Elements (EIEs). IPFIX templates define the order and sizes of the Information Elements (Internet Explorer) in the flow record. The templates are sent to the collectors at regular intervals, as described in RFC 5101.

A template can include the following EIEs:

- transactionID

  An unsigned 32-bit number identifying an application-level transaction. For HTTP, it corresponds to a request and response pair. All flow records that correspond to this request and response pair have the same transaction ID. In the most common case, there are four `uniflow` records that correspond to this transaction. If the Citrix ADC generates the response by itself (served from the integrated cache or by a security policy), there might be only two flow records for this transaction.

- connectionID

  An unsigned 32-bit number identifying a layer-4 connection (TCP or UDP). The Citrix ADC flows are bidirectional, with two separate flow records for each direction of the flow. This information element can be used to link the two flows.

  For the Citrix ADC, a connectionID is an identifier for the connection data structure to track the progress of a connection. In an HTTP transaction, for instance, a given connectionID might have multiple transactionID elements corresponding to multiple requests that were made on that connection.

- tcpRTT

The round trip time, in milliseconds, as measured on the TCP connection. It can be used as a metric to determine the client or server latency on the network.

- httpRequestMethod

  An 8-bit number indicating the HTTP method used in the transaction. An options template with the number-to-method mapping is sent along with the template.

- httpRequestSize

  An unsigned 32-bit number indicating the request payload size.

- httpRequestURL

  The HTTP URL requested by the client.

- httpUserAgent

  The source of incoming requests to the Web server.

- httpResponseStatus

  An unsigned 32-bit number indicating the response status code.

- httpResponseSize

  An unsigned 32-bit number indicating the response size.

- httpResponseTimeToFirstByte

  An unsigned 32-bit number indicating the time taken to receive the first byte of the response.

- httpResponseTimeToLastByte

  An unsigned 32-bit number indicating the time taken to receive the last byte of the response.

- flowFlags

  An unsigned 64-bit flag used to indicate different flow conditions.

EIEs for webpage performance data

- clientInteractionStartTime

  Time at which the browser receives the first byte of the response to load any objects of the page such as images, scripts, and style sheets.

- clientInteractionEndTime

  Time at which the browser received the last byte of response to load all the objects of the page such as images, scripts, and style sheets.

- clientRenderStartTime

  Time at which the browser starts to render the page.

- clientRenderEndTime

  Time at which a browser finished rendering the entire page, including the embedded objects.

EIEs for database information

- dbProtocolName

  An unsigned 8-bit number indicating the database protocol. Valid values are 1 for MS SQL and 2 for MySQL.

- dbReqType

  An unsigned 8-bit number indicating the database request method used in the transaction. For MS SQL, valid values are 1 is for QUERY, 2 is for TRANSACTION, and 3 is for RPC. For valid values for MySQL, see the MySQL documentation.

- dbReqString

  Indicates the database request string without the header.

- dbRespStatus

  An unsigned 64-bit number indicating the status of the database response received from the web server.

- dbRespLength

  An unsigned 64-bit number indicating the response size.

- dbRespStatString

  The response status string received from the web server.

Configuring the AppFlow feature

September 14, 2021

You configure AppFlow in the same manner as most other policy-based features. First, you enable the AppFlow feature. Then you specify the collectors to which the flow records are sent. After that, you define actions, which are sets of configured collectors. Then you configure one or more policies and associate an action to each policy. The policy tells the Citrix ADC appliance to select requests the flow records of which are sent to the associated action. Finally, you bind each policy either globally or to the specific virtual server to put it into effect.

You can further set AppFlow parameters to specify the template refresh interval and to enable the exporting of httpURL, httpCookie, and httpReferer information. On each collector, you must specify the Citrix ADC IP address as the address of the exporter.

> **Note**
>
> For information about configuring the Citrix ADC as an exporter on the collector, see the documentation for the specific collector.

The configuration utility provides tools that help users define the policies and actions. It determines exactly how the Citrix ADC appliance export records for a particular flow to a set of collectors(action.) The command line interface provides a corresponding set of CLI-based commands for experienced users who prefer a command line.

Enabling AppFlow

To be able to use the AppFlow feature, you must first enable it.

> **Note**
>
> AppFlow can be enabled only on nCore Citrix ADC appliances.

To enable the AppFlow feature by using the command line interface

At the command prompt, type one of the following commands:

```
enable ns feature AppFlow
<!--NeedCopy-->
```

To enable the AppFlow feature by using the configuration utility

Navigate to **System > Settings**, click **Configure Advanced Features**, and select the **AppFlow** option.

Specifying a Collector

A collector receives AppFlow records generated by the Citrix ADC appliance. To send the AppFlow records, you must specify at least one collector. By default, the collector listens to IPFIX messages on UDP port 4739. You can change the default port, when configuring the collector. Similarly, by default, NSIP is used as the source IP for AppFlow traffic. You can change this default source IP to a SNIP address when configuring a collector. You can also remove unused collectors.

To specify a collector by using the command line interface

> **Important**
>
> Starting from Citrix ADC release 12.1 build 55.13, you can specify the type of collector that you want to use. A new parameter “Transport” is introduced in the `add appflow collector` com-

> mand. By default, the collector listens to IPFIX messages. You can change the type of collector to either `logstream` or `ipfix` or rest by using "Transport" parameter. For more information on configuration, see the example.

At the command prompt, type the following commands to add a collector and verify the configuration:

```
- add appflow collector <name> -IPAddress <ipaddress> -port <
    port_number> -netprofile <netprofile_name> -Transport <Transport>

- show appflow collector <name>
<!--NeedCopy-->
```

Example

```
add appflow collector col1 -IPaddress 10.102.29.251 -port 8000 -
    netprofile n2 -Transport ipfix
<!--NeedCopy-->
```

To specify multiple collectors by using the command line interface

At the command prompt, type the following commands to add and send same data to multiple collectors:

```
add appflow collector <collector1> -IPAddress <IP>

add appflow collector <collector2> -IPAddress <IP>

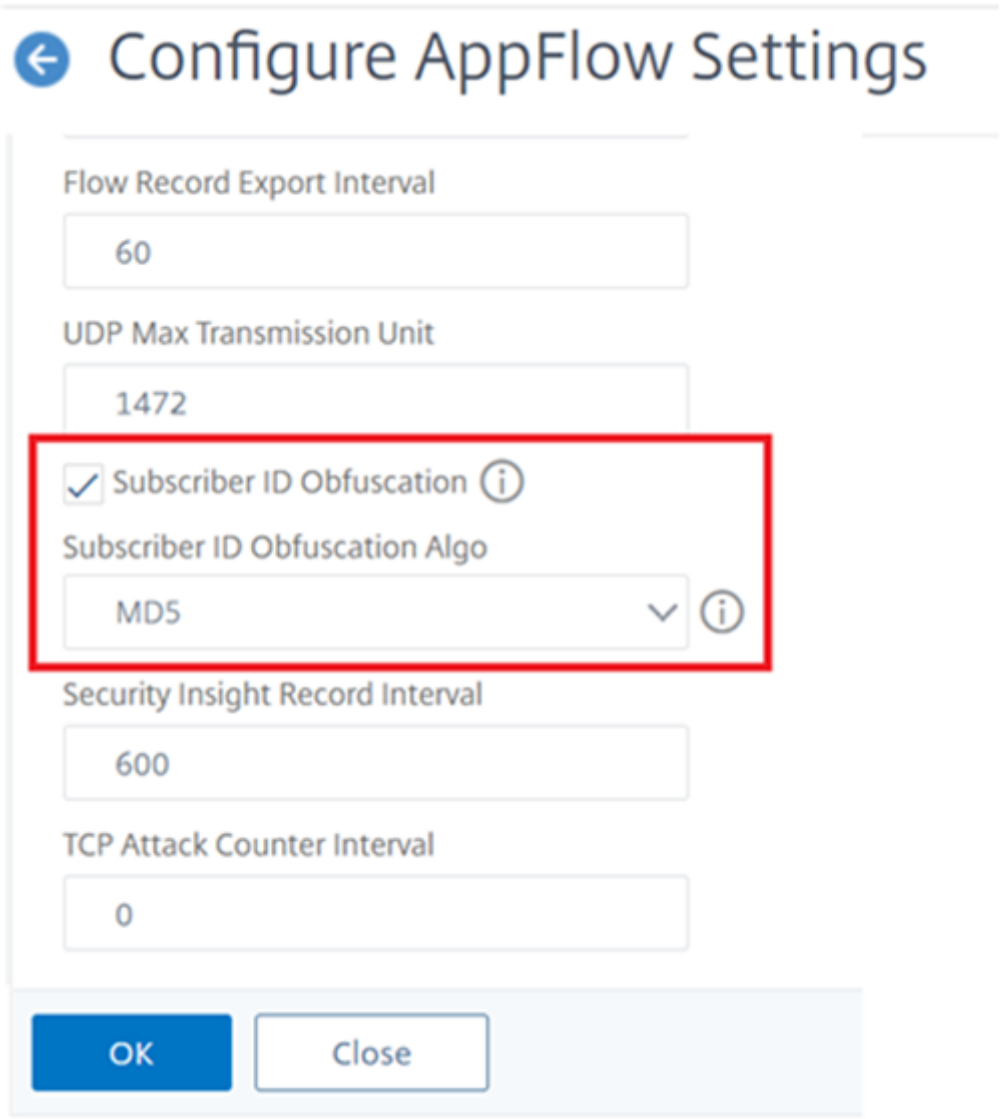
add appflow action <action> -collectors <collector1> <collector2>

add appflow policy <policy> true <action>

bind lbvserver <lbvserver> -policy <policy> -priority <priority>
<!--NeedCopy-->
```

To specify one or more collectors by using the configuration utility

Navigate to **System > AppFlow > Collectors**, and create the AppFlow collector.

Configuring an AppFlow Action

An AppFlow action is a set collector, to which the flow records are sent if the associated AppFlow policy matches.

To configure an AppFlow action by using the command line interface

At the command prompt, type the following commands to configure an AppFlow action and verify the configuration:

```
add appflow action <name> --collectors <string> ... [-
   clientSideMeasurements (Enabled|Disabled) ] [-comment <string>]

show appflow action
<!--NeedCopy-->
```

Example

```
add appflow action apfl-act-collector-1-and-3 -collectors collector-1
   collecter-3
<!--NeedCopy-->
```

To configure an AppFlow action by using the configuration utility

Navigate to **System > AppFlow > Actions**, and create the AppFlow action.

Configuring an AppFlow Policy

After you configure an AppFlow action, you must next configure an AppFlow policy. An AppFlow policy is based on a rule, which consists of one or more expressions.

> **Note**
>
> For creating and managing AppFlow policies, the configuration utility provides assistance that is not available at the command line interface.

To configure an AppFlow policy by using the command line interface

At the command prompt, type the following command to add an AppFlow policy and verify the configuration:

```
add appflow policy <name> <rule> <action>

show appflow policy <name>
<!--NeedCopy-->
```


Example

```
add appflow policy apfl-pol-tcp-dsprt client.TCP.DSTPORT.EQ(22) apfl-
   act-collector-1-and-3
<!--NeedCopy-->
```

To configure an AppFlow policy by using the configuration utility

Navigate to **System > AppFlow > Policies**, and create the AppFlow policy.

To add an expression by using the Add Expression dialog box

1. In the Add Expression dialog box, in the first list box choose the first term for your expression.
   - HTTP
     The HTTP protocol. Choose the option if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - SSL

     The protected websites. Choose the option if you want to examine some aspect of the request that pertains to the recipient of the request.
   - CLIENT

     The computer that sent the request. Choose the option if you want to examine some aspect of the sender of the request.
     When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
2. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
3. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Binding an AppFlow Policy

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the Citrix ADC, or to a specific virtual server, so that the policy applies only to the traffic related to that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the Citrix ADC operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first. Later, the policy assigned with a priority of 100, and finally the policy assigned an order of 1000.

You can leave yourself plenty of a room to add other policies in any order, and still set them to evaluate in the order you want. You can achieve by setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add more policies at any time without having to change the priority of an existing policy.

To globally bind an AppFlow policy by using the command line interface

At the command prompt, type the following command to globally bind an AppFlow policy and verify the configuration:

```
bind appflow global <policyName> <priority> [<gotoPriorityExpression [-
    type <type>] [-invoke (<labelType> <labelName>)]

show appflow global
<!--NeedCopy-->
```

Example

```
bind appflow global af_policy_lb1_10.102.71.190 1 NEXT -type
    REQ_OVERRIDE -invoke vserver google
<!--NeedCopy-->
```

To bind an AppFlow policy to a specific virtual server by using the command line interface

At the command prompt, type the following command to bind an AppFlow policy to a specific virtual server and verify the configuration:

```
bind lb vserver <name> -policyname <policy_name> -priority <priority>
<!--NeedCopy-->
```

Example


```
bind lb vserver google -policyname af_policy_google_10.102.19.179 -
   priority 251
<!--NeedCopy-->
```

To globally bind an AppFlow policy by using the configuration utility

Navigate to **System > AppFlow**, click **AppFlow policy Manager**, and select the relevant Bind Point (Default Global) and Connection Type, and then bind the AppFlow policy.

To bind an AppFlow policy to a specific virtual server by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, select the virtual server, and click **Policies**, and bind the AppFlow policy.

Enabling AppFlow for Virtual Servers

If you want to monitor only the traffic through certain virtual servers, enable AppFlow specifically for those virtual servers. You can enable AppFlow for load balancing, content switching, cache redirection, SSL VPN, GSLB, and authentication virtual servers.

To enable AppFlow for a virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver <name> <protocol> <IPAddress> <port> -appflowLog ENABLED
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -appflowLog ENABLED
<!--NeedCopy-->
```

To enable AppFlow for a virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select the virtual server, and enable AppFlow Logging option.

Enabling AppFlow for a Service

You can enable AppFlow for services that are to be bound to the load balancing virtual servers.

To enable AppFlow for a service by using the command line interface

At the command prompt, type:

```
set service <name> -appflowLog ENABLED
<!--NeedCopy-->
```

Example

```
set service ser -appflowLog ENABLED
<!--NeedCopy-->
```

To enable AppFlow for a service by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services**, select the service, and enable AppFlow Logging option.

Setting the AppFlow Parameters

You can set AppFlow parameters to customize the exporting of data to the collectors.

To set the AppFlow Parameters by using the command line interface

> **Important**
>
> - Starting from Citrix ADC release 12.1 build 55.13, you can use the NSIP to send `Logstream` records instead of the SNIP. A new parameter "logstreamOverNSIP" is introduced in the `set appflow param` command. By default, the "logstreamOverNSIP" parameter is DISABLED, you must "ENABLE" it. For more information on configuration, see the example.
> - Starting from Citrix ADC release 13.0 build 58.x release, you can enable the Web SaaS application option in AppFlow feature. It can be enabled to receive the data usage of Web or SaaS applications from the Citrix Gateway service. For more information on configuration, see the example.

At the command prompt, type the following commands to set the AppFlow parameters and verify the settings:

```
-  set appflow param [-templateRefresh <secs>] [-appnameRefresh <secs>]
     [-flowRecordInterval <secs>] [-udpPmtu <positive_integer>] [-
    httpUrl ( **ENABLED** | **DISABLED** )] [-httpCookie ( **ENABLED** |
     **DISABLED** )] [-httpReferer ( **ENABLED** | **DISABLED** )] [-
    httpMethod ( **ENABLED** | **DISABLED** )] [-httpHost ( **ENABLED**
```



```
      | **DISABLED** )] [-httpUserAgent ( **ENABLED** | **DISABLED** )] [-
      httpXForwardedFor ( **ENABLED** | **DISABLED** )][-clientTrafficOnly
       ( **YES** | **NO**)] [-webSaaSAppUsageReporting ( **ENABLED** | **
      DISABLED** )] [-logstreamOverNSIP ( **ENABLED** | **DISABLED** )]
2
3 -  show appflow Param
4 <!--NeedCopy-->
```

Example

```
1 set appflow Param -templateRefresh 240 -udpPmtu 128 -httpUrl enabled -
     webSaaSAppUsageReporting ENABLED -logstreamOverNSIP ENABLED
2 <!--NeedCopy-->
```

To set the AppFlow parameters by using the configuration utility

Navigate to **System > AppFlow**, click **Change AppFlow Settings**, and specify relevant AppFlow parameters.

Support for subscriber ID obfuscation

Starting from Citrix ADC release 13.0 build 35.xx, the AppFlow configuration is enhanced to support “subscriberIdObfuscation” algorithm for obfuscating MSISDN in layer 4 or layer 7, AppFlow records. However, before configuring the algorithm as MD5 or SHA256, you must first enable it as an AppFlow parameter. The parameter is disabled by default.

To configure the subscriber ID obfuscation algorithm by using the CLI

At the command prompt, type:

```
1 set appflow param [-subscriberIdObfuscation ( ENABLED | DISABLED )  [-
     subscriberIdObfuscationAlgo ( MD5 | SHA256 )]]
2 <!--NeedCopy-->
```

Example

```
1 set appflow param  -subscriberIdObfuscation ENABLED  -
     subscriberIdObfuscationAlgo SHA256
2 <!--NeedCopy-->
```


To configure the subscriber ID obfuscation algorithm by using the GUI

1. Navigate to **System > AppFlow**.
2. In the AppFlow detailed pane, click **Change AppFlow Setting** under **Settings**.
3. In Configure AppFlow Settings page, set the following parameters:
   - **Subscriber ID Obfuscation.** Enable the option for obfuscation MSISDN in L4/L7 AppFlow records.
   - **Subscriber ID Obfuscation Algo.** Select algorithm type as MD5 or SHA256.
4. Click **OK** and **Close**.



Example: Configuring AppFlow for DataStream

The following example illustrates the procedure for configuring AppFlow for DataStream using the command line interface.

```
enable feature appflow

add db user sa password freebsd

add lbvserver lb0 MSSQL 10.102.147.97 1433 -appflowLog ENABLED

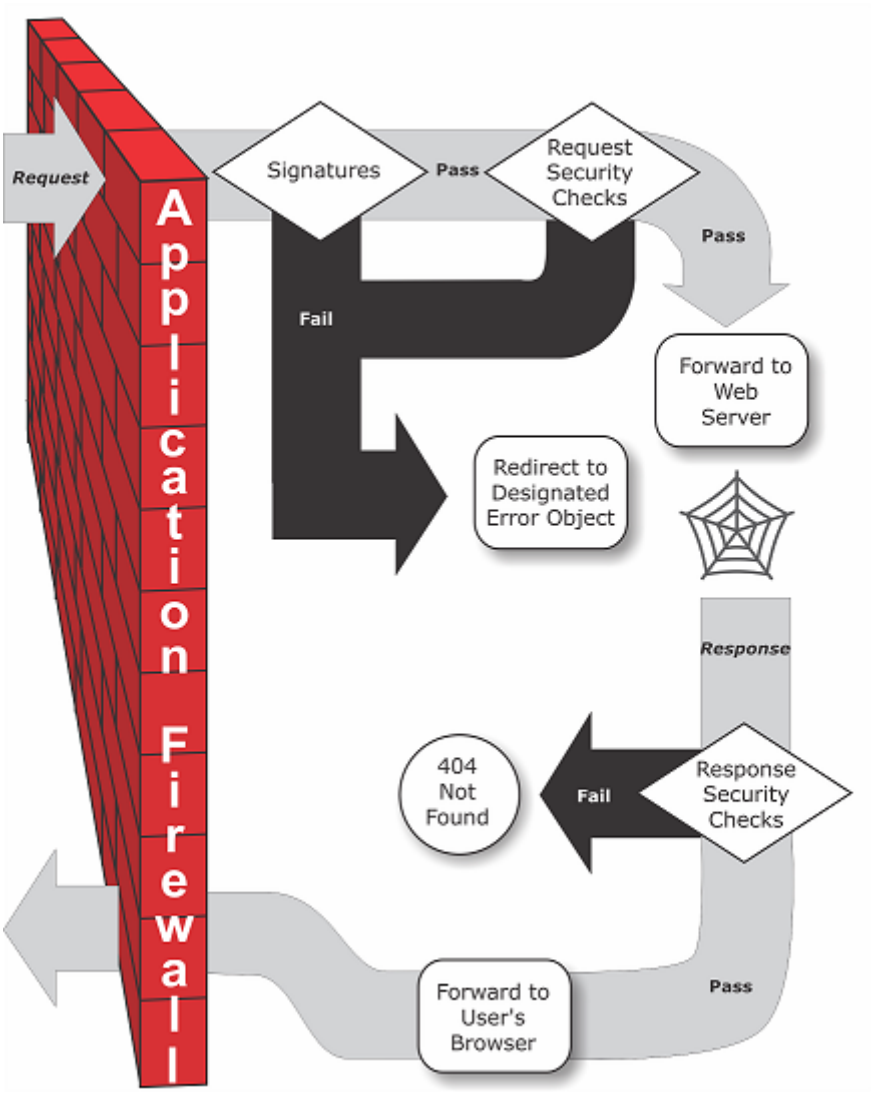
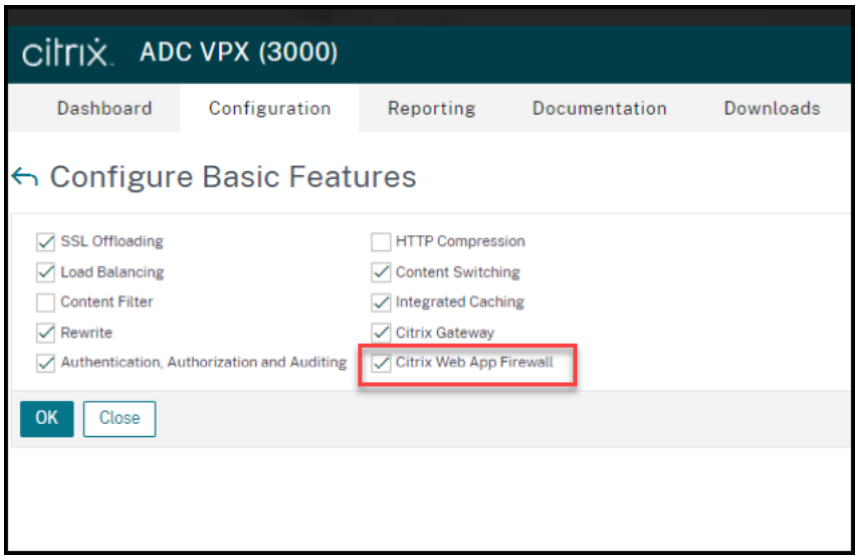
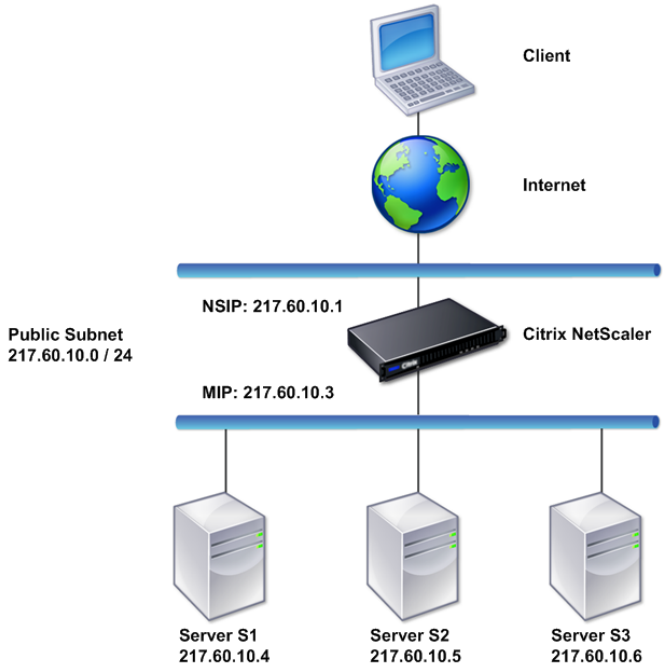
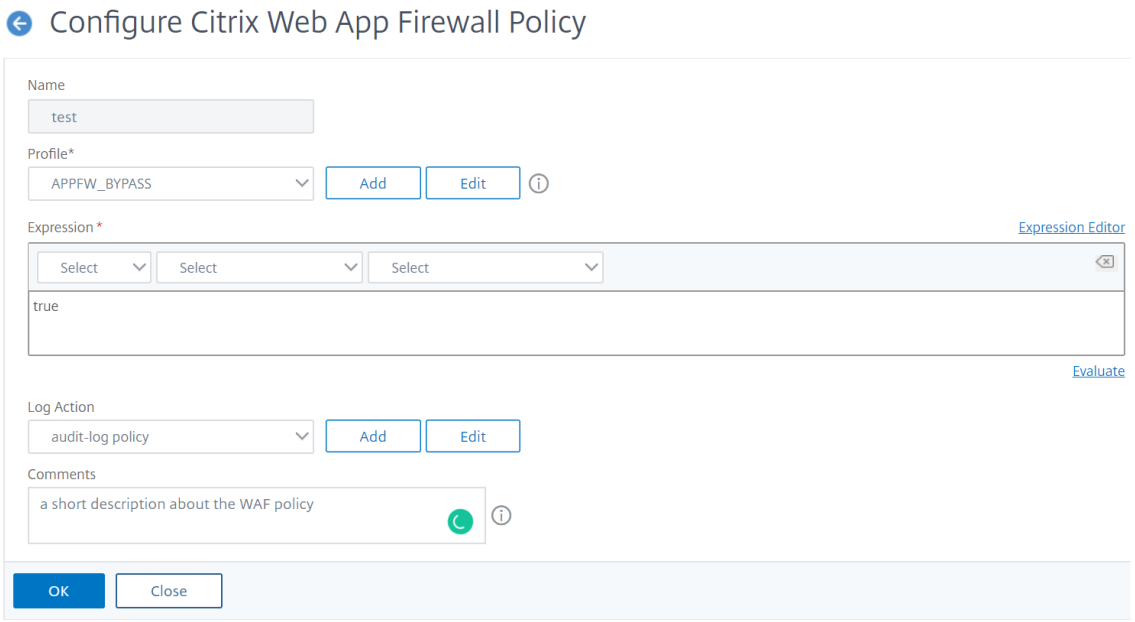
add service sv0 10.103.24.132 MSSQL 1433 -appflowLog ENABLED

bind lbvserver lb0 sv0

add appflow collector col0 -IPAddress 10.102.147.90

add appflow action act0 -collectors col0

add appflow policy pol0 "mssql.req.query.text.contains("select")" act0

bind lbvserver lb0 -policyName pol0 -priority 10
<!--NeedCopy-->
```

When the Citrix ADC appliance receives a database request, the appliance evaluates the request against a configured policy. If a match is found, the details are sent to the AppFlow collector configured in the policy.

Exporting performance data of webpages to AppFlow collector

September 14, 2021

The EdgeSight Monitoring application provides webpage monitoring data with which you can monitor the performance of various Web applications served in a Citrix ADC environment. You can now export this data to AppFlow collectors to get an in-depth analysis of the webpage applications. AppFlow, which is based on the IPFIX standard, provides more specific information about web application performance than does EdgeSight monitoring alone.

You can configure both load balancing and content switching virtual servers to export EdgeSight Monitoring data to AppFlow collectors. Before configuring a virtual server for AppFlow export, associate an AppFlow action with the EdgeSight Monitoring responder policy.

The following webpage performance data is exported to AppFlow:

- **Page Load Time**. Elapsed time, in milliseconds, from when the browser starts to receive the first byte of a response until the user starts to interact with the page. At this stage, all the page content might not be loaded.
- **Page Render Time**. Elapsed time, in milliseconds, from when the browser receives the first byte of response until either all page content has been rendered or the page load action has timed out.
- **Time Spent on the Page**. Time spent by users on a page. Represents the time from one page request to the next one.

AppFlow transmits the performance data by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. The AppFlow templates use the following enterprise-specific Information Elements (EIEs) to export the information:

- **Client Load End Time**. Time at which the browser received the last byte of a response to load all the objects of the page such as images, scripts, and style sheets.
- **Client Load Start Time**. Time at which the browser receives the first byte of the response to load any objects of the page such as images, scripts, and Style sheets.
- **Client Render End Time**. Time at which a browser finished rendering the entire page, including the embedded objects.
- **Client Render Start Time**. Time at which the browser started rendering the page.

Prerequisites for exporting performance data of webpages to AppFlow collectors

Before associating the AppFlow action with the AppFlow policy, verify that the following prerequisites have been met:

- The AppFlow feature has been enabled and configured.
- The Responder feature has been enabled.
- The EdgeSight Monitoring feature has been enabled.
- EdgeSight Monitoring has been enabled on the load balancing or content switching virtual servers bound to the services of applications for which you want to collect the performance data.

Associating an AppFlow action with the EdgeSight monitoring responder policy

To export the webpage performance data to the AppFlow collector, you must associate an AppFlow action with the EdgeSight Monitoring responder policy. An AppFlow action specifies which set of collectors receive the traffic.

To associate an AppFlow action with the EdgeSight Monitoring Responder policy by using the CLI

At the command prompt, type:

```
set responder policy <name> -appflowAction <action_Name>
<!--NeedCopy-->
```

Example

```
set responder policy pol -appflowAction actn
<!--NeedCopy-->
```

To associate an AppFlow action with the EdgeSight Monitoring Responder policy by using the GUI

1. Navigate to **AppExpert** > **Responder** > **Policies**.
2. In the details pane, select an EdgeSight Monitoring responder policy, and then click **Open**.
3. In the **Configure Responder Policy** dialog box, in the **AppFlow Action** drop-down list, select the AppFlow action associated with the collectors to which you want to send the webpage performance data.
4. Click **OK**.

Configuring a virtual server to export EdgeSight statistics to AppFlow collectors

To export EdgeSight statistics information from a virtual server to the AppFlow collector, you must associate an AppFlow action with the virtual server.

To associate an AppFlow action with a Load Balancing or Content Switching virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**. You can also navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**.
2. In the details pane, select a virtual server, or multiple virtual servers, and then click **Enable EdgeSight Monitoring**.
3. In the Enable EdgeSight Monitoring dialog box, select the **Export EdgeSight statistics to Appflow** check box.
4. From the AppFlow Action drop-down list, select the **AppFlow** action. The AppFlow action defines the list of AppFlow collectors to which it exports EdgeSight Monitoring statistics. If you have selected multiple load balancing virtual servers, the same AppFlow Action is associated

with the responder policies bound to them. You can later change the AppFlow Action configured for each of the selected Load Balancing virtual server individually, if necessary.

5. Click **OK**.

Session reliability on Citrix ADC high availability pair

September 14, 2021

When a network disruption or a device failover occurs during an ICA session, session reconnection can use one of two mechanisms: session reliability or Auto Client Reconnect.

Session reliability. The preferred mode, is a smooth experience for the user. The disruption is barely noticeable for brief network interruptions.

Auto client reconnect. The fallback option, involves restarting the client. This mechanism is disruptive for the user and is not always supported.

Receivers can reconnect their ICA sessions seamlessly using the ICA session reliability feature, when HDX Insight is enabled.
This feature works both in standalone and in a Citrix ADC HA pair configuration, and even when a Citrix ADC failover happens.

> **Note:**
>
> - Citrix ADC appliances must be running on software version 11.1 build 49.16 or later.
> - You must not enable or disable session reliability mode when the Citrix ADC appliances have active connections.
> - Enabling or disabling the feature when connections are still active causes HDX Insight to stop parsing those sessions after a failover occurs. It results in the loss of information about the sessions.
> - Session reliability on a high availability setup is disabled by default for Citrix ADC software version 11.1 49.16 or later. Session reliability is supported on a high availability setup only if both the nodes of the setup run the same build (for example, release 11.1 build 53). In other words, session reliability is not supported on a high availability setup if both the nodes run different builds (for example, one node has release 11.1 build 53 whereas the other has release 11.1 build 56). Session reliability for SSL VDA is supported if the following conditions are met:
>   - The "EnableSRonHAFailover" parameter in the `set ica parameter` command must be YES.
>   - The HTTPS must be used instead of HTTP while configuring the virtual server.
> - When HDX Insight is enabled, basic encryption applications and desktops reconnect after

> high availability failover even if the EnableSRonHAFailover parameter is disabled.

To configure Session reliability by using CLI:

1. At the command line, use the default system administrator credentials to log on to the system.
2. To enable session reliability on HA failover, at the prompt, type: `set ica parameter EnableSRonHAFailover YES`
3. To disable session reliability on HA failover, at the prompt, type: `set ica parameter EnableSRonHAFailover NO`

To Enable Session reliability on HA failover by using GUI:

1. In a web browser, type the IP address of the primary Citrix ADC instance in the HA pair (for example, http://192.168.100.1).
2. In **User Name** and **Password**, enter the administrator credentials.
3. On the **Configuration** tab, navigate to **System** > **Settings**, and click **Change ICA Parameters**.
4. In the **Change ICA Parameters** section, select **Session Reliability on HA Failover**.
5. Click **OK**.

Limitations

- Enabling this feature results in increased bandwidth consumption which is due to ICA compression being turned off by the feature. And the extra traffic between the primary and secondary nodes to keep them in sync.
- This feature is supported in Active-Passive mode only. Active-Active mode is not currently supported.
- When HDX Insight is enabled, and session reliability on the HA knob is set to NO, only ACR reconnect mode is supported in the Citrix ADC high availability failover scenario. The HA knob does not disable session reliability if HDX Insight is disabled.

The **Session Reconnect Semantics** table is as follows:

Session reconnects semantics

Status	EnableSRonHAFailover Yes	EnableSRonHAFailover No (Default)
HDX Insight enabled	Session reconnect for ICA Session works	Session reconnect for ICA Sessions does not work
HDX Insight disabled	Session reconnect for ICA Session works	Session reconnect for ICA Sessions works

Points to note

- Session reliability for ICA sessions works out of the box with Citrix Gateway.
- Session reliability for ICA sessions does not work ONLY when both the following conditions are met:
  - HDX Insight is enabled
  - EnableSRonHAFailover is set to NO
- Setting the EnableSRonHAFailover knob to either YES or NO does not make any difference, when HDX Insight is disabled.

Citrix Web App Firewall

September 14, 2021

The following topics cover the installation and configuration details of the Citrix Web App Firewall feature.

Introduction	An overview of web security and how the Web App Firewall works.
Configuration	How to configure the Web App Firewall to protect a website, a web service, or a Web 2.0 site.
Signatures	A detailed description about signatures and how to configure it from a supported vulnerability scanning tool, and define your own signatures, with examples.
Overview of Security Checks	A detailed description of Web App Firewall security checks, with configuration information and examples.
Profiles	A description of how profiles are configured and used in the Web App Firewall.
Policies	A description of how policies are used when configuring the Web App Firewall, with examples of useful policies.
Imports	A description of how the Web App Firewall uses different types of imported files, and how to import and export files.

Global Configuration	A description of Web App Firewall features that apply to all profiles, and how to configure them.
Use Cases	Extended examples that demonstrate how to set up the Web App Firewall to best protect specific types of more complex websites and web services.
Logs, Statistics, and Reports	How to access and use the Web App Firewall logs, the statistics, and the reports to help with configuring the Web App Firewall.

The Citrix Web App Firewall offers easy to configure options to meet a wide range of application security requirements. Web App Firewall profiles, which consist of sets of security checks, can be used to protect both the requests and the responses by providing deep packet-level inspections. Each profile includes an option to select basic protections or advanced protections. Some protections might require use of other files. For example, xml validation checks might require WSDL or schema files. The profiles can also use other files, such as signatures or error objects. These files can be added locally, or they can be imported ahead of time and saved on the appliance for future use.

Each policy identifies a type of traffic, and that traffic is inspected for the security check violations specified in the profile that is associated with the policy. The policies can have different bind points, which determine the scope of the policy. For example, a policy that is bound to a specific virtual server is invoked and evaluated for only the traffic flowing through that virtual server. The policies are evaluated in the order of their designated priorities, and the first one that matches the request or response is applied.

- Quick Deployment of Web App Firewall Protection

  You can use the following procedure for quick deployment of Web App Firewall security:

  1. Add a Web App Firewall profile and select the appropriate type (html, xml, JSON) for the security requirements of the application.
  2. Select the required level of security (basic or advanced).
  3. Add or import the required files, such as signatures or WSDL.
  4. Configure the profile to use the files, and make any other necessary changes to the default settings.
  5. Add a Web App Firewall policy for this profile.
  6. Bind the policy to the target bind point and specify the priority.

- Web App Firewall entities

Profile—An Web App Firewall profile specifies what to look for and what to do. It inspects both the request and the response to determine which potential security violations must be checked and what actions must be taken when processing a transaction. A profile can protect an HTML, XML, or HTML and XML payload. Depending on the security requirements of the application, you can create either a basic or an advanced profile. A basic profile can protect against known attacks. If higher security is required, you can deploy an advanced profile to allow controlled access to the application resources, blocking zero day attacks. However, a basic profile can be modified to offer advanced protections, and conversely. Multiple action choices (for example, block, log, learn, and transform) are available. Advanced security checks might use session cookies and hidden form tags for controlling and monitoring the client connections. Web App Firewall profiles can learn the triggered violations and suggest the relaxation rules.

Basic Protections—A basic profile includes a preconfigured set of Start URL and Deny URL relaxation rules. These relaxation rules determine which requests must be allowed and which must be denied. Incoming requests are matched against these lists and the configured actions are applied. This allows the user to be able to secure applications with minimal configuration for relaxation rules. The Start URL rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-site scripting can also be easily detected.

Advanced Protections—As the name indicates, advanced protections are used for applications that have higher security requirements. Relaxation rules are configured to allow access to only specific data and block the rest. This positive security model mitigates unknown attacks, which might not be detected by basic security checks. In addition to all the basic protections, an advanced profile keeps track of a user session by controlling the browsing, checking for cookies, specifying input requirements for various form fields, and protecting against tampering of forms or cross-site request forgery attacks. Learning, which observes the traffic and deploys the appropriate relaxations, is enabled by default for many security checks. Although easy to use, advanced protections require due consideration, because they offer tighter security but also require more processing and do not allow use of caching, which can affect performance.

Import—Import functionality is useful when Web App Firewall profiles must use external files, that is, files hosted on an external or internal web server, or that have to be copied from a local machine. Importing a file and storing it on the appliance is useful, especially in situations where you have to control access to external websites, or where compilation takes a long time, large files have to be synced across HA deployments, or you can reuse a file by copying it across multiple devices. For example:

- WSDLs hosted on external web servers can be imported locally before blocking access to external websites.
- Large signature files generated by an external scan tool such as Cenzic can be imported

and precompiled, using schema on the Citrix appliance.
- A customized HTML or XML error page can be imported from an external web server or copied from a local file.

Signatures—Signatures are powerful, because they use pattern matching to detect malicious attacks and can be configured to check both the request and the response of a transaction. They are a preferred option when a customizable security solution is needed. Multiple choices (for example, block, log, learn, and transform) are available for the action to take when a signature match is detected. The Web App Firewall has a built-in default signature object consisting of more than 1,300 signature rules, with an option to get the latest rules by using the auto-update feature. Rules created by other scan tools can also be imported. The signature object can be customized by adding new rules, which can work with the other security checks specified in the Web App Firewall profile. A signature rule can have multiple patterns and can flag a violation only when all the patterns are matched, thereby avoiding false positives. Careful selection of a literal `fastmatch` pattern for a rule can significantly optimize processing time.

Policies—Web App Firewall Policies are used to filter and separate the traffic into different types. This provides the flexibility to implement different levels of security protections for the application data. Access to highly sensitive data can be directed to advanced security-check inspections, while less sensitive data is protected by basic-level security inspections. Policies can also be configured to bypass security-check inspection for harmless traffic. Higher security requires more processing, so careful design of the policies can provide desired security along with optimized performance. The priority of the policy determines the order in which it is evaluated, and its bind point determines the scope of its application.

Highlights

1. Ability to secure a wide range of applications by protecting different types of data, implementing the right level of security for different resources, and still getting maximum performance.
2. Flexibility to add or modify a security configuration. You can tighten or relax security checks by enabling or disabling basic and advanced protections.
3. Option to convert an HTML profile to an XML or Web2.0 (HTML+XML) profile and conversely, providing the flexibility to add security for different types of payload.
4. Easily deployed actions to block attacks, monitor them in logs, collect statistics, or even transform some attack strings to render them harmless.
5. Ability to detect attacks by inspecting incoming requests, and to prevent leakage of sensitive data by inspecting the responses sent by the servers.
6. Capability to learn from the traffic pattern to get recommendations for easily editable relaxation rules that can be deployed to allow exceptions.
7. Hybrid security model that applies the power of customizable signatures to block attacks that match specified patterns, and provides the flexibility to use the positive-security-model checks

for basic or advanced security protections.

8. Availability of comprehensive configuration reports, including information about PCI-DSS compliance.

FAQs and Deployment Guide

September 14, 2021

Q: Why is Citrix Web App Firewall the preferred choice for securing applications?

With the following features, the Citrix Web App Firewall offers a comprehensive security solution:

- **Hybrid security model:** Citrix ADC hybrid security model allows you to take advantage of both a positive security model and a negative security model to come up with a configuration ideally suited for your applications.
  - **Positive security model** protects against Buffer Overflow, CGI-BIN Parameter Manipulation, Form/Hidden Field Manipulation, Forceful Browsing, Cookie or Session Poisoning, Broken ACLs, Cross-Site Scripting (cross-site scripting), Command Injection, SQL Injection, Error Triggering Sensitive Information Leak, Insecure Use of Cryptography, Server Misconfiguration, Back Doors and Debug Options, Rate-Based Policy Enforcement, Well Known Platform Vulnerabilities, Zero-Day Exploits, Cross Site Request Forgery (CSRF), and leakage of Credit Card and other sensitive data.
  - **Negative security model** uses a rich set signatures to protect against L7 and HTTP application vulnerabilities. The Web App Firewall is integrated with several third party scanning tools, such as those offered by Cenzic, Qualys, Whitehat, and IBM. The built-in XSLT files allow easy importation of rules, which can be used in conjunction with the native-format Snort based rules. An auto-update feature gets the latest updates for new vulnerabilities.

The positive security model might be the preferred choice for protecting applications that have a high need for security, because it gives you the option to fully control who can access what data. You allow only what you want and block the rest. This model includes a built-in security check configuration, which is deployable with few clicks. However, keep in mind that the tighter the security, the greater the processing overhead.

The negative security model might be preferable for customized applications. The signatures allow you to combine multiple conditions, and a match and the specified action are triggered only when all the conditions are satisfied. You block only what you don't want and allow the rest. A specific fast-match pattern in a specified location can significantly reduce processing overhead to optimize performance. The option to add your own signature rules, based on the specific security needs of your applications, gives you the flexibility to design your own customized security solutions.

- **Request as well as response side detection and protection:** You can inspect the incoming requests to detect any suspicious behavior and take appropriate actions, and you can check the responses to detect and protect against leakage of sensitive data.
- **Rich set of built-in protections for HTML, XML and JSON payloads:** The Web App Firewall offers 19 different security checks. Six of them (such as Start URL and Deny URL) apply to both HTML and XML data. Five checks (such as Field Consistency and Field Format) are specific to HTML, and eight (such as XML Format and Web Service Interoperability) are specific to XML payloads. This feature includes a rich set of actions and options. For example, URL Closure enables you to control and optimize the navigation through your website, to safeguard against forceful browsing without having to configure relaxation rules to allow each and every legitimate URL. You have the option to remove or x-out the sensitive data, such as credit-card numbers, in the response. Be it SOAP Array attack protection, XML denial of Service (XDoS), WSDL scan prevention, Attachment check, or any number of other XML attacks, you have the comfort of knowing that you have an ironclad shield protecting your data when your applications are protected by the Web App Firewall. The signatures allow you to configure rules using XPATH-Expressions to detect violations in the body as well as the header of a JSON payload.
- **GWT:** Support for protecting Google Web Toolkit applications to safeguard against SQL, cross-site scripting and Form Field Consistency check violations.
- **Java-free, user friendly graphical user interface (GUI):** An intuitive GUI and preconfigured security checks make it easy to deploy security by clicking a few buttons. A wizard prompts and guides you to create the required elements, such as profiles, policies, signatures, and bindings. The HTML5 based GUI is free of any Java dependency. It's performance is significantly better than that of the older, Java based versions.
- **Easy to Use and automatable CLI:** Most of the configuration options that are available in GUI are also available in the command line interface (CLI). The CLI commands can be executed by a batch file and are easy to automate.
- **Support for REST API:** The Citrix ADC NITRO protocol supports a rich set of REST API's to automate Web App Firewall configuration and collect pertinent statistics for ongoing monitoring of security violations.
- **Learning:** The Web App Firewall's ability to learn by monitoring traffic to fine tune security is very user friendly. The learning engine recommends rules, which makes it easy to deploy relaxations without proficiency in regular expressions.
- **RegEx editor support:** Regular expression offer an elegant solution to the dilemma of wanting to consolidate rules and yet optimize search. You can capitalize on the power of regular expressions to configure URLs, field names, signature patterns, and so on. The rich built-in GUI RegEx editor offers you a quick reference for the expressions and provides a convenient way to validate and test your RegEx for accuracy.

- **Customized error page:** Blocked requests can be redirected to an error URL. You also have the option to display a customized error object that uses supported variables and Citrix default syntax (advanced PI expressions) to embed troubleshooting information for the client.
- **PCI-DSS, stats, and other violation reports:** The rich set of reports makes it easy to meet the PCI-DSS compliance requirement, gather stats about traffic counters, and view violation reports for all profiles or just one profile.
- **Logging and click-to-rule from log:** Detailed logging is supported for native as well as CEF format. The Web App Firewall offers you the ability to filter targeted log messages in the syslog viewer. You can select a log message and deploy a corresponding relaxation rule by a simple click of a button. You have the flexibility to customize log messages and also have support for generating web logs. For additional details, see Web App Firewall Logstopic.
- **Include violation logs in trace records:** The ability to include log messages in the trace records makes it very easy to debug unexpected behavior such as reset and block.
- **Cloning:** The useful Import/Export profile option allows you to clone the security configuration from one Citrix ADC appliance to others. Export learned data options make it easy to export the learned rules to an Excel file. You can then get them reviewed and approved by application owner before applying them.
- **An AppExpert template** (a set of configuration settings) can be designed to provide appropriate protection for your websites. You can simplify and expedite the process of deploying similar protection on other appliances by exporting these cookie-cutter templates to a template.

For additional details, see AppExpert template topic.

- **Sessionless security checks:** Deploying sessionless security checks can help you reduce the memory footprint and expedite the processing.
- **Interoperability with other Citrix ADC features:** The Web App Firewall works seamlessly with other Citrix ADC features, such as rewrite, URL transformation, integrated caching, CVPN, and rate limiting.
- **Support of PI expressions in policies:** You can leverage the power of advanced PI expressions to design policies to implement different levels of security for different parts of your application.
- **Support for IPv6:** The Web App Firewall supports both IPv4 and IPv6 protocols.
- **Geolocation based security protection:** You have the flexibility of using Citrix default syntax (PI Expressions) for configuring location based policies, which can be used in conjunction with a built-in location database to customize firewall protection. You can identify the locations from which malicious requests originate, and enforce the desired level of security-check inspections for requests that originate from a specific geographical location.
- **Performance:** Request-side **streaming** significantly improves performance. As soon as a field is processed, the resulting data is forwarded to the back end while evaluation continues for the remaining fields. The improvement in processing time is especially significant when handling

large posts.

- **Other security features:** The Web App Firewall has several other security settings that can help ensure the security of your data. For example, the **Confidential Field** lets you block leakage of sensitive information in the log messages, and **Strip HTML Comment** allows you to remove the HTML comments from the response before forwarding it to the client. **Field Types** can be used to specify what inputs are allowed in the forms submitted to your application.

Q: What do I need to do to configure Web App Firewall?

Do the following:

- Add an Web App Firewall profile and select the appropriate type (html, xml, web2.0) for the security requirements of the application.
- Select the required level of security (basic or advanced).
- Add or import the required files, such as signatures or WSDL.
- Configure the profile to use the files, and make any other necessary changes to the default settings.
- Add an Web App Firewall policy for this profile.
- Bind the policy to the target bind point and specify the priority.

Q: How do I know what profile type to choose?

The Web App Firewall profile offers protection for both HTML and XML payloads. Depending on the need of your application, you can choose either a HTML profile or XML profile. If your application supports both HTML and XML data, you can choose a Web2.0 profile.

Q: What is the difference between basic and advanced profiles? How do I decide which one I need?

The decision to use a basic or an advance profile depends on the security need of your application. A basic profile includes a preconfigured set of Start URL and Deny URL relaxation rules. These relaxation rules determine which requests are allowed and which are denied. Incoming requests are matched with the preconfigured rules, and the configured actions are applied. The user can secure applications with minimal configuration of relaxation rules. The Start URL rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-Site Scripting can also be easily detected.

As the name indicates, advanced protections are for applications that have higher security requirements. Relaxation rules are configured to allow access to only specific data and block the rest. This positive security model mitigates unknown attacks, which might not be detected by basic security

checks. In addition to all the basic protections, an advanced profile keeps track of a user session by controlling the browsing, checking for cookies, specifying input requirements for various form fields, and protecting against tampering of forms or Cross-Site Request Forgery attacks. Learning, which observes the traffic and recommends the appropriate relaxations, is enabled by default for many security checks. Although easy to use, advanced protections require due consideration, because they offer tighter security but also require more processing. Some advance security checks do not allow use of caching, which can affect performance.

Keep the following points in mind when deciding whether to use basic or advanced profiles:

- Basic and advanced profiles are just starting templates. You can always modify the basic profile to deploy advanced security features, and vice versa.
- Advanced security checks require more processing and can affect performance. Unless your application needs advanced security, you might want to start with a basic profile and tighten the security as required for your application.
- You do not want to enable all security checks unless your application needs it.

Q: What is a policy? How do I select the bind point and set the priority?

Web App Firewall policies can help you sort your traffic into logical groups for configuring different levels of security implementation. Carefully select the bind points for the policies to determine which traffic is matched against which policy. For example, if you want every incoming request to be checked for SQL/cross-site scripting attacks, you can create a generic policy and bind it globally. Or, if you want to apply more stringent security checks to the traffic of a virtual server hosting applications that contain sensitive data, you can bind a policy to that virtual server.

Careful assignment of priorities can enhance the traffic processing. You want to assign higher priorities to more specific policies and lower priorities to generic policies. Note that the higher the number, the lower the priority. A policy with a priority of 10 is evaluated before a policy that has a priority of 15.

You can apply different levels of security for different kinds of contents, e.g. requests for static objects like images and text can be by-passed by using one policy and requests for other sensitive contents can be subjected to a much stringent check by using a second policy.

Q: How do I go about configuring the rules to secure my application?

The Web App Firewall makes it very easy to design the right level of security for your web-site. You can have multiple Web App Firewall policies, bound to different Web App Firewall profiles, to implement different levels of security-check inspections for your applications. You can initially monitor the logs to observe what security threats are being detected and which violations are being triggered. You can either manually add the relaxation rules or take advantage of the Web App Firewall's recommended learned rules to deploy the required relaxations to avoid false positives.

The Citrix Web App Firewall offers **visualizer** support in GUI, which makes rule management very easy. You can easily view all the data on one screen, and take action on several rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate several rules. You can select a subset of the rules, basing your selection on the delimiter and Action URL. Visualizer support is available for viewing 1) learned rules and 2) relaxation rules.

1. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. You can also skip (ignore) rules.
2. The visualizer for deployed relaxations offers you the option to add a new rule or edit an existing one. You can also enable or disable a group of rules by selecting a node and clicking the **Enable** or **Disable** button in the relaxation visualizer.

Q: What are signatures? How do I know which signatures to use?

A signature is an object that can have multiple rules. Each rule consists of one or more patterns that can be associated with a specified set of actions. The Web App Firewall has a built-in default signature object consisting of more than 1,300 signature rules, with an option to get the latest rules by using the **auto-update** feature to get protection against new vulnerabilities. Rules created by other scan tools can also be imported.

Signatures are very powerful because they use pattern matching to detect malicious attacks and can be configured to check both the request and the response of a transaction. They are a preferred option when a customizable security solution is needed. Multiple action choices (for example, block, log, learn, and transform) are available for when a signature match is detected. The default signatures cover rules to protect different types of applications, such as web-cgi, web-coldfusion, web-frontpage, web-iis, web-php, web-client, web-activex, web-shell-shock, and web-struts. To match the needs of your application, you can select and deploy the rules belonging to a specific category.

Signature-usage tips:

- You can just make a copy of the default signature object and modify it to enable the rules you need and configure the actions you want.
- The signature object can be customized by adding new rules, which can work in conjunction with other signature rules.
- The signature rules can also be configured to work in conjunction with the security checks specified in the Web App Firewall profile. If a match indicating a violation is detected by a signature as well as a security check, the more restrictive action is the one that gets enforced.
- A signature rule can have multiple patterns and be configured to flag a violation only when all the patterns are matched, thereby avoiding false positives.
- Careful selection of a literal fast-match pattern for a rule can significantly optimize processing time.

Q: Does the Web App Firewall work with other Citrix ADC features?

The Web App Firewall is fully integrated into the Citrix ADC appliance and works seamlessly with other features. You can configure maximum security for your application by using other Citrix ADC security features in conjunction with the Web App Firewall. For example, **AAA-TM** can be used to authenticate the user, check the user's authorization to access the content, and log the accesses, including invalid login attempts. **Rewrite** can be used to modify the URL or to add, modify or delete headers, and **Responder** can be used to deliver customized content to different users. You can define the maximum load for your website by using **Rate Limiting** to monitor the traffic and throttle the rate if it is too high. **HTTP Denial-of-Service (DoS)** protection can help distinguish between real HTTP clients and malicious DoS clients. You can narrow the scope of security-check inspection by binding the Web App Firewall policies to virtual servers, while still optimizing the user experience by using the **Load Balancing** feature to manage heavily used applications. Requests for static objects such as images or text can bypass security check inspection, taking advantage of **integrated caching** or **compression** to optimize the bandwidth usage for such content.

Q: How is the payload processed by the Web App Firewall and the other Citrix ADC features?

A diagram showing details of the L7 packet flow in a Citrix ADC appliance is available in the Processing Order of Features section.

Q: What is the recommended workflow for Web App Firewall deployment?

Now that you know the advantages of using the state-of-the-art security protections of the Citrix Web App Firewall, you might want to collect additional information that can help you design the optimal solution for your security needs. Citrix recommends that you do the following:

- **Know your environment:** Knowing your environment will help you to identify the best security protection solution (signatures, security checks, or both) for your needs. Before you begin configuration, you must gather the following information.
  - **OS:** What kind of OS (MS Windows, Linux, BSD, Unix, others) do you have?
  - **Web Sever:** What web server (IIS, Apache or Citrix ADC Enterprise Server) are you running?
  - **Application:** What type of applications are running on your application server (for example, ASP.NET, PHP, Cold Fusion, ActiveX, FrontPage, Struts, CGI, Apache Tomcat, Domino, and WebLogic)?
  - Do you have customized applications or off-the-shelf (for example, Oracle, SAP) applications? What version you are using?
  - **SSL:** Do you require SSL? If so, what key size (512, 1024, 2048, 4096) is used for signing certificates?

  - **Traffic Volume:** What is the average traffic rate through your applications? Do you have seasonal or time-specific spikes in the traffic?
  - **Server Farm:** How many servers do you have? Do you need to use load balancing?
  - **Database:** What type of database (MS-SQL, MySQL, Oracle, Postgres, SQLite, nosql, Sybase, Informix and so forth.) do you use?
  - **DB Connectivity:** What kind of data base connectivity do you have (DSN, per-file connection string, single file connection string) and what drivers are used?
- **Identify your security needs:** You might want to evaluate which applications or specific data need maximum security protection, which ones are less vulnerable, and the ones for which security inspection can safely be bypassed. This will help you in coming up with an optimal configuration, and in designing appropriate policies and bind points to segregate the traffic. For example, you might want to configure a policy to bypass security inspection of requests for static web content, such as images, MP3 files, and movies, and configure another policy to apply advanced security checks to requests for dynamic content. You can use multiple policies and profiles to protect different contents of the same application.
- **License requirement:** Citrix offers a unified solution to optimize the performance of your application by taking advantage of a rich set of features such as load balancing, content switching, caching, compression, responder, rewrite, and content filtering, to name a few. Identifying the features that you want can help you decide which license you need.
- **Install and baseline a Citrix ADC appliance:** Create a virtual server and run test traffic through it to get an idea of the rate and amount of traffic flowing through your system. This information will help you to identify your capacity requirement and select the right appliance (VPX, MPX, or SDX).
- **Deploy the Web App Firewall:** Use the Web App Firewall wizard to proceed with a simple security configuration. The wizard walks you through several screens and prompts you to add a profile, policy, signature, and security checks.
  - **Profile:** Select a meaningful name and the appropriate type (HTML, XML or WEB 2.0) for your profile. The policy and signatures will be auto-generated using the same name.
  - **Policy:** The auto-generated policy has the default Expression (true), which selects all traffic and is bound globally. This is a good starting point unless you have in mind a specific policy that you want to use.
  - **Protections:** The wizard helps you take advantage of the hybrid security model, in which you can use the default signatures offering a rich set of rules to protect different types of applications. **Simple** edit mode allows you to view the various categories (CGI, Cold Fusion, PHP, and so forth.). You can select one or more categories to identify a specific set of rules applicable to your application. Use the **Action** option to enable all the signature rules in the selected categories. Make sure that blocking is disabled, so that you can monitor the traffic before tightening the security. Click **Continue**. In the **Specify Deep protections** pane, you can make changes as needed to deploy the security check protections. In most

cases, basic protections are sufficient for initial security configuration. Run the traffic for a while to collect a representative sample of the security-inspection data.

- **Tightening the security:** After deploying Web App Firewall and observing the traffic for a while, you can start tightening the security of your applications by deploying relaxations and then enabling blocking. **Learning**, **Visualizer**, and **Click to deploy rules** are useful features that make it very easy to tweak your configuration to come up with just the right level of relaxation. At this point, you can also change the policy expression and/or configure additional policies and profiles to implement desired levels of security for different types of content.
- **Debugging:** If you see unexpected behavior of your application, the Web App Firewall offers various options for easy debugging:
  * **Log**. If legitimate requests are getting blocked, your first step is to check the ns.log file to see if any unexpected security-check violation is being triggered.
  * **Disable feature.** If you do not see any violations but are still seeing unexpected behavior, such as an application resetting or sending partial responses, you can disable the Web App Firewall feature for debugging. If the issue persists, it rules out the Web App Firewall as a suspect.
  * **Trace records with log messages.** If the issue appears to be Web App Firewall related and needs closer inspection, you have the option to include security violation messages in an nstrace. You can use "Follow TCP stream" in the trace to view the details of the individual transaction, including headers, payload, and the corresponding log message, together on the same screen. Details of how to use this functionality are available at Appendixes.

Introduction to Citrix Web Application Firewall

September 14, 2021

The Citrix Web App Firewall prevents security breaches, data loss, and possible unauthorized modifications to websites that access sensitive business or customer information. It does so by filtering both requests and responses, examining them for evidence of malicious activity, and blocking requests that exhibit such activity. Your site is protected not only from common types of attacks, but also from new, as yet unknown attacks. In addition to protecting web servers and websites from unauthorized access, the Web App Firewall protects against vulnerabilities in legacy CGI code or scripts, web frameworks, web server software, and other underlying operating systems.

The Citrix Web App Firewall is available as a stand-alone appliance, or as a feature on a Citrix ADC virtual appliance (VPX). In the Web App Firewall documentation, the term Citrix ADC refers to the platform on which the Web App Firewall is running, regardless of whether that platform is a dedicated

firewall appliance, a Citrix ADC on which other features have also been configured, or a Citrix ADC VPX.

To use the Web App Firewall, you must create at least one security configuration to block connections that violate the rules that you set for your protected websites. The number of security configurations that you might want to create depends on the complexity of your website. Sometimes, a single configuration is sufficient. In other cases, particularly those that include interactive websites, websites that access database servers, online stores with shopping carts, you might need several different configurations to best protect sensitive data without wasting significant effort on content that is not vulnerable to certain types of attacks. You can often leave the defaults for the global settings, which affect all security configurations, unchanged. However, you can change the global settings if they conflict with other parts of your configuration or you prefer to customize them.

Web application security

Web application security is network security for computers and programs that communicate by using the HTTP and HTTPS protocols. This is a broad area in which security flaws and weaknesses abound. Operating systems on both servers and clients have security issues and are vulnerable to attack. Web server software and website enabling technologies such as CGI, Java, JavaScript, PERL, and PHP have underlying vulnerabilities. Browsers and other client applications that communicate with web-enabled applications also have vulnerabilities. Websites that use any technology but the simplest of HTML, including any site that allows interaction with visitors, often have vulnerabilities of their own.

In the past, a breach in security was often just an annoyance, but today that is seldom the case. For example, attacks in which a hacker gained access to a web server and made unauthorized modifications to (defaced) a website used to be common. They were usually launched by hackers who had no motivation beyond demonstrating their skills to fellow hackers or embarrassing the targeted person or company. Most current security breaches, however, are motivated by a desire for money. The majority attempt to accomplish one or both of the following goals: to obtain sensitive and potentially valuable private information, or to obtain unauthorized access to and control of a website or web server.

Certain forms of web attacks focus on obtaining private information. These attacks are often possible even against websites that are secure enough to prevent an attacker from taking full control. The information that an attacker can obtain from a website can include customer names, addresses, phone numbers, social security numbers, credit card numbers, medical records, and other private information. The attacker can then use this information or sell it to others. Much of the information obtained by such attacks is protected by law, and all of it by custom and expectation. A breach of this type can have serious consequences for customers whose private information is compromised. At best, these customers have to exercise vigilance to prevent others from abusing their credit cards, opening unauthorized credit accounts in their name, or appropriating their identities outright (identity theft). At

worst, the customers may face ruined credit ratings or even be blamed for criminal activities in which they had no part.

Other web attacks are aimed at obtaining control of (or *compromising*) a website or the server on which it operates, or both. A hacker who gains control of a website or server can use it to host unauthorized content, act as a proxy for content hosted on another web server, provide SMTP services to send unsolicited bulk email, or provide DNS services to support such activities on other compromised web servers. Most websites that are hosted on compromised web servers promote questionable or outright fraudulent businesses. For example, most phishing websites and child exploitation websites are hosted on compromised web servers.

Protecting your websites and web services against these attacks requires a multilayered defense capable of both blocking known attacks with identifiable characteristics and protecting against unknown attacks, which can often be detected because they look different from the normal traffic to your websites and web services.

Known web attacks

The first line of defense for your websites is protection against the large number of attacks that are known to exist and have been observed and analyzed by web security experts. Common types of attacks against HTML-based websites include:

- **Buffer overflow attacks.** Sending a long URL, long cookie, or long information to a web server causes the system to hang, crash, or provide unauthorized access to the underlying operating system. A buffer overflow attack can be used to gain access to unauthorized information, to compromise a web server, or both.
- **Cookie security attacks.** Sending a modified cookie to a web server, usually in hopes of obtaining access to unauthorized content by using falsified credentials.
- **Forceful browsing.** Accessing URLs on a website directly, without navigating to the URLs with hyperlinks on the home page or other common start URLs on the website. Individual instances of forceful browsing might indicate a user who bookmarked a page on your website, but repeated attempts to access nonexistent content, or content that users must never access directly, often represent an attack on website security. Forceful browsing is normally used to gain access to unauthorized information, but can also be combined with a buffer overflow attack in an attempt to compromise your server.
- **Web form security attacks.** Sending inappropriate content to your website in a web form. Inappropriate content can include modified hidden fields, HTML, or code in a field intended for alphanumeric data only, an overly long string in a field that accepts only a short string, an alphanumeric string in a field that accepts only an integer, and a wide variety of other data that your website does not expect to receive in that web form. A web form security attack can be used either to obtain unauthorized information from your website or to compromise the website outright, usually when combined with a buffer overflow attack.

Two specialized types of attacks on web form security deserve special mention:

- **SQL injection attacks.** Sending an active SQL command or commands in a web form or as part of a URL, with the goal of causing an SQL database to run the command or commands. SQL injection attacks are normally used to obtain unauthorized information.
- **Cross-site scripting attacks.** Using a URL or a script on a webpage to violate the same-origin policy, which forbids any script from obtaining properties from or modifying any content on a different website. Since scripts can obtain information and modify files on your website, allowing a script access to content on a different website can provide an attacker the means to obtain unauthorized information, to compromise a web server, or both.

Attacks against XML-based web services normally fall into at least one of the following two categories: attempts to send inappropriate content to a web service, or attempts to breach security on a web service. Common types of attacks against XML-based web services include:

- **Malicious code or objects.** XML requests that contain code or objects that can either directly obtain sensitive information or can give an attacker control of the web service or underlying server.
- **Badly-formed XML requests.** XML requests that do not conform to the W3C XML specification, and that can therefore breach security on an insecure web service
- **Denial of service (DoS) attacks.** XML requests that are sent repeatedly and in high volume, with the intent of overwhelming the targeted web service and denying legitimate users access to the web service.

In addition to standard XML-based attacks, XML web services and Web 2.0 sites are also vulnerable to SQL injection and cross-site scripting attacks, as described below:

- **SQL injection attacks.** Sending an active SQL command or commands in an XML-based request, with the goal of causing an SQL database to run that command or commands. As with HTML SQL injection attacks, XML SQL injection attacks are normally used to obtain unauthorized information.
- **Cross-site scripting attacks.** Using a script included in an XML based application to violate the same-origin policy, which does not allow any script to obtain properties from or modify any content on a different application. Since scripts can obtain information and modify files by using your XML application, allowing a script access to content belonging to a different application can give an attacker the means to obtain unauthorized information, to compromise the application, or both

Known web attacks can usually be stopped by filtering website traffic for specific characteristics (signatures) that always appear for a specific attack and must never appear in legitimate traffic. This approach has the advantages of requiring relatively few resources and posing relatively little risk of false positives. Therefore, it is a valuable tool in fighting attacks on websites and web services, and configuring basic signature protection.

Unknown web attacks

The greatest threat against websites and applications does not come from known attacks, but from unknown attacks. Most unknown attacks fall into one of two categories: newly launched attacks for which security firms have not yet developed an effective defense (zero-day attacks), and carefully targeted attacks on a specific website or web service rather than many websites or web services (spear attacks). These attacks, like known attacks, are intended to obtain sensitive private information, compromise the website or web service and allow it to be used for further attacks, or both of those goals.

Zero-day attacks are a major threat to all users. These attacks are usually of the same types as known attacks; zero-day attacks often involve injected SQL, a cross-site script, a cross-site request forgery, or another type of attack similar to known attacks. Usually, they target vulnerabilities that the developers of the targeted software, website, or web service either are unaware of or have learned about. Security firms have therefore not developed defenses against these attacks, and even if they have, users have not obtained and installed the patches or performed the workarounds necessary to protect against these attacks. The time between discovery of a zero-day attack and availability of a defense (the vulnerability window) is shrinking, but perpetrators can still count on hours or even days in which many websites and web services lack any specific protection against the attack.

Spear attacks are a major threat, but to a more select group of users. A common type of spear attack, a spear phishes, is targeted at customers of a specific bank or financial institution, or (less commonly) at employees of a specific company or organization. Unlike other phishes, which are often crudely written forgeries that a user with any familiarity with the actual communications of that bank or financial institution can recognize, spear phishes are letter perfect and convincing. They can contain information specific to the individual that, at first look, no stranger must know or be able to obtain. The spear phisher is therefore able to convince the target to provide the requested information, which the phisher can then use to loot accounts, to process illegitimately obtained money from other sources, or to gain access to other, even more sensitive information.

Both of these types of attack have certain characteristics that can usually be detected, although not by using static patterns that look for specific characteristics, as do standard signatures. Detecting these types of attacks requires more sophisticated and more resource-intensive approaches, such as heuristic filtering and positive security model systems. Heuristic filtering looks, not for specific patterns, but for patterns of behaviors. Positive security model systems model the normal behavior of the website or web service that they are protecting, and then block connections that do not fit within that model of normal use. URL based and web-form based security checks profile normal use of your websites, and then control how users interact with your websites, using both heuristics and positive security to block anomalous or unexpected traffic. Both heuristic and positive security, properly designed and deployed, can catch most attacks that signatures miss. However, they require considerably more resources than do signatures, and you must spend some time configuring them properly to avoid false positives. They are therefore used, not as the primary line of defense, but as backups to signatures or other less resource-intensive approaches.

By configuring these advanced protections in addition to signatures, you create a hybrid security model, which enables the Web App Firewall to provide comprehensive protection against both known and unknown attacks.

How Citrix Web Application Firewall works

When you install the Web App Firewall, you create an initial security configuration, which consists of a policy, a profile, and a signatures object. The policy is a rule that identifies the traffic to be filtered, and the profile identifies the patterns and types of behavior to allow or block when the traffic is filtered. The simplest patterns, which are called signatures, are not specified within the profile, but in a signatures object that is associated with the profile.

A signature is a string or pattern that matches a known type of attack. The Web App Firewall contains over a thousand signatures in seven categories, each directed at attacks on specific types of web servers and web content. Citrix updates the list with new signatures as new threats are identified. During configuration, you specify the signature categories that are appropriate for the web servers and content that you need to protect. Signatures provide good basic protection with low processing overhead. If your applications have special vulnerabilities or you detect an attack against them for which no signature exists, you can add your own signatures.

The more advanced protections are called security checks. A security check is a more rigorous, algorithmic inspection of a request for specific patterns or types of behavior that might indicate an attack or constitute a threat to your protected websites and web services. It can, for example, identify a request that attempts to perform a certain type of operation that might breach security, or a response that includes sensitive private information such as a social security number or credit card number. During configuration, you specify the security checks that are appropriate for the web servers and content that you need to protect. The security checks are restrictive. Many of them can block legitimate requests and responses if you do not add the appropriate exceptions (relaxations) when configuring them. Identifying the needed exceptions is not difficult if you use the adaptive learning feature, which observes normal use of your website and creates recommended exceptions.

The Web App Firewall can be installed as either a Layer 3 network device or a Layer 2 network bridge between your servers and your users, usually behind your company's router or firewall. It must be installed in a location where it can intercept traffic between the web servers that you want to protect and the hub or switch through which users access those web servers. You then configure the network to send requests to the Web App Firewall instead of directly to your web servers, and responses to the Web App Firewall instead of directly to your users. The Web App Firewall filters that traffic before forwarding it to its final destination, using both its internal rule set and your additions and modifications. It blocks or renders harmless any activity that it detects as harmful, and then forwards the remaining traffic to the web server. The following figure provides an overview of the filtering process.

> **Note:**
>
> The figure omits the application of a policy to incoming traffic. It illustrates a security configuration in which the policy is to process all requests. Also, in this configuration, a signatures object has been configured and associated with the profile, and security checks have been configured in the profile.

Figure 1. A Flowchart of Web App Firewall Filtering



As the figure shows, when a user requests a URL on a protected website, the Web App Firewall first examines the request to ensure that it does not match a signature. If the request matches a signature, the Citrix Web Application Firewall either displays the error object (a webpage located on the Web App Firewall appliance and which you can configure by using the imports feature) or forwards the request to the designated error URL (the error page). Signatures do not require as many resources as do security checks, so detecting and stopping attacks that are detected by a signature before running any of the security checks reduces the load on the server.

If a request passes signature inspection, the Web App Firewall applies the request security checks that have been enabled. The request security checks verify that the request is appropriate for your website or web service and does not contain material that might pose a threat. For example, security checks examine the request for signs indicating that it might be of an unexpected type, request unexpected content, or contain unexpected and possibly malicious web form data, SQL commands, or scripts. If the request fails a security check, the Web App Firewall either sanitizes the request and then sends it back to the Citrix ADC appliance (or Citrix ADC virtual appliance), or displays the error object. If the request passes the security checks, it is sent back to the Citrix ADC appliance, which completes any other processing and forwards the request to the protected web server.

When the website or web service sends a response to the user, the Web App Firewall applies the response security checks that have been enabled. The response security checks examine the response for leaks of sensitive private information, signs of website defacement, or other content that must not be present. If the response fails a security check, the Web App Firewall either removes the content that must not be present or blocks the response. If the response passes the security checks, it is sent back to the Citrix ADC appliance, which forwards it to the user.

Citrix Web Application Firewall features

The basic Web App Firewall features are policies, profiles, and signatures, which provide a hybrid security model as described in Known Web Attacks, Unknown Web Attacks, and How the Web App Firewall Works. Of special note is the learning feature, which observes traffic to your protected applications and recommends appropriate configuration settings for certain security checks.

The imports feature manages files that you upload to the Web App Firewall. These files are then used by the Web App Firewall in various security checks, or when responding to a connection that matches a security check.

You can use the logs, statistics, and reports features to evaluate the performance of the Web App Firewall and identify possible needs for more protections.

How Citrix Web Application Firewall modifies application traffic

The Citrix Web Application Firewall affects the behavior of a web application it protects by modifying the following:

- Cookies
- HTTP Headers
- Forms/Data

Citrix Web Application Firewall session cookie

To maintain the state of the session, Citrix ADC Web App Firewall generates its own session cookie. This cookie is passed only between the web browser and the Citrix ADC Web Application Firewall and not to the web server. If any hacker tries to modify the session cookie, the Application Firewall drops the cookie before forwarding the request to the server and treats the request as a new user session. The session cookie is present as long as the web browser is open. When the web browser is closed, the Application Firewall session cookie becomes longer valid. The state of the session maintains the information of the URLs and forms visited by the client.

The configurable Web App Firewall session cookie is `citrix_ns_id`.

From Citrix ADC build 12.1 54 and 13.0 onwards, the cookie consistency is sessionless and it does not enforce the adding session cookie `citrix_ns_id` generated by the appliance.

Citrix Web App Firewall cookies

Many web applications generate cookies to track user or session specific information. This information can be user preferences or shopping cart items. A web application cookie can be one of the following two types:

- **Persistent Cookies** - These cookies are stored locally on the computer and used again the next time you visit the site. This type of cookie usually contains information about the user, such as, logon, password, or preferences.
- **Session or Transient Cookies** - These cookies are used only during the session and are destroyed after the session is terminated. This type of cookie contains application state information, such as, shopping cart items or session credentials.

Hackers can attempt to modify or steal application cookies to hijack a user session or masquerade as a user. The Application Firewall prevents such attempts by hashing the application cookies and then adding more cookies with the digital signatures. By tracking the cookies, the Application Firewall ensures that the cookies are not modified or compromised between the client browser and the Application Firewall. The Application Firewall does not modify the application cookies.

The Citrix Web Application Firewall generates the following default cookies to track the application cookies:

- **Persistent Cookies**: `citrix_ns_id_wlf`. Note: wlf stands for will live forever.
- **Session or Transient Cookies**: `citrix_ns_id_wat`. Note: wat stands for will act transiently. To track the application cookies, the Application Firewall groups the persistent or session application cookies together and then hash and sign all the cookies together. Thus, the Application Firewall generates one `wlf` cookie to track all persistent application cookies and one `wat` cookie to track all application session cookies.

The following table shows the number and types of cookies generated by the Application Firewall based on the cookies generated by the web application:

Before Citrix ADC Web App Firewall	To
One persistent cookie	Persistent cookie: `citix_ns_id_wlf`
One transient cookie	Transient cookie: `citix_ns_id_wat`
Multiple persistent cookies, Multiple transient cookies	One Persistent cookie: `citrix_ns_id_wlf`, One Transient cookie: `citix_ns_id_wat`

Citrix Web App Firewall allows encrypting the application cookie. Application Firewall also provides an option to proxy the session cookie sent by the application, by storing it with the rest of the Application Firewall session data and not sending it to the client. When a client sends a request to the application that includes an Application Firewall session cookie, Application Firewall inserts the application sent cookie back into the request before sending the request on to the origin application. Application Firewall also allows adding the HTTPOnly and/or Secure flags to cookies.

How the application firewall affects HTTP headers

Both HTTPs requests and HTTPs responses use headers to send information about one or more HTTPs' message. A header is a series of lines with each line containing a name followed by a colon and a space, and a value. For example, the Host header has the following format:

```
Host: www.citrix.com
```

Some header fields are used in both request and response headers, while others are appropriate only for either a request or a response. The Application Firewall might add, modify, or delete some headers in one or more HTTPs request or response to maintain the security of the application.

Request headers dropped by the Citrix Web Application Firewall

Many of the request headers related to caching is dropped to view every request within the context of a session. Similarly, if the request includes an encoding header to allow the web server to send compressed responses, the Application Firewall deletes this header so the contents in the uncompressed server response is inspected by the Web App Firewall to prevent any leakage of sensitive data to the client.

The Application Firewall drops the following request headers:

- Range – Used to recover from a failed or partial file transfers.
- If-Range – Allows a client to retrieve a partial object when it contains a part of that object in its cache already (conditional GET).

- If-Modified-Since – If the requested object is not modified since the time specified in this field, an entity is not returned from the server. You get an HTTP 304 not modified error.
- If-None-Match – Allows efficient updates of cached information with a minimum amount of overhead.
- Accept-Encoding – What encoding methods are allowed for a particular object, such as gzip.

Request header modified by the Citrix Web Application Firewall

If a web browser uses the HTTP/1.0 or earlier protocols, the browser continually opens and closes the TCP socket connection after receiving each response. This adds overhead to the web server and prevents maintaining session state. The HTTP/1.1 protocol allows the connection to remain open during the session. The Application Firewall modifies the following request header to use HTTP/1.1 between the Application Firewall and the web server regardless of the protocol used by the web browser:
Connection: keep-alive

Request headers added by the Citrix Web Application Firewall

The Application Firewall acts as a reverse proxy and replaces the original source IP address of the session with the IP address of the Application Firewall. Therefore, all requests logged in the web server log indicate that the requests are sent from the Application Firewall.

Response header dropped by the Citrix Web Application Firewall

The Application Firewall might block or modify content such as removing credit card numbers or stripping comments, and this might result in a mismatch in size. To prevent such a scenario, the Application Firewall drops the following header:

Content-Length – Indicates the size of the message sent to the recipient.
Response Headers Modified by the Application Firewall

Many of the response headers modified by the Application Firewall are related to caching. Caching headers in HTTP(S) responses must be modified to force the web browser to always send a request to the web server for the latest data and not use the local cache. However, some ASP applications use separate plug-ins to display dynamic contents and might require the ability to cache the data temporarily in the browser. To allow temporary caching of data when Advanced Security protections such as FFC, URL closure, or CSRF checks are enabled, Application Firewall adds or modifies the cache-control headers in the server response using the following logic:

- If Server sends Pragma: no-cache, then the Application Firewall does not do any modification.
- If Client Request is HTTP 1.0, then Application Firewall inserts Pragma: no-cache.
- If Client Request is HTTP 1.1 and has Cache-control: no-store, then Application Firewall does not make any modification.

- If Client Request is HTTP 1.1 and Server Response has Cache-Control header with no store or no cache directive, then Application Firewall does not make any modification.
- If Client Request is HTTP 1.1 and Server Response has either No Cache-control Header or Cache-Control header does not have no store or no-cache directive, the Application Firewall completes the following tasks:

1. Inserts Cache-control: max-age=3, must-revalidate,private.
2. Inserts X-Cache-Control-orig = Original value of Cache-Control Header.
3. Deletes Last-Modified header.
4. Replaces Etag.
5. Inserts X-Expires-Orig=Original value of the Expire Header sent by the server.
6. Modifies the Expires Header and sets the expiration date of the webpage to the past, so it is always picked up again.
7. Modifies Accept-Ranges and sets it to none.

To replace temporarily cached data in the client browser when Application Firewall changes the response such as, for StripComments, X-out/Remove SafeObject, xout or remove Credit Card or URL Transform, Application Firewall takes the following actions:

1. Deletes Last-Modified from server before forwarding to client.
2. Replaces Etag with a value determined by Application Firewall.

Response headers added by the Citrix Web App Firewall

- `Transfer-Encoding`: Chunked. This header streams information back to a client without having to know the total length of the response before sending the response. This header is required because the content-length header is removed.
- `Set-Cookie`: The cookies added by the Application Firewall.
- `Xet-Cookie`: If the session is valid and if the response is not expired in cache, you can serve from cache and do not have to send a new cookie because the session is still valid. In such a scenario, the Set-Cookie is changed to Xet-Cookie. For the web browser.

How form data is affected

The Application Firewall protects against attacks that attempt to modify the content of the original form sent by the server. It can also protect against Cross-site Request forgery attacks. The Application Firewall accomplishes by inserting the hidden form tag as_fid in the page.

Example: `<input type="hidden"name="as_fid"value="VRgWq0I196Jmg/+LOY7C"/>`

The hidden field as_fid is used for field consistency. This field is used by Application Firewall to track all fields of the form including the hidden field name/value pairs and to ensure that none of the fields of the form sent by the server are changed on the client side. The CSRF check also uses this unique

form tag as_fid to ensure that the forms submitted by the user were served to the user in this session and no hacker is attempting to hijack the user session.

Sessionless form check

Application Firewall also offers an option to protect form data using sessionless field consistency. This is useful for applications where the forms might have large number of dynamic hidden fields that lead to high memory allocation per session by the Application Firewall. The sessionless field consistency check is accomplished by inserting another hidden field as_ffc_field for only POST requests or for both GET and POST requests based on the configured setting. The Application Firewall changes the method GET to POST when it forwards the form to the client. The appliance then reverts the method to GET when submitting it back to the server. The as_ffc_field value can be large because it contains the encrypted digest of the form being served. The following is an example of the sessionless form check:

```
1 <input type="hidden" name="as_ffc_field" value="CwAAAAVIGLD/
     luRRi1Wu1rbYrFYargEDcO5xVAxsEnMP1megXuQfiDTGbwk0fpgndMHqfMbzfAFdjwR+
     TOm1oT
2 +u+Svo9+NuloPhtnbkxGtNe7gB/o8GlxEcK9ZkIIVv3oIL/
     nIPSRWJljgpWgafzVx7wtugNwnn8/
     GdnhneLCJTaYU7ScnC6LexJDLisI1xsEeONWt8Zm
3 +vJTa3mTebDY6LVyhDpDQfBgI1XLgfLTexAUzSNWHYyloqPruGYfnRPw+
     DIGf6gGwn1BYLEsRHKNbjJBrKpOJo9JzhEqdtZ1g3bMzEF9PocPvM1Hpvi5T6VB
4 /YFunUFM4f+bD7EAVcugdhovzb71CsSQX5+qcC1B8WjQ==" />
5 <!--NeedCopy-->
```

HTML comment stripping

The Application Firewall also offers an option to strip all HTML comments in the responses before sending them to the client. This affects not only forms, but all response pages. The Application Firewall locates and removes any text embedded between “<!-“ and “->” comment tags. The tags remain to indicate that a comment existed in that location of the HTML source code. Any text embedded within any other HTML or JavaScript tags is ignored.
Some applications might not work correctly if they have JavaScript incorrectly embedded within comment tags. A comparison of the page source code before and after the comments were stripped by Application Firewall can help in identifying if any of the stripped comments had the required JavaScript embedded in them.

Credit card protection

The Application Firewall offers an option to inspect the headers and body of the response and either removes or x-outs the Credit Card numbers before forwarding the response to the client. Currently

Application Firewall offers protection for the following major credit cards: American Express, Diners Club, Discover, JCB, MasterCard, and Visa. The x-out action works independent of the Block action.

Safe object protection

Similar to Credit Card numbers, leakage of other sensitive data can also be prevented by using Application Firewall Safe Object security check to either remove or x-out the sensitive content in the response.

Cross-site scripting transforms action

When the transform is enabled for cross-site scripting, the Web App Firewall changes `"<"into "%26lt;"and ">"into "%26gt;"` in the requests. If the checkRequestHeaders setting in the Web App Firewall is enabled, then the Web App Firewall inspects the Request Headers and transforms these characters in Header and cookies also. The transform action does not block or transform values that were originally sent by the server. There is a set of default attributes and tags for cross-site scripting which the Web App Firewall allows. A default list of denied cross-site scripting patterns is also provided. These can be customized by selecting the signatures object and clicking the **Manage SQL/cross-site scripting Patterns dialogue** in the GUI.

Transforming SQL special characters

Application Firewall has the following default transformation rules for SQL special characters:

From	To	Transformation
’ (single quote that is, %27)	”	Another single quote
\ (backslash that is %5C)	|Another backslash added	
; (semicolon that is %3B)		Dropped

When the transformation of special characters is enabled and the checkRequestHeaders is set to ON, then the transformation of special characters happens in Headers and cookies also.
Note: Some request headers such as User-Agent, Accept-Encoding usually contain semicolons and might be impacted by SQL transformation.

Citrix Web Application Firewall behavior wherein it corrupts the EXPECT header

1. Whenever NetScaler receives an HTTP request with the EXPECT header in it, NetScaler sends the EXPECT: 100 -continue response to client on behalf of the back end server.

2. This behavior is because Application Firewall protections must be run on the entire request before forwarding the request to the server, NetScaler must get the entire request from the client.
3. On receiving a `100 continue` response, the client sends the remaining portion of request that completes the request.
4. NetScaler then runs all the protections and then forwards the request to the server.
5. Now as NetScaler is forwarding the complete request the EXPECT header that came in the initial request becomes obsolete as a result NetScaler corrupts this header and sends it to the server.
6. Server on receiving the request ignores any header which is corrupted.

Configuring the Web App Firewall

September 14, 2021

You can configure the Citrix Web App Firewall (Web App Firewall) by using any of the following methods:

- **Web App Firewall Wizard.** A dialog box consisting of a series of screens that step you through the configuration process.
- **Citrix Web Interface AppExpert Template.** A AppExpert template (a set of configuration settings) that are designed to provide appropriate protection for websites. This AppExpert template contains appropriate Web App Firewall configuration settings for protecting many websites.
- **Citrix ADC GUI.** The web-based configuration interface.
- **Citrix ADC Command Line Interface.** The command line configuration interface.

Citrix recommends that you use the Web App Firewall Wizard. Most users will find it the easiest method to configure the Web App Firewall, and it is designed to prevent mistakes. If you have a new Citrix ADC or VPX that you will use primarily to protect websites, you may find the Web Interface AppExpert template a better option because it provides a good default configuration, not just for the Web App Firewall, but for the entire appliance. Both the GUI and the command line interface are intended for experienced users, primarily to modify an existing configuration or use advanced options.

The Web App Firewall Wizard

The Web App Firewall wizard is a dialog box that consists of several screens that prompt you to configure each part of a simple configuration. The Web App Firewall then creates the appropriate configuration elements from the information that you give it. This is the simplest and, for most purposes, the best way to configure the Web App Firewall.

To use the wizard, connect to the GUI with the browser of your choice. When the connection is established, verify that the Web App Firewall is enabled, and then run the Web App Firewall wizard,

which prompts you for configuration information. You do not have to provide all of the requested information the first time you use the wizard. Instead, you can accept default settings, perform a few relatively straightforward configuration tasks to enable important features, and then allow the Web App Firewall to collect important information to help you complete the configuration.

For example, when the wizard prompts you to specify a rule for selecting the traffic to be processed, you can accept the default, which selects all traffic. When it presents you with a list of signatures, you can enable the appropriate categories of signatures and turn on the collection of statistics for those signatures. For this initial configuration, you can skip the advanced protections (security checks). The wizard automatically creates the appropriate policy, signatures object, and profile (collectively, the security configuration), and binds the policy to global. The Web App Firewall then begins filtering connections to your protected websites, logging any connections that match one or more of the signatures that you enabled and collecting statistics about the connections that each signature matches. After the Web App Firewall processes some traffic, you can run the wizard again and examine the logs and statistics to see if any of the signatures that you have enabled are matching legitimate traffic. After determining which signatures are identifying the traffic that you want to block, you can enable blocking for those signatures. If your website or web service is not complex, does not use SQL, and does not have access to sensitive private information, this basic security configuration will probably provide adequate protection.

You may need additional protection if, for example, your website is dynamic. Content that uses scripts may need protection against cross-site scripting attacks. Web content that uses SQL—such as shopping carts, many blogs, and most content management systems—may need protection against SQL injection attacks. Websites and web services that collect sensitive private information such as social security numbers or credit card numbers may require protection against unintentional exposure of that information. Certain types of web-server or XML-server software may require protection from types of attacks tailored to that software. Another consideration is that specific elements of your websites or web services may require different protection than do other elements. Examining the Web App Firewall logs and statistics can help you identify the additional protections that you might need.

After deciding which advanced protections are needed for your websites and web services, you can run the wizard again to configure those protections. Certain security checks require that you enter exceptions (relaxations) to prevent the check from blocking legitimate traffic. You can do so manually, but it is usually easier to enable the adaptive learning feature and allow it to recommend the necessary relaxation. You can use the wizard as many times as necessary to enhance your basic security configuration and/or create additional security configurations.

The wizard automates some tasks that you would have to perform manually if you did not use the wizard. It automatically creates a policy, a signatures object, and a profile, and assigns them the name that you provided when you were prompted for the name of your configuration. The wizard also adds your advanced-protection settings to the profile, binds the signatures object to the profile, associates the profile with the policy, and puts the policy into effect by binding it to Global.

A few tasks cannot be performed in the wizard. You cannot use the wizard to bind a policy to a bind point other than Global. If you want the profile to apply to only a specific part of your configuration, you must manually configure the binding. You cannot configure the engine settings or certain other global configuration options in the wizard. While you can configure any of the advanced protection settings in the wizard, if you want to modify a specific setting in a single security check, it may be easier to do so on the manual configuration screens in the GUI.

For more information on using the Web App Firewall Wizard, see The Web App Firewall Wizard.

The Citrix Web Interface AppExpert Template

AppExpert Templates are a different and simpler approach to configuring and managing complex enterprise applications. The AppExpert display in the GUI consists of a table. Applications are listed in the left-most column, with the Citrix ADC features that are applicable to that application appearing each in its own column to the right. (In the AppExpert interface, those features that are associated with an application are called *application units*.) In the AppExpert interface, you configure the interesting traffic for each application, and turn on rules for compression, caching, rewrite, filtering, responder and the Web App Firewall, instead of having to configure each feature individually.

The Web Interface AppExpert Template contains rules for the following Web App Firewall signatures and security checks:

- **Deny URL check.** Detects connections to content that is known to pose a security risk, or to any other URLs that you designate.
- **Buffer Overflow check.** Detects attempts to cause a buffer overflow on a protected web server.
- **Cookie Consistency check.** Detects malicious modifications to cookies set by a protected website.
- **Form Field Consistency check.** Detects modifications to the structure of a web form on a protected website.
- **CSRF Form Tagging check.** Detects cross-site request forgery attacks.
- **Field Formats check.** Detects inappropriate information uploaded in web forms on a protected website.
- **HTML SQL Injection check.** Detects attempts to inject unauthorized SQL code.
- **HTML Cross-Site Scripting check.** Detects cross-site scripting attacks.

For information on installing and using an AppExpert Template, see AppExpert Applications and Templates.

The Citrix GUI

The GUI is a web-based interface that provides access to all configuration options for the Web App Firewall feature, including advanced configuration and management options that are not available

from any other configuration tool or interface. Specifically, many advanced Signatures options can be configured only in the GUI. You can review recommendations generated by the learning feature only in the GUI. You can bind policies to a bind point other than Global only in the GUI.

For a description of the GUI, see The Web App Firewall Configuration Interfaces. For more information on using the GUI to configure the Web App Firewall, see Manual Configuration By Using the GUI.

For instructions on configuring the Web App Firewall by using the GUI, see Manual Configuration By Using the GUI. For information on the citrix-adc GUI, see The Web App Firewall Configuration Interfaces.

The Citrix ADC command line interface

The Citrix ADC command line interface is a modified UNIX shell based on the FreeBSD bash shell. To configure the Web App Firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. You can configure most parameters and options for the Web App Firewall by using the NetScaler command line. Exceptions are the signatures feature, many of whose options can be configured only by using the GUI or the Web App Firewall wizard, and the learning feature, whose recommendations can only be reviewed in the GUI.

For instructions on configuring the Web App Firewall by using the Citrix ADC command line, see Manual Configuration By Using the Command Line Interface.

Enable Citrix Web App Firewall

September 14, 2021

Before you can create a security configuration, you must enable the Citrix Web App Firewall feature on the appliance.

Points to remember

- If you are configuring a dedicated Citrix Web App Firewall appliance or upgrading an existing appliance, the feature is already enabled. You do not have to perform either of the procedures described here.
- If you have a new Citrix ADC or VPX, you must enable the Citrix Web App Firewall feature before you configure it.
- If you are upgrading a Citrix ADC or VPX from a previous version, you must first enable the Citrix Web App Firewall feature before you configure it.

> **Note:**
>
> If you are upgrading a Citrix ADC or VPX from a previous version, you might need to update the licenses on your appliance before you enable Citrix Web App Firewall. Check with your Citrix representative or reseller to obtain the correct license.

Enable Citrix Web App Firewall by using the command interface

At the command prompt, type the following command:

```
enable ns feature AppFW
```

Enable the Web App Firewall by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Advanced Features**.
3. In the **Configure Advanced Features** page, select **Citrix Web App Firewall**.
4. Click **OK**.



The Web App Firewall wizard

September 14, 2021

Unlike most wizards, the Citrix Web App Firewall Wizard is designed not just to simplify the initial configuration process, but also to modify previously created configurations and to maintain your Web App Firewall setup. A typical user runs the wizard multiple times, skipping some of the screens each time.

The Web App Firewall Wizard automatically creates profiles, policies, and signatures.

Opening the wizard

To run the Web App Firewall Wizard, open the GUI and follow these steps:

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**. The wizard opens.

For more information about the GUI, see "The Web App Firewall Configuration Interfaces."

The Wizard screens

The Web App Firewall wizard displays the following screens on a tabular page:

1. Specify Name: on this screen, when creating a new security configuration, specify a meaningful name and the appropriate type (HTML, XML or WEB 2.0) for your profile. The default policy and signatures are auto-generated by using the same name.

Profile Name

The name can begin with a letter, number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. Choose a name that makes it easy for others to tell what content your new security configuration protects.

> **Note:**
>
> Because the wizard uses this name for both the policy and the profile, it is limited to 31 characters. Manually created policies can have names up to 127 characters in length.

When modifying an existing configuration, you select Modify Existing Configuration and then, in the Name drop-down list, select the name of the existing configuration that you want to modify.

> **Note:**
>
> Only policies that are bound to global or to a bind point appear in this list; you cannot modify an unbound policy by using the Application Firewall wizard. You must either manually bind it to Global or a bind point, or modify it manually. (For manual modification, in the GUI) **Application Firewall** > **Policies** > **Firewall** pane, select the policy and click **Open**.

Profile Type

You also select a profile type on this screen. The profile type determines the types of advanced protection (security checks) that can be configured. Because certain kinds of content are not vulnerable to certain types of security threats, restricting the list of available checks saves time during configuration. The types of Web App Firewall profiles are:

- Web Application (HTML). Any HTML-based website that does not use XML or Web 2.0 technologies.
- XML Application (XML, SOAP). Any XML-based Web service.
- Web 2.0 Application (HTML, XML, REST). Any Web 2.0 site that combines HTML and XML-based content, such as an ATOM-based site, a blog, an RSS feed, or a wiki.

Note: If you are unsure which type of content is used on your website, you can choose Web 2.0 Application to ensure that you protect all types of web application content.

2. Specify Rule: on this screen, you specify the policy rule (expression) that defines the traffic the current configuration examines. If you create an initial configuration to protect your websites and web services, you can accept the default value, **true**, which selects all web traffic .

If you want this security configuration to examine, not all HTTP traffic that is routed through the appliance, but specific traffic, you can write a policy rule specifying the traffic that you want it to examine. Rules are written in Citrix ADC expressions language, which is a fully functional object-oriented programming language.

Note: In addition to the default expressions syntax, for backward compatibility the Citrix ADC operating system supports the Citrix ADC classic expressions syntax on Citrix ADC Classic and nCore appliances and virtual appliances. Classic expressions are not supported on Citrix ADC Cluster appliances and virtual appliances. Current users who want to migrate their existing configurations to the Citrix ADC cluster must migrate any policies that contain classic expressions to the default expressions syntax.

- For a simple description of using the Citrix ADC expressions syntax to create Web App Firewall rules, and a list of useful rules, see Firewall Policies.
- For a detailed explanation of how to create policy rules in Citrix ADC expressions syntax, see Policies and Expressions.

4. Select Signatures: on this screen, you select the categories of signatures that you want to use to protect your websites and web services.

This is not a mandatory step, and you can skip it if you want to and go to the **Specify Deep Protections** screen. If the Select Signatures screen is skipped, only a profile and associated policies are created, and the signatures are not created.

You can select **Create New Signature** or **Select Existing Signature**.

If you are creating a new security configuration, the signature categories that you select are enabled, and by default they are recorded in a new signatures object. The new signatures object is assigned the same name that you entered on the Specify name screen as the name of the security configuration.

If you have previously configured signatures objects and want to use one of them as the signatures object associated with the security configuration that you are creating, click **Select Existing Signature** and select a signatures object from the Signatures list.

If you are modifying an existing security configuration, you can click Select Existing Signature and assign a different signatures object to the security configuration.

If you click Create New Signature, you can choose the edit mode as **Simple** or **Advanced**.

1. **Specify Signature Protections (Simple mode)**

The simple mode allows for easy configuration of the signature, with a preset list of protection definitions for common applications such as IIS (Internet Information Server), PHP and ActiveX. The default categories in Simple mode are:

- CGI. Protection against attacks on websites that use CGI scripts in any language, including PERL scripts, Unix shell scripts, and Python scripts.
- Cold Fusion. Protection against attacks on websites that use the Adobe Systems® ColdFusion® Web development platform.
- FrontPage. Protection against attacks on websites that use the Microsoft® FrontPage® Web development platform.
- PHP. Protection against attacks on websites that use the PHP open-source Web development scripting language.
- Client side. Protection against attacks on client-side tools used to access your protected websites, such as Microsoft Internet Explorer, Mozilla Firefox, the Opera browser, and the Adobe Acrobat Reader.
- Microsoft IIS. Protection against attacks on websites that run the Microsoft Internet Information Server (IIS)
- Miscellaneous. Protection against attacks on other server-side tools, such as Web servers and database servers.

On this screen, you select the actions associated with the signature categories that you selected on the Select Signatures screen. The actions that you can configure are:

- Block
- Log
- Stats

By default the Log and Stats actions are enabled but not the Block action. To configure actions, click **Settings**. You can change the action settings of all the selected categories by using the **Action** drop-down list.

1. **Specify Signature Protections (Advanced mode)**

The advanced mode allows for more granular control over the signature definitions and provides significantly more information. Use the advanced mode if you want complete control over signature definition.

The contents of this screen are the same as the contents of the Modify Signatures Object dialog box, as described in Configuring or Modifying a Signatures Object. In this screen, you can configure actions either by clicking the **Actions** drop-down list or the actions menu, which appears as a cirle with three dots.

7. Specify Deep Protections: on this screen, you choose the advanced protections (also called security checks or simply checks) that you want to use to protect your websites and web services. Which checks are available depends on the profile type that you chose on the Specify Name screen. All checks are available for Web 2.0 Application profiles.

For more information, see Overview of Security Checksand see Advanced Form Protections Checks.

You configure the actions for the advanced protections that you have enabled.The actions that you can configure are:

- Block: blocks connections that match the signature. Disabled by default.
- Log: logs connections that match the signature for later analysis. Enabled by default.
- Stats: maintains statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.
- Learn. Observe traffic to this website or web service, and use connections that repeatedly violate this check to generate recommended exceptions to the check, or new rules for the check. Available only for some checks. For more information about the learning feature see Configuring and Using the Learning Feature, and how learning works and how to configure exceptions (relaxations) or deploy learned rules for a check, see Manual Configuration By Using the GUI.

To configure actions, select the protection by clicking the check box, and then click **Action Settings** to select the required actions. Select other parameters, if required, and then click **OK** to close the Action Settings window.

To view all logs for a specific check, select that check, and then click **Logs** to display the Syslog Viewer, as described in Web App Firewall Logs. If a security check is blocking legitimate access to your protected website or web service, you can create and implement a relaxation for that security check by selecting a log that shows the unwanted blocking, and then clicking **Deploy**.

After you completing specifying Action Settings, click **Finish** to complete the wizard.

Following are four procedures that show how to perform specific types of configuration by using the Web App Firewall wizard.

Create a new configuration

Follow these steps to create a new firewall configuration and signature objects, by using the Applicaiton Firewall wizard.

1. Navigate to **Security** > **Application Firewall**.

2. In the details pane, under **Getting Started**, click **Application Firewall. The wizard opens.



3. On the **Specify Name** screen, select **Create New Configuration.
4. In the **Name** field, type a name, and then click **Next**.
5. In the **Specify Rule** screen, click **Next** again.
6. In the **Select Signatures** screen, select **Create New Signature** and **Simple** as the edit mode, and then click **Next**.
7. In the **Specify Signature Protections** screen, configure the required settings. For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see Signatures.
8. In the **Specify Deep Protections** screen configure the required actions and parameters in **Action Settings**.
9. When you complete, click **Finish** to close the Application Firewall wizard.

Modify an existing configuration

Follow these steps to modify an existing configuration and existing signature categories.

1. Navigate to **Security** > **Application Firewall**.

2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**. The wizard opens.
3. On the **Specify Name** screen, select Modify Existing Configuration and, in the **Name** drop-down list, choose the security configuration that you created during new configuration, and then click **Next**.
4. In the **Specify Rule** screen, click Next to keep the default value "true." If you want to modify the rule, follow the steps described in Configure a Custom Policy Expresssion.
5. In the **Select Signatures** screen, click **Select Existing Signature**. From the **Existing Signature** drop-down list, select the appropriate option, and then click **Next**. The advanced signature protection screen appears.
   Note: If you select an existing signature, the default edit mode for signature protected is advanced.
6. In the Specify Signature Protections screen, configure the required settings and click **Next**. For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see Signatures.
7. In the **Specify Deep Protections** screen, configure the settings and click **Next**.
8. After you complete, click **Finish** to close the **Web App Firewall Wizard**.

Create a new configuration without signatures

Follow these steps to use the Application Firewall Wizard to skip the Select Signatures screen and create a new configuration with just the profile and the associated policies but without any signatures.

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**. The wizard opens.
3. On the **Specify Name** screen, select **Create New Configuration**.
4. In the **Name** field, type a name, and then click **Next**.
5. In the **Specify Rule** screen, click **Nex**t again.
6. In the **Select Signatures** screen, click **Skip.**
7. In the **Specify Deep Protections** screen configure the required actions and parameters in **Action Settings**.
8. When you complete, click **Finish** to close the Application Firewall Wizard.

Configure a custom policy expression

Follow these steps to use the Application Firewall Wizard to create a specialized security configuration to protect only specific content. In this case, you create a new security configuration instead of modifying the initial configuration. This type of security configuration requires a custom rule, so that the policy applies the configuration to only the selected Web traffic.

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.
3. On the Specify Name screen, type a name for your new security configuration in the Name text box, select the type of security configuration from the Type drop-down list, and then click **Next**.
4. On the **Specify Rule** screen, enter a rule that matches only that content that you want this web application to protect. Use the **Frequently Used Expressions** drop-down list and the **Expression Editor** to create a custom expression. When you complete, click **Next**.
5. In the **Select Signatures** screen, select the edit mode, and then click **Next**.
6. In the **Specify Signature Protections** screen, configure the required settings.
7. In the **Specify Deep Protections** screen configure the required actions and parameters in **Action Settings**.
8. When you complete, click **Finish** to close the **Application Firewall Wizard**.

Manual configuration

September 14, 2021

If you want to bind a profile to a bind point other than Global, you must manually configure the binding. Also, certain security checks require that you either manually enter the necessary exceptions or enable the learning feature to generate the exceptions that your websites and Web services need. Some of these tasks cannot be performed by using the Web App Firewall wizard.

If you are familiar with how the Web App Firewall works and prefer manual configuration, you can manually configure a signatures object and a profile, associate the signatures object with the profile, create a policy with a rule that matches the web traffic that you want to configure, and associate the policy with the profile. You then bind the policy to Global, or to a bind point, to put it into effect, and you have created a complete security configuration.

For manual configuration, you can use the GUI (a graphical interface) or the command line. Citrix recommends that you use the GUI. Not all configuration tasks can be performed at the command line. Certain tasks, such as enabling signatures and reviewing learned data, must be done in the GUI. Most other tasks are easier to perform in the GUI.

Replicating configuration

When you use the GUI (GUI) or the command line interface (CLI) to manually configure the Web App Firewall, the configuration is saved in the /nsconfig/ns.conf file. You can use the commands in that file to replicate the configuration on another appliance. You can cut and paste the commands into the CLI one by one, or you can save multiple commands in a text file in the /var/tmp folder and run them

as a batch file. Following is an example of running a batch file containing commands copied from the /nsconfig/ns.conf file of a different appliance:

```
> batch -f /var/tmp/appfw_add.txt
```

> **Warning:**
>
> Import commands are not saved in the ns.conf file. Before running commands from the ns.conf file to replicate the configuration on another appliance, you must import all the objects used in the configuration (for example, signatures, error page, WSDL, and Schema) to the appliance on which you replicate the configuration. The add command to add a Web App Firewall profile saved in an ns.conf file might include the name of an imported object, but such a command might fail when run on another appliance if the referenced object does not exist on that appliance.

For more information on import or export details for replicating configuration, see Signature export, and Common import export topics.

Manual configuration by using the Citrix ADC GUI

September 14, 2021

If you need to manually configure the Web App Firewall feature, Citrix recommends you to use the Citrix ADC GUI procedure.

To create and configure signatures object

Before you can configure the signatures, you must create a signatures object from the appropriate default signatures object template. Assign the copy a new name, and then configure the copy. You cannot configure or modify the default signatures objects directly. The following procedure provides basic instructions for configuring a signatures object. For more detailed instructions, see Manually Configuring the Signatures Feature.

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to use as a template, and then click **Add**.

   Your choices are:

   - **Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - **XPath Injection.** Contains all of the items in the Default Signatures, and in addition, contains the XPath injection rules.

3. In the **Add Signatures Object** dialog box, type a name for your new signatures object, click OK, and then click **Close**. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
4. Select the signatures object that you created, and then click **Open**.
5. In the **Modify Signatures Object** dialog box, set the **Display Filter Criteria** options at the left to display the filter items that you want to configure.

   As you modify these options, the results that you specify are displayed in the Filtered Results window at the right. For more information about the categories of signatures, see Signatures.
6. In the **Filtered Results** area, configure the settings for a signature by selecting and clearing the appropriate check boxes.
7. When finished, finished, click **Close**.

To create a Web App Firewall profile by using the GUI

Creating a Web App Firewall profile requires that you specify only a few configuration details.

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, click **Add**.
3. In the **Create Web App Firewall** Profile dialog box, type a name for your profile.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
4. Choose the profile type from the drop-down list.
5. Click **Create**, and then click **Close**.

To configure a Web App Firewall profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**.
3. In the **Configure Web App Firewall Profile** dialog box, on the **Security Checks** tab, configure the security checks.
   - To enable or disable an action for a check, in the list, select or clear the check box for that action.

- To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.

  You can also select a check and, at the bottom of the dialog box, click Open to display the Configure Relaxation dialog box or Configure Rule dialog box for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations or user-defined rules, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click Open to modify it.

- To review learned exceptions or rules for a check, select the check, and then click Learned Violations. In the Manage Learned Rules dialog box, select each learned exception or rule in turn.
  - To edit the exception or rule, and then add it to the list, click **Edit & Deploy**.
  - To accept the exception or rule without modification, click **Deploy**.
  - To remove the exception or rule from the list, click **Skip**.
- To refresh the list of exceptions or rules to be reviewed, click **Refresh**.
- To open the **Learning Visualizer** and use it to review learned rules, click **Visualizer**.
- To review the log entries for connections that matched a check, select the check, and then click **Logs**. You can use this information to determine which checks are matching attacks so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see Logs, Statistics, and Reports.
- To completely disable a check, in the list, clear all of the check boxes to the right of that check.

4. On the **Settings** tab, configure the profile settings.
   - To associate the profile with the set of signatures that you previously created and configured, under **Common Settings**, choose that set of signatures in the Signatures drop-down list.

     > **Note:**
     >
     > You may must use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.

   - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.

> **Note:**
>
> You must first upload the error object that you want to use in the Import pane.

- To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types.

5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile. For more information, see Configure and Learning feature.
6. Click **OK** to save your changes and return to the Profiles pane.

Configuring a Web App Firewall rule or relaxation

You configure two different types of information in this dialog box, depending upon which security check you are configuring. In most cases, you configure an exception (or relaxation) to the security check. If you are configuring the Deny URL check or the Field Formats check, you configure an addition (or rule). The process for either of these is the same.

To configure a relaxation rule by using the Citrix ADC GUI

1. Navigate to **Security** > **Citrix Web App Firewall** > **Profiles**.
2. In the **Profiles** pane, select the profile you want to configure, and then click **Edit**.
3. In the **Configure Web App Firewall Profile** page, click **Relaxation Rule** from **Advanced Settings** section. The **Relaxation Rule** section contains the complete list of Web App Firewall relaxation rules.
4. Click a security rule that you want to configure, and then click **Edit**.
5. The URL Relaxation Rules page contains a list of actions and that you can configure for this rule and a list of existing relaxations or rules. The list might be empty if you have not either manually added any relaxations or approved any relaxations that were recommended by the learning engine. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable the relaxations on the list.
6. To add or modify a relaxation or a rule, do one of the following:

   - To add a new relaxation, click **Add**.
   - To modify an existing relaxation, select the relaxation that you want to modify, and then click **Open**.

   The **Start URL Relaxation Rule** page is displayed. Except for the title, these dialog boxes are identical.

7. Fill in the dialog box as described below. The dialog boxes for each check are different. The list below covers all elements that might appear in any dialog box.

   - **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
   - **Attachment Content Type**—The Content-Type attribute of an XML attachment. In the text area, enter a regular expression that matches the Content-Type attribute of the XML attachments to allow.
   - **Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
   - **Cookie**—In the text area, enter a PCRE-format regular expression that defines the cookie.
   - **Field Name**—A web form field name element may be labeled Field Name, Form Field, or another similar name. In the text area, enter a PCRE-format regular expression that defines the name of the form field.
   - **From Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the web form.
   - **From Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
   - **Name**—An XML element or attribute name. In the text area, enter a PCRE-format regular expression that defines the name of the element or attribute.
   - **URL**—A URL element may be labeled Action URL, Deny URL, Form Action URL, Form Origin URL, Start URL, or simply URL. In the text area, enter a PCRE-format regular expression that defines the URL.
   - **Format**—The format section contains multiple settings that include list boxes and text boxes. Any of the following can appear:
     - **Type**—Select a field type in the Type drop-down list. To add a new field type definition, click Manage—
     - **Minimum Length**—Type a positive integer that represents the minimum length in characters if you want to force users to fill in this field. Default: 0 (Allows field to be left blank.)
     - **Maximum length**—To limit the length of data in this field, type a positive integer that represents the maximum length in characters. Default: 65535
   - **Location**—Choose the element of the request that your relaxation applies to from the drop-down list. For HTML security checks, the choices are:
     - FORMFIELD—Form fields in web forms.
     - HEADER—Request headers.

- COOKIE—Set-Cookie headers.

For XML security checks, the choices are:

- ELEMENT—XML element.
- ATTRIBUE—XML attribute.

- **Maximum Attachment Size**—The maximum size in bytes allowed for an XML attachment.
- **Comments**—In the text area, type a comment. Optional.

Note: For any element that requires a regular expression, you can type the regular expression, use the Regex Tokens menu to insert regular expression elements and symbols directly into the text box, or click **Regex Editor** to open the **Add Regular Expression** dialog box, and use it to construct the expression.

8. To remove a relaxation or rule, select it, and then click **Delete**.
9. To enable a relaxation or rule, select it, and then click **Enable**.
10. To disable a relaxation or rule, select it, and then click **Disable**.
11. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click **Visualizer**, and use the display tools.

> **Note:**
>
> The **Visualizer** button does not appear on all check relaxation dialog boxes.

12. To review learned rules for this check, click Learning and perform the steps in To configure and use the Learning feature
13. Click **OK**.

To configure the Learned Rules by using the Citrix ADC GUI

1. Navigate to **Security** > **Citrix Web App Firewall** > **Profiles**.
2. In the **Profiles** pane, select the profile, and then click **Edit**.
3. In the **Citrix Web App Firewall Profile** page, click **Learned Rules** from **Advanced Settings**. In the **Learned Rules** section you can see a list of security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and click **Settings**.
5. In the **Dynamic Profiling and Learning Rules Settings** page, you can set the settings. For more information, see Dynamic profile settings

   - **Minimum number threshold.** Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of

total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1

- **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0

6. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, select **Remove All Learned Data** action.

   > **Note:**
   >
   > This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.

7. To restrict the learning engine to traffic from a specific set of IPs, click **Trusted Learning Clients**, and add the IP addresses that you want to use to the list.
   a) To add an IP address or IP address range to the Trusted Learning Clients list, click **Add**.
   b) In the **AppFirewall Profile to Trusted Clint Binding** page, click **Add**.
   c) Select the **Enabled** check box to enable the feature.
   d) In Trusted Learning Client** box, type the IP address or an IP address range in CIDR format.
   e) In the **Comments** text area, type a comment that describes this IP address or range.
   f) Click **Create** and **Close**.
8. To modify an existing IP address or range, click the IP address or range, and then click **Edit**. Except for the name, the dialog box that appears is identical to the Add Trusted Learning Clients dialog box.
9. To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click **Disable** or **Enable**, as appropriate.
10. To remove an IP address or range completely, click the IP address or range, and then click **Delete**.
11. Click **Close** to return to the **Citrix Web App Firewall Profile** page.

To create a Citrix Web App Firewall policy by using the Citrix ADC GUI

1. Navigate to **Security** > **Citrix Web App firewall** > **Policies**.
2. In the **Policies** page, click **Citrix Web App Firewall Policy** link.

3. In the Citrix Web App Firewall Policies page, click **Add**.
4. In the Create Citrix Web App Firewall Policy page, set the following parameters.
   a) Name. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
   b) Profile. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a profile to associate with your policy by clicking New, and you can modify an existing profile by clicking **Modify**.
   c) Expression. In the Expression text area, create a rule for your policy.
   d) Log Action. Add a log action or you can modify an existing log action.
   e) Comments. A brief description about the policy.
5. Click **Create** or **OK**, and then click **Close**.



To create or configure a Web App Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the Web App Firewall filters by using the profile associated with the policy. Like other Citrix ADC policy rules (or *expressions*), Web App Firewall rules use Citrix ADC expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the Web App Firewall wizard or the Citrix ADC GUI to create your policy rule:
   - If you are configuring a policy in the Web App Firewall wizard, in the navigation pane, click

Citrix Web App Firewall Wizard, then in the details pane click **Citrix Web App Firewall Wizard**, and then navigate to the **Specify Rule** tab page.

- In the **Specify Rule** page, choose the prefix for your expression from the drop-down list. Your choices are:
- **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
- **SYS.** One or more protected websites. Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
- **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
- **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

After you choose a prefix, the Web App Firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.

2. Choose your next term.

   If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The Web App Firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

   When you have decided which term you want, double-click it to insert it into the Expression window.

3. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(“”), type the header name between the quotation marks.

4. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

   Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

For shopping.example.com, substitute the name of the web host that you want to match.

- **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:


```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
```

For www.example.com, substitute the name of the web host. For folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

- **Specific type of content: GIF images.** To match GIF format images:

```
HTTP.REQ.URL.ENDSWITH(".png")
```

To match other format images, substitute another string in place of .png.

- **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
```

To match all JavaScripts with .js extensions:

```
HTTP.REQ.URL.ENDSWITH(".js")
```

For more information about creating policy expressions, see Policies and Expressions.

> **Note:**
>
> If you use the command line to configure a policy, remember to escape any double quotation marks within Citrix ADC expressions. For example, the following expression is correct if entered in the GUI:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

If entered at the command line, however, you must type this instead:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

```
1 ![Policy expression configuration](/en-us/citrix-adc/media/waf-rule.png
     )
```

To add a firewall rule (expression) by using the Add Expression dialog box

The **Add Expression** dialog box (also referred to as the Expression Editor) helps users who are not familiar with the Citrix ADC expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the Web App Firewall wizard or the Citrix ADC GUI:
   - If you are configuring a policy in the **Web App Firewall** wizard, in the navigation pane, click **Web App Firewall**, then in the details pane click **Web App Firewall** Wizard, and then navigate to the **Specify Rule** screen.

- If you are configuring a policy manually, in the navigation pane, expand **Web App Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.

2. On the **Specify Rule** screen, in the **Create Web App Firewall Profile** dialog box, or in the **Configure Web App Firewall Profile** dialog box, click **Add**.
3. In the **Add Expression** dialog box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
   - **SYS.** One or more protected websites. Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the Web App Firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the Preview Expression window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The Preview Expression window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or more terms that you added after the term that you modified is cleared.
7. When you have finished constructing your expression, click OK to close the Add Expression dialog box. Your expression is inserted into the Expression text area.

To bind a Web App Firewall policy by using the Citrix ADC GUI

1. Do one of the following:
   - Navigate to **Security** > **Web App Firewall**, and in the details pane, click **application firewall policy manager**.
   - Navigate to **Security** > **Citrix Web App Firewall** > **Policies** > **Firewall**, and in the “Citrix Web App Firewall Policies” pane, click **Policy Manager**.

2. In the **Application Firewall Policy Manager** dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
    - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the Citrix ADC appliance, and are applied before any other policies.
    - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
    - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
    - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the Citrix ADC appliance.
    - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
    - **None.** Do not bind the policy to any bind point.
3. Click **Continue**. A list of existing Web App Firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
    - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select **Regenerate Priorities** to renumber the priorities evenly.
    - To modify the policy expression, double-click that field to open the **Configure Web App Firewall Policy** dialog box, where you can edit the policy expression.
    - To set the Goto Expression, double-click field in the **Goto Expression** column heading to display the drop-down list, where you can choose an expression.
    - To set the Invoke option, double-click field in the Invoke column heading to display the drop-down list, where you can choose an expression.
6. Repeat steps 3 through 6 to add any additional Web App Firewall policies you want to globally bind.
7. Click **OK**. A message appears in the status bar, stating that the policy has been successfully bound.

Manual configuration By using the command line interface

September 14, 2021

> **Note:**
>
> If you need to manually configure the Web App Firewall feature, Citrix recommends you to use the Citrix ADC GUI procedure.

You can configure the Web App Firewall features from the **Citrix ADC** command interface. However, there are important exceptions. You cannot enable signatures from the command interface. There are around 1,000 default signatures in seven categories and the task is too complex for the command interface. You can enable or disable features and configure parameters from the command line, but cannot configure manual relaxations. While you can configure the adaptive learning feature and enable learning from the command line, you cannot review learned relaxations or learned rules and approve or skip them. The command line interface is intended for advanced users who are familiar in using the Citrix ADC appliance and Web App Firewall.

To manually configure the Web App Firewall by using the Citrix ADC command line, use a telnet or secure shell client of your choice to log on to the Citrix ADC command line.

To create a profile by using the command line interface

At the command prompt, type the following commands:

- `add appfw profile <name> [-defaults ( basic | advanced )]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `save ns config`

Example

The following example adds a profile named pr-basic, with basic defaults, and assigns a profile type of HTML. This is the appropriate initial configuration for a profile to protect an HTML website.

```
1 add appfw profile pr-basic -defaults basic
2 set appfw profile pr-basic -type HTML
3 save ns config
4 <!--NeedCopy-->
```

To configure a profile by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> <arg1> [<arg2> ...]` where `<arg1>` represents a parameter and `<arg2>` represents either another parameter or the value to assign to the parameter represented by `<arg1>`. For descriptions of the parameters to use when configuring specific

security checks, see Advanced Protections and its subtopics. For descriptions of the other parameters, see "Parameters for Creating a Profile."

- `save ns config`

Example

The following example shows how to configure an HTML profile created with basic defaults to begin protecting a simple HTML-based website. This example turns on logging and maintenance of statistics for most security checks, but enables blocking only for those checks that have low false positive rates and require no special configuration. It also turns on transformation of unsafe HTML and unsafe SQL, which prevents attacks but does not block requests to your websites. With logging and statistics enabled, you can later review the logs to determine whether to enable blocking for a specific security check.

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
<!--NeedCopy-->
```

To create and configure a policy

At the command prompt, type the following commands:

- `add appfw policy <name> <rule> <profile>`
- `save ns config`

Example

The following example adds a policy named pl-blog, with a rule that intercepts all traffic to or from the host blog.example.com, and associates that policy with the profile pr-blog.


```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
<!--NeedCopy-->
```

To bind a Web App Firewall policy

At the command prompt, type the following commands:

- `bind appfw global <policyName> <priority>`
- `save ns config`

Example

The following example binds the policy named pl-blog and assigns it a priority of 10.

```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

To configure session limit per PE

At the command prompt, type the following commands:

- `set appfw settings <session limit>`

Example

The following example configures the session limit per PE.

```
> set appfw settings -sessionLimit 500000`

Done

Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

Signatures

September 14, 2021

The Web App Firewall signatures provide specific, configurable rules to simplify the task of protecting your websites against known attacks. A signature represents a pattern that is a component of a known attack on an operating system, web server, website, XML-based web service, or other resource. A rich set of preconfigured Web App Firewall built-in or native rules offers an easy to use security solution, applying the power of pattern matching to detect attacks and protect against application vulnerabilities.

You can create your own signatures or use signatures in the built-in templates. The Web App Firewall has two built-in templates:

- **Default Signatures**: This template contains a preconfigured list of over 1,300 signatures, in addition to a complete list of SQL injection keywords, SQL special strings, SQL transform rules, and SQL wildchar characters. It also contains denied patterns for cross-site scripting, and allowed attributes and tags for cross-site scripting. This is a read-only template. You can view the contents, but you cannot add, edit, or delete anything in this template. To use it, you must make a copy. In your own copy, you can enable the signature rules that you want to apply to your traffic, and specify the actions to be taken when the signature rules match the traffic.

The Web App Firewall signatures are derived from the rules published by Snort, which is an open source intrusion prevention system capable of performing real-time traffic analysis to detect various attacks and probes.

- ***Xpath Injection Patterns**: This template contains a preconfigured set of literal and PCRE keywords and special strings that are used to detect XPath (XML Path Language) injection attacks.

Blank Signatures: In addition to making a copy of the built-in *Default Signatures template, you can use a blank signatures template to create a signature object. The signature object that you create with the blank signatures option does not have any native signature rules, but, just like the *Default template, it has all the SQL/cross-site scripting built-in entities.

External-Format Signatures: The Web App Firewall also supports external format signatures. You can import the third party scan report by using the XSLT files that are supported by the Citrix Web App Firewall. A set of built-in XSLT files is available for the following scan tools to translate external format files to native format:

- Cenzic
- Deep Security for Web Apps
- IBM AppScan Enterprise
- IBM AppScan Standard.
- Qualys
- Qualys Cloud
- Whitehat
- Hewlett Packard Enterprise WebInspect
- Rapid7 Appspider

- Acunetix

Security protection for your application

Tighter security increases processing overhead. Signatures provide the following deployment options to help you to optimize the protection of your applications:

- **Negative Security Model**: With the negative security model, you use a rich set of preconfigured signature rules to apply the power of pattern matching to detect attacks and protect against application vulnerabilities. You block only what you don't want and allow the rest. You can add your own signature rules, based on the specific security needs of your applications, to design your own customized security solutions.
- **Hybrid security Model**: In addition to using signatures, you can use positive security checks to create a configuration ideally suited for your applications. Use signatures to block what you don't want, and use positive security checks to enforce what is allowed.

To protect your application by using signatures, you must configure one or more profiles to use your signatures object. In a hybrid security configuration, the SQL injection and cross-site scripting patterns, and the SQL transformation rules, in your signatures object are used not only by the signature rules, but also by the positive security checks configured in the Web App Firewall profile that is using the signatures object.

The Web App Firewall examines the traffic to your protected websites and web services to detect traffic that matches a signature. A match is triggered only when every pattern in the rule matches the traffic. When a match occurs, the specified actions for the rule are invoked. You can display an error page or error object when a request is blocked. Log messages can help you to identify attacks being launched against your application. If you enable statistics, the Web App Firewall maintains data about requests that match a Web App Firewall signature or security check.

If the traffic matches both a signature and a positive security check, the more restrictive of the two actions are enforced. For example, if a request matches a signature rule for which the block action is disabled, but the request also matches an SQL Injection positive security check for which the action is block, the request is blocked. In this case, the signature violation might be logged as <`not blocked` >, although the request is blocked by the SQL injection check.

Customization: If necessary, you can add your own rules to a signatures object. You can also customize the SQL/cross-site scripting patterns. The option to add your own signature rules, based on the specific security needs of your applications, gives you the flexibility to design your own customized security solutions. You block only what you don't want and allow the rest. A specific fast-match pattern in a specified location can significantly reduce processing overhead to optimize performance. You can add, modify, or remove SQL injection and cross-site scripting patterns. Built-in RegEx and expression editors help you configure your patterns and verify their accuracy.

Auto-update: You can manually update the signature object to get the latest signature rules, or you can apply the auto-update feature so that the Web App Firewall can automatically update the signatures from the cloud-based Web App Firewall updates service.

> **Note:**
>
> If new signature rules are added during auto-update, they are disabled by default. You must periodically review the updated signatures and enable the newly added rules that are pertinent for protecting your applications.
>
> You must configure CORS to host signatures on IIS servers.
>
> Signature auto update feature does not work on the local web server when you access the URL from the Citrix ADC GUI.

Getting started

Using Citrix signatures to protect your application is easy and can be accomplished in a few simple steps:

1. Add a signature object.

- You can use the Wizard that prompts you to create the entire Web App Firewall configuration, including adding the profile and policy, selecting and enabling signatures, and specifying actions for signatures and positive security checks. The signatures object is created automatically.
- You can create a copy of the signatures object from the *Default Signatures template, use a blank template to create a signature with your own customized rules, or add an external format signature. Enable the rules and configure the actions that you want to apply.

1. Configure the target Web App Firewall profile to use this signatures object.
2. Send traffic to validate the functionality

Highlights

- The Default signatures object is a template. It cannot be edited or deleted. To use it, you must create a copy. In your own copy, you can enable the rules and the desired action for each rule as required for your application. To protect the application, you must configure the target profile to use this signature.
- Processing signature patterns has overhead. Try to enable only those signatures that are applicable for protecting your application, rather than enabling all signature rules.
- Every pattern in the rule must match to trigger a signature match.
- You can add your own customized rules to inspect incoming requests to detect various types of attacks, such as SQL injection or cross-site scripting attacks. You can also add rules to inspect the responses to detect and block leakage of sensitive information such as credit card numbers.

- You can make a copy of an existing signature object and tweak it, by adding or editing rules and SQL/cross-site scripting patterns, to protect another application.
- You can use auto-update to download the latest version of the Web App Firewall default rules without need for ongoing monitoring to check for the availability of the new update.
- A signature object can be used by more than one profile. Even after you have configured one or more profiles to use a signature object, you can still enable or disable signatures or change the action settings. You can manually create and modify your own custom signature rules. The changes apply to all the profiles that are currently configured to use this signature object.
- You can configure signatures to detect violations in various types of payloads, such as HTML, XML, JSON, and GWT.
- You can export a configured signatures object and import it to another Citrix ADC appliance for easy replication of your customized signature rules.

Signatures are patterns that are associated with a known vulnerability. You can use signature protection to identify the traffic that attempts to exploit these vulnerabilities, and take specific actions.

Signatures are organized into categories. You can optimize the performance and reduce the processing overhead by enabling only the rules in the categories that are appropriate for protecting your application.

Manually configuring the signatures feature

September 14, 2021

To use signatures to protect your websites, you must review the rules, and enable and configure the ones that you want to apply. The rules are disabled by default. Citrix recommends that you enable all rules that are applicable to the type of content that your website uses.

To manually configure the signatures feature, use a browser to connect to the GUI. Then, create a signatures object from a built-in template, an existing signatures object, or by importing a file. Next, configure the new signatures object as explained in Configuring or Modifying a Signatures Object.

Adding or removing a signature object

September 14, 2021

You can add a new signature object to the Web App Firewall by:

- Copying a built-in template.
- Copying an existing signatures object.

- Importing a signatures object from an external file.

The signature file includes CPU usage, latest applicable year, and severity level details. You can see the CPU usage, latest year, and CVE severity level every time a signature file is modified and uploaded periodically. After observing these values, you can decide to enable or disable the signature on the appliance.

You must use the GUI to copy a template or existing signatures object. You can use either the GUI or the command line to import a signatures object. You can also use either the GUI or the command line to remove a signatures object.

To create a signatures object from a template

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to use as a template.

   Your choices are:

   - **Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - **XPath Injection.** Contains the XPath injection patterns.
   - **Any existing signatures object.**

   > **Attention:**
   >
   > If you do not choose a signatures type to use as a template, the Web App Firewall prompts you to create signatures from scratch.

3. Click **Add**.
4. In the Add Signatures Object dialog box, type a name for your new signatures object, and then click OK. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
5. Click **Close**.

To create a signatures object by importing a file

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, click **Add**.
3. In the **Add Signatures Object** dialog box, select the format of the signatures you want to import.
   - To import a Citrix ADC format signatures file, select the **Native Format** tab.
   - To import an external signatures format file, select the **External Format** tab.
4. Choose the file that you want to use to create your signatures object.

- To import a native Citrix ADC format signatures file, in the Import section select either Import from Local File or Import from URL, then type or browse to the path or URL to the file.
- To import a Cenzic, IBM AppScan, Qualys, or Whitehat format file, in the XSLT section select Use Built-in XSLT File, Use Local File, or Reference from URL. Next, if you chose Use Built-in XSLT File, select the appropriate file format from the list. If you chose Use Local File or Reference from URL, then type or browse to the path or URL to the file.

5. Click **Add**, and then click **Close**.

To create a signatures object by importing a file by using the command line

At the command prompt, type the following commands:

- `import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]`
- `save ns config`

Example #1

The following example creates a signatures object from a file named signatures.xml and assigns it the name MySignatures.

```
1 import appfw signatures signatures.xml MySignatures
2 save ns config
3 <!--NeedCopy-->
```

To remove a signatures object by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to remove.
3. Click **Remove**.

To remove a signatures object by using the command line

At the command prompt, type the following commands:

- `rm appfw signatures <name>`
- `save ns config`

Configuring or modifying a signatures object

September 14, 2021

You configure a signatures object after creating it, or modify an existing signatures object, to enable or disable signature categories or specific signatures, and configure how the Web App Firewall responds when a signature matches a connection.

To configure or modify a signatures object

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, set the **Display Filter Criteria** options at the left to display the filter items that you want to configure.

   As you modify these options, the results that you requested are displayed in the Filtered Results window at the right.

   - To display only selected categories of signatures, check or clear the appropriate signature-category check boxes. The signature categories are:

Name	Type of Attack that this Signature Protects Against
cgi	CGI scripts. Includes Perl and UNIX shell scripts.
client	Browsers and other clients.
coldfusion	websites that use the Adobe Systems ColdFusion application server.
frontpage	websites that use Microsoft's FrontPage server.
iis	websites that use the Microsoft Internet Information Server (IIS).
misc	Miscellaneous attacks.
php	websites that use PHP
web-activex	websites that contain ActiveX controls.
web-struts	websites that contain Apache struts, which are java-ee based applets.

- To display only signatures that have specific check actions enabled, select the ON check box for each of those actions, clear the ON check boxes for the other actions, and clear all of the OFF check boxes. To display only signatures that have a specific check action disabled, select their respective OFF check boxes and clear all of the ON check boxes. To display signatures regardless of whether they have a check action enabled or disabled, select or clear both the ON and the OFF check boxes for that action. The check actions are:

Criterion	Description
Enabled	The signature is enabled. The Web App Firewall checks only for signatures that are enabled when it processes traffic.
Block	Connections that match this signature are blocked.
Log	A log entry is produced for any connection that matches this signature.
Stats	The Web App Firewall includes any connection that matches this signature in the statistics that it generates for that check.

- To display only signatures that contain a specific string, type the string into the text box under the filter criteria, and then click Search.
- To reset all display filter criteria to the default settings and display all signatures, click Show All.

4. For information about a specific signature, select the signature, and then click the blue double arrow in the More field. The Signature Rule Vulnerability Detail message box appears. It contains information about the purpose of the signature and provides links to external web-based information about the vulnerability or vulnerabilities that this signature addresses. To access an external link, click the blue double arrow to the left of the description of that link.
5. Configure the settings for a signature by selecting the appropriate check boxes.
6. If you want to add a local signature rule to the signatures object, or modify an existing local signature rule, see The Signatures Editor.
7. If you have no need for SQL injection, cross-site scripting, or Xpath injection patterns, click OK, and then click Close. Otherwise, in the lower left-hand corner of the details pane, click Manage SQL/cross-site scripting Patterns.
8. In the Manage SQL/cross-site scripting Patterns dialog box, Filtered Results window, navigate to the pattern category and pattern that you want to configure. For information about the SQL

injection patterns, see HTML SQL Injection Check. For information about the cross-site scripting patterns, see HTML Cross-Site Scripting Check.

9. To add a new pattern:
   a) Select the branch to which you want to add the new pattern.
   b) Click the **Add** button directly below the lower section of the **Filtered Results** window.
   c) In the Create Signature Item dialog box, fill in the Element text box with the pattern that you want to add. If you are adding a transformation pattern to the transform rules branch, under Elements, fill in the From text box with the pattern that you want to change and the To text box with the pattern to which you want to change the previous pattern.
   d) Click **OK**.
10. To modify an existing pattern:
    a) In the **Filtered Results** window, select the branch that contains the pattern that you want to modify.
    b) In the detail window beneath the **Filtered Results** window, select the pattern that you want to modify.
    c) Click **Modify**.
    d) In the **Modify Signature Item** dialog box, **Element** text box, modify the pattern. If you are modifying a transformation pattern, you can modify either or both patterns under Elements, in the From and the To text boxes.
    e) Click **OK**.
11. To remove a pattern, select the pattern that you want to remove, then click the **Remove** button below the details pane beneath the **Filtered Results** window. When prompted, confirm your choice by clicking **Close**.
12. To add the patterns category to the cross-site scripting branch:
    a) Select the branch to which you want to add the patterns category.
    b) Click the **Add** button directly below the **Filtered Results** window.

       > **Note:** Currently you can add only one category, named patterns, to the cross-site scripting branch, so after you click **Add**,you must accept the default choice, which is patterns.

    c) Click **OK**.
13. To remove a branch, select that branch, and then click the Remove button directly below the **Filtered Results** window. When prompted, confirm your choice by clicking **OK**.

    > **Note:** If you remove a default branch, you remove all of the patterns in that branch. Doing so can disable the security checks that use that information.

14. When you are finished modifying the SQL injection, cross-site scripting, and XPath injection patterns, click **OK**, and then click **Close** to return to the **Modify Signatures Object** dialog box.
15. Click **OK** at any point to save your changes, and when you are finished configuring the signatures object, click **Close**.

Protecting JSON applications using signatures

September 14, 2021

JavaScript Object Notation (JSON) is a text-based open standard derived from the JavaScript scripting language. JSON is preferred for human readable representation of simple data structures and associative arrays, called objects. It serves as an alternative to XML and is primarily used to transmit serialized data structures for communicating with web applications. The JSON files are typically saved with a .json extension.

The JSON payload is typically sent with the MIME type specified as **application/json**. The other "standard" content types for JSON are:

- **application/x-javascript**
- **text/javascript**
- **text/x-javascript**
- **text/x-json**

Using the Citrix Web App Firewall signatures to protect JSON applications

To allow JSON requests, the appliance is preconfigured with the JSON content type as shown in the following show-command output:

```
> sh appfw jsonContentType
1)      JSONContenttypevalue:  "^application/json$" IsRegex:  REGEX
Done
<!--NeedCopy-->
```

The Citrix Web App Firewall processes the post body for the following content-types only:

- **application/x-www-form-urlencoded**
- **multipart/form-data**
- **text/x-gwt-rpc**

The requests that are received with other content-type headers including application/json (or any other allowed content type) are forwarded to the backend after header inspection. The post body in

such requests is not inspected for security check violations even when the profile's security checks such as SQL or cross-site scripting are enabled.

In order to protect JSON applications and detect violations, Web App Firewall signatures can be used. All requests that contain the allowed content-type header are processed by the Web App Firewall for signature match. You can add your own customized signature rules to process JSON payload to perform various security check inspections (for example, cross-site scripting, SQL, and Field Consistency), to detect violations in the headers as well as the post body, and take specified actions.

> **Tip**
>
> Unlike the other built-in defaults, the preconfigured JSON content type can be edited or removed by using the CLI or the GUI (GUI). If legitimate requests for JSON applications are getting blocked and triggering content-type violations, check to make sure that the content type value is configured accurately. For additional details regarding how Web App Firewall processes content-type header, see Content type protection

To add or remove JSON content-type by using the command line interface

At the command prompt, type one of the following commands:

```
add appfw jsonContentType ^application/json$ IsRegEx REGEX
```

```
rm appfw JSONContentType "^application/json$"
```

To managing JSON content types by using the GUI

Navigate to **Security** > **Web App Firewall** and, in the **Settings** section, select **Manage JSON Content Types**.

In the **Configure Web App Firewall JSON Content Type** panel, add, edit, or delete JSON content types to suit the needs of your applications.

Configuring signature protection to detect attacks in JSON payload

In addition to a valid JSON content type, you need to configure signatures to specify the pattern(s) that, when detected in a JSON request, indicate a security breach. The specified actions, such as block and log, are taken when an incoming request triggers a match for all the target patterns in the signature rule.

To add a customized signature rule, Citrix recommends that you use the GUI. Navigate to **System** > **Security** > **Web App Firewall** > **Signatures**. Double click the target signature object to access the **Edit Web App Firewall Signatures** panel. Click on the **Add** button to configure the actions, category, log string, rule patterns and so on. Although Web App Firewall inspects all allowed content-type payload

for signature match, you can optimize the processing by specifying the JSON expression in the rule. When you **Add** a new rule pattern, select **Expression** in the drop-down options for **Match** and provide the target match expression from your JSON payload to identify the specific requests that need to be inspected. An expression must begin with a **TEXT.** prefix. You can add other rule patterns to specify additional match patterns to identify the attack.

The following example shows a signature rule. If any cross-site script tag is detected in the POST body of the JSON payload that matches the specified XPATH_JSON expression, a signature match is triggered.

Example of a signature to detect cross-site scripting in JSON payload

```
<SignatureRule actions="log,stats" category="JSON" enabled="ON" id="
    1000001" severity="" source="" type="" version="1">

  <PatternList>

    <RequestPatterns>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="Expression">TEXT.XPATH_JSON(xp%/glossary/title%).
           CONTAINS("example glossary")</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_METHOD"/>

        <Match type="LITERAL">POST</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="CrossSiteScripting"/>

       </Pattern>

```



```
    </RequestPatterns>

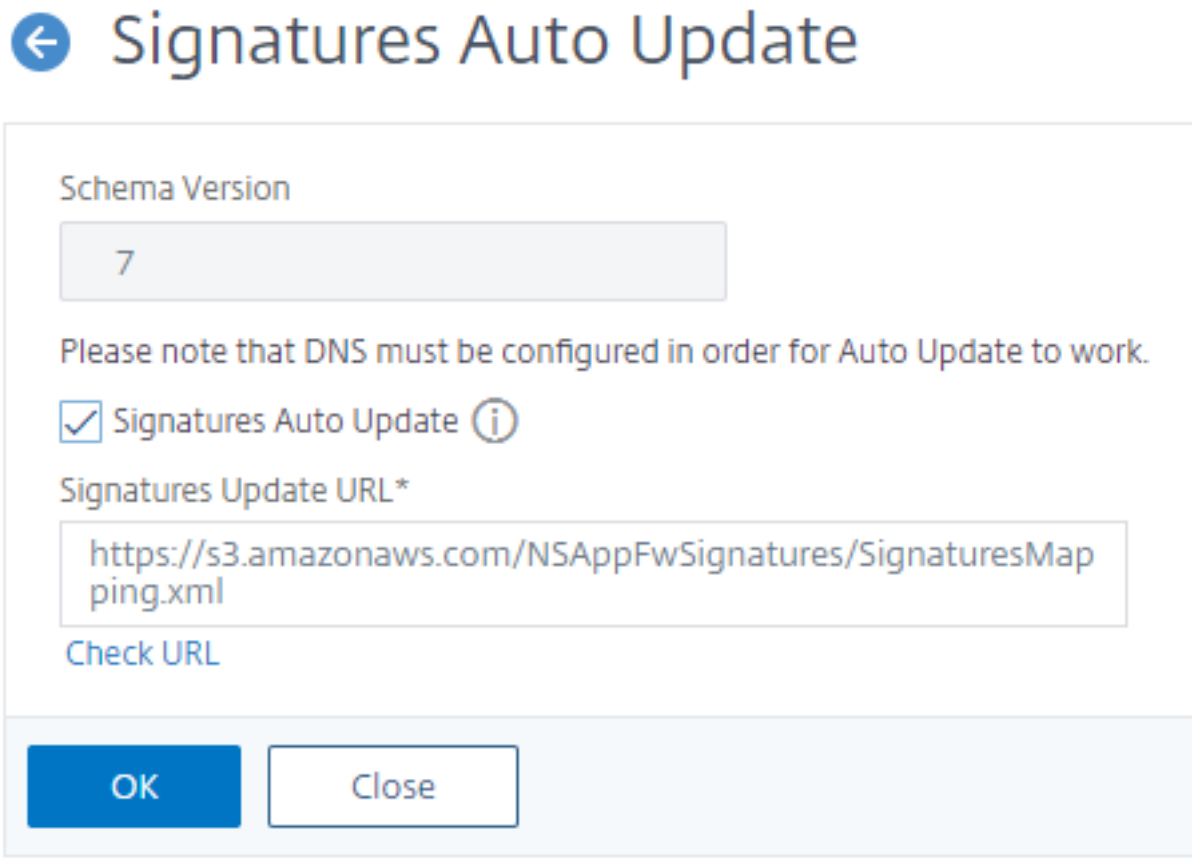
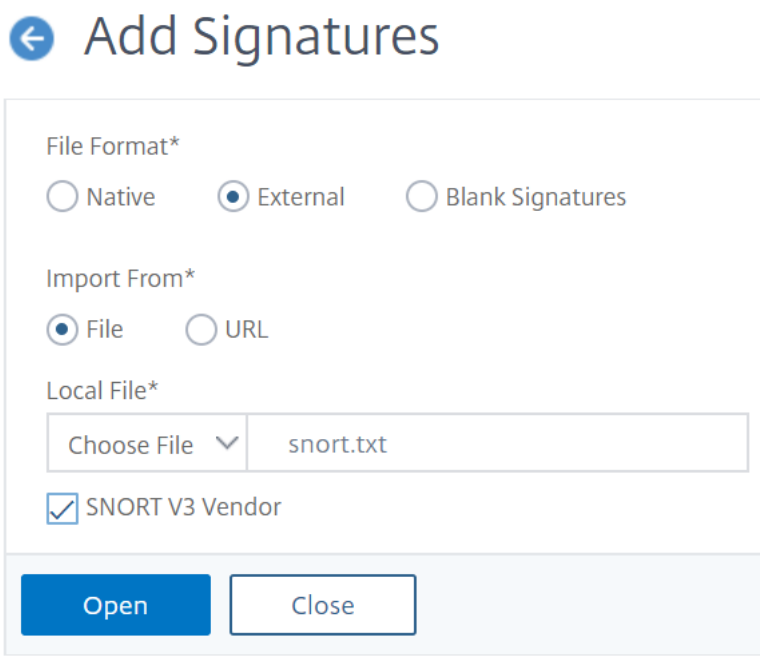
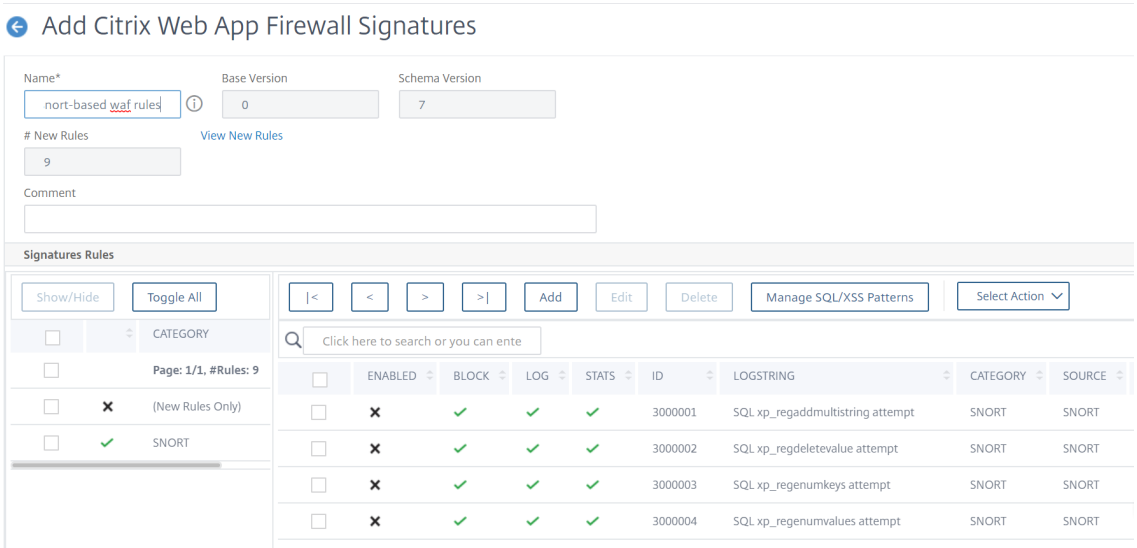
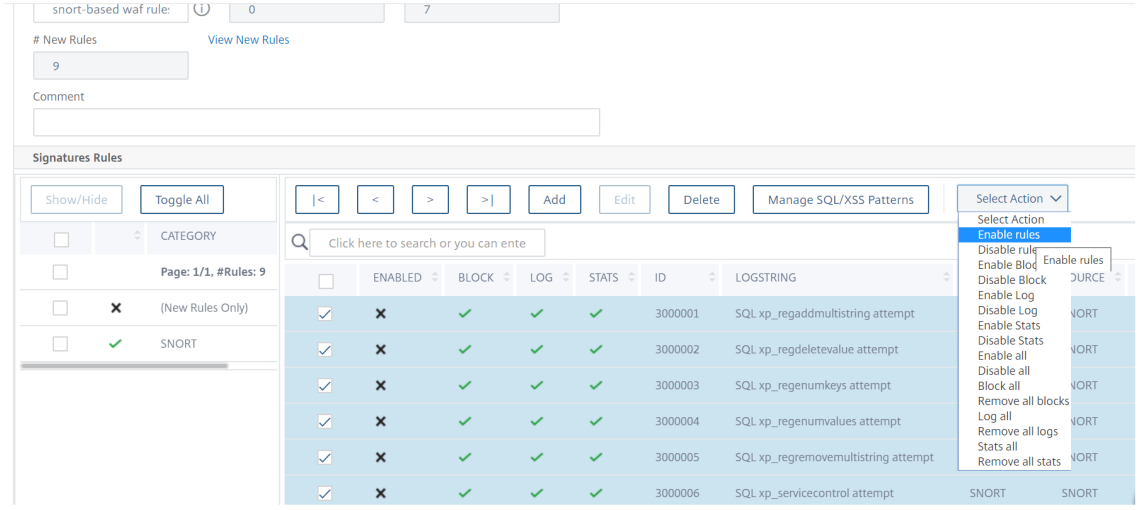
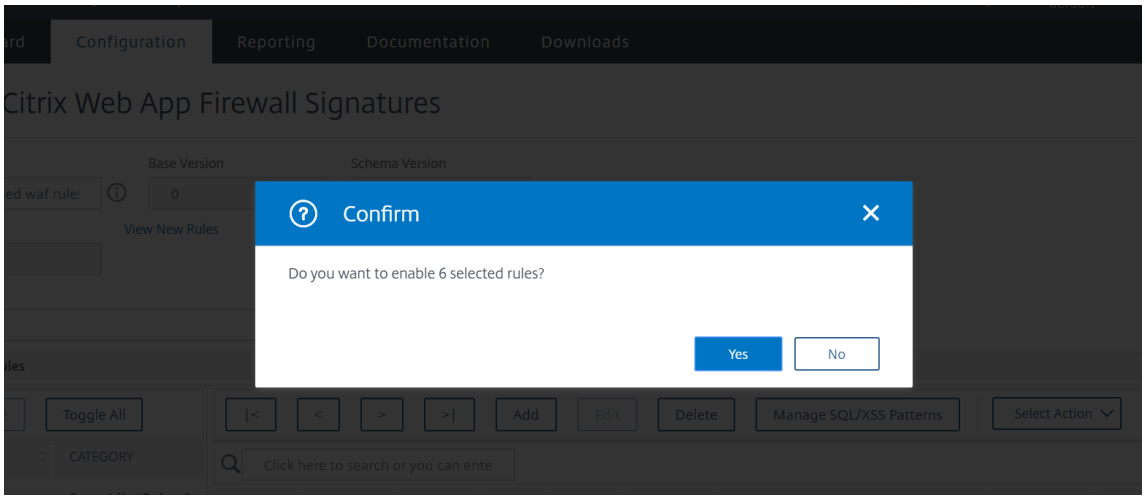
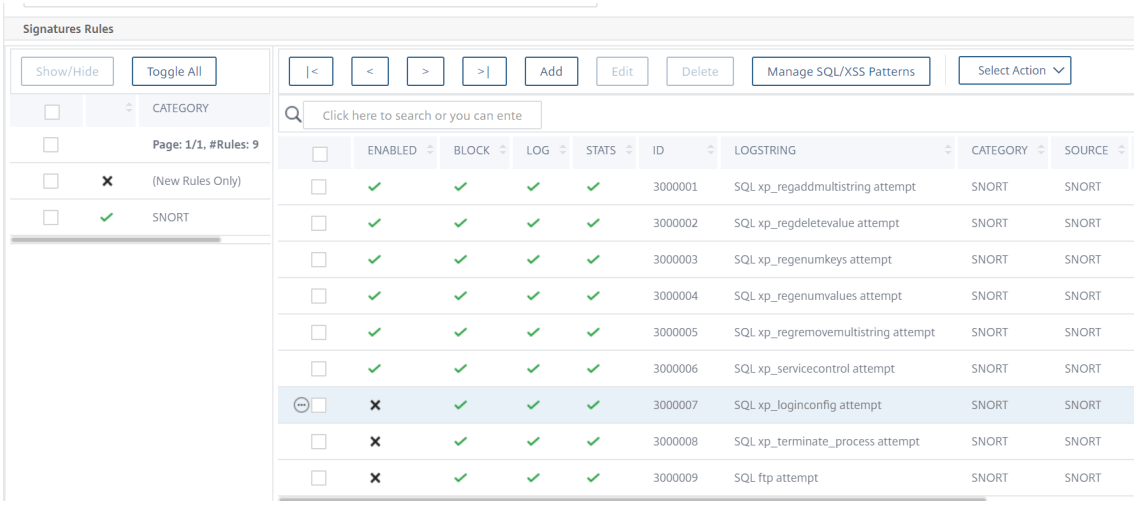
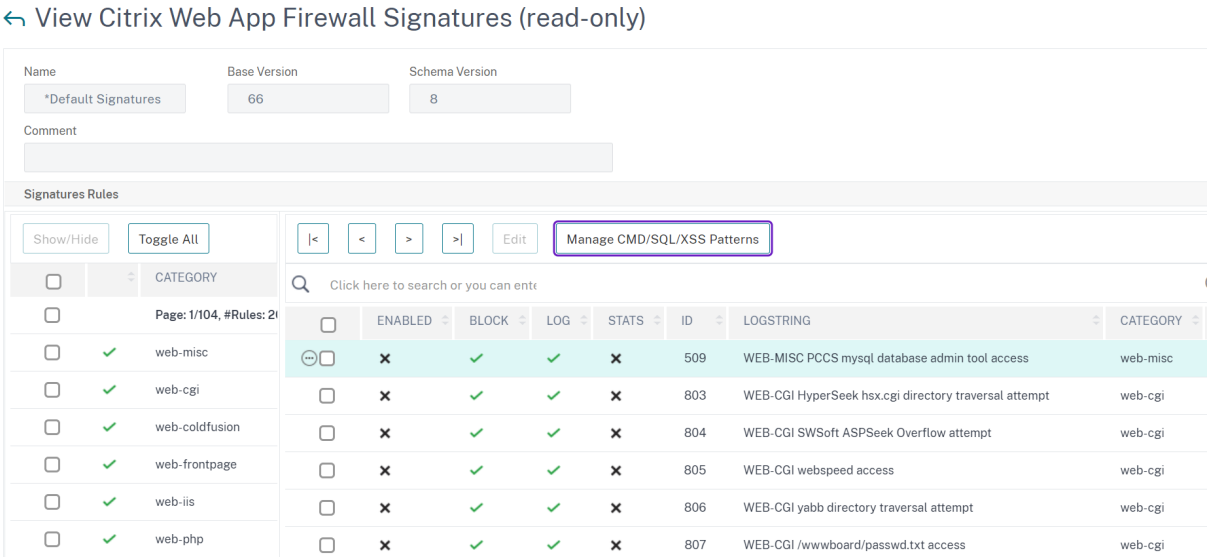
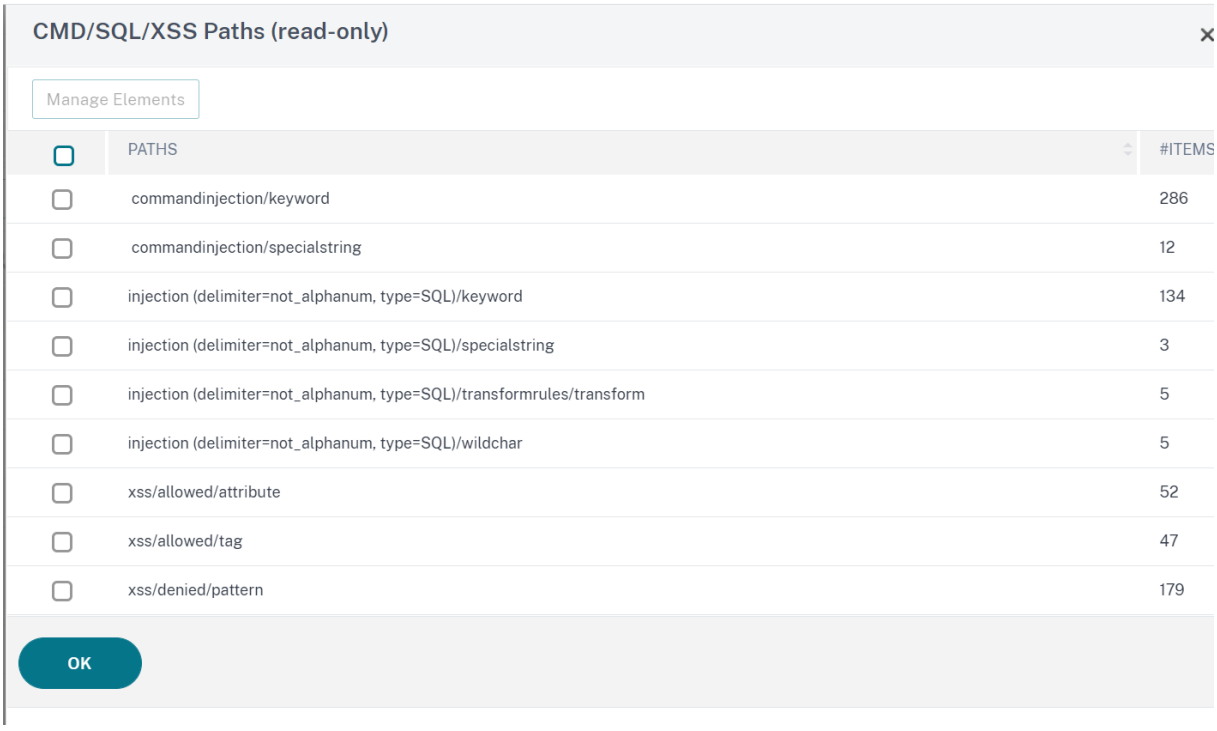
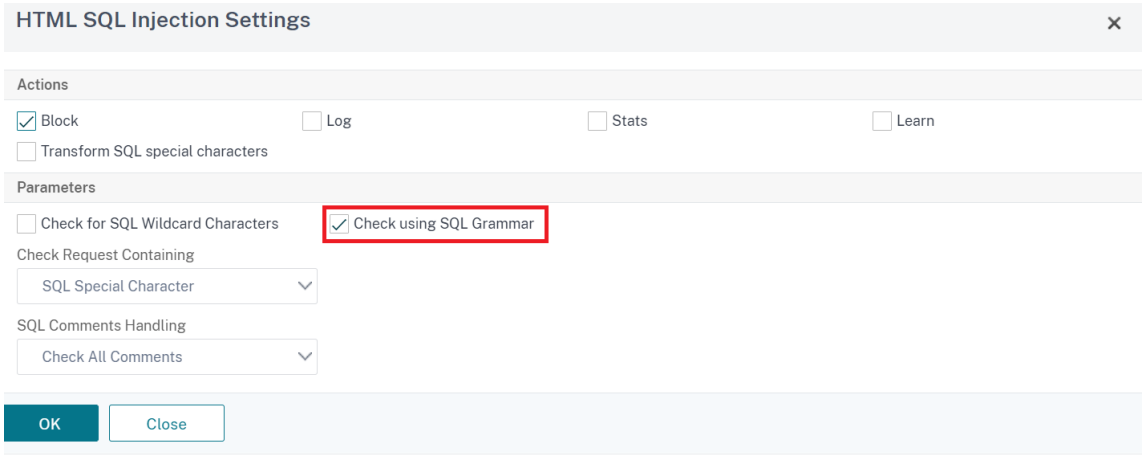
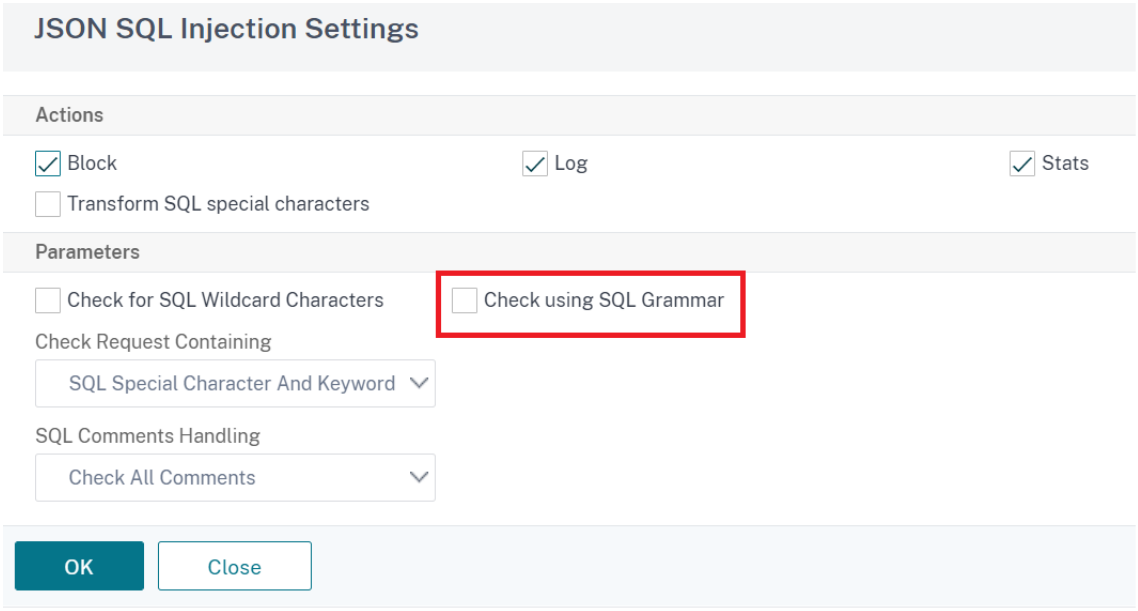
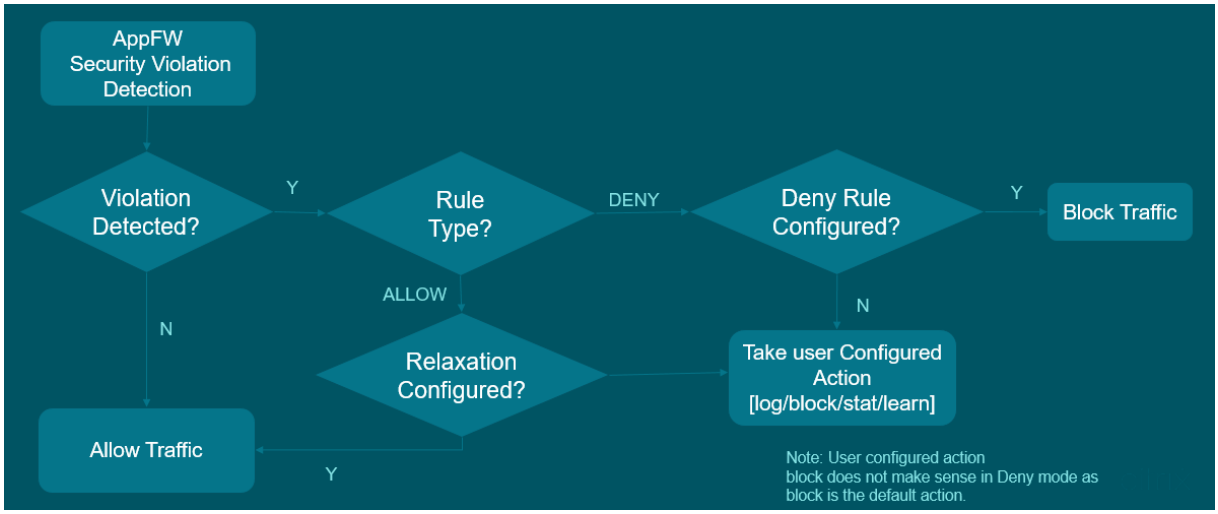
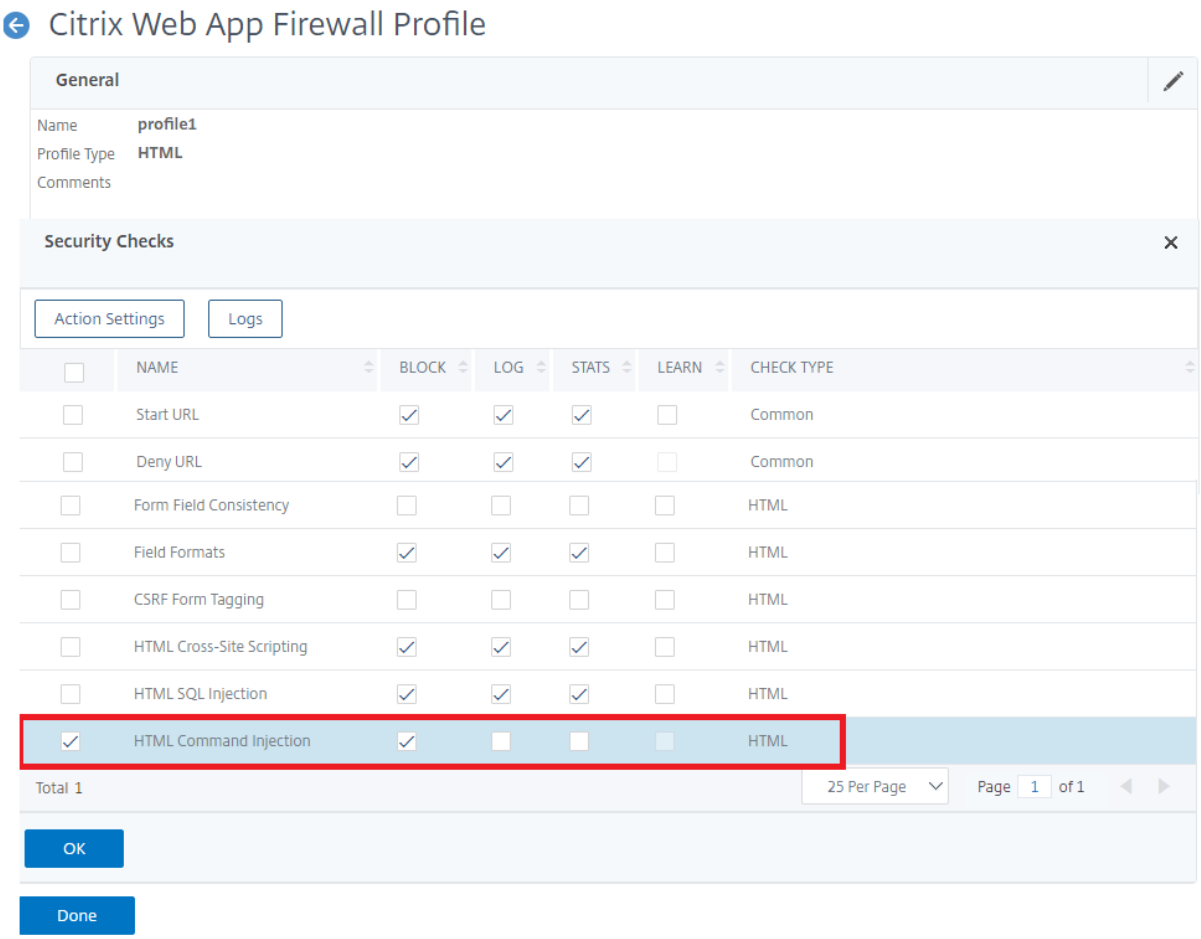
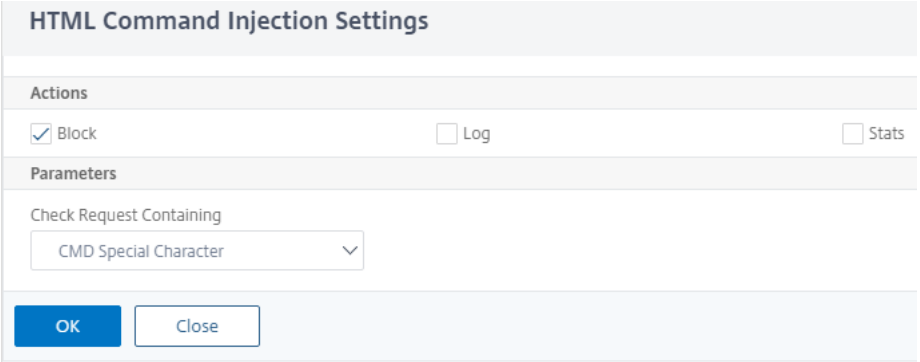
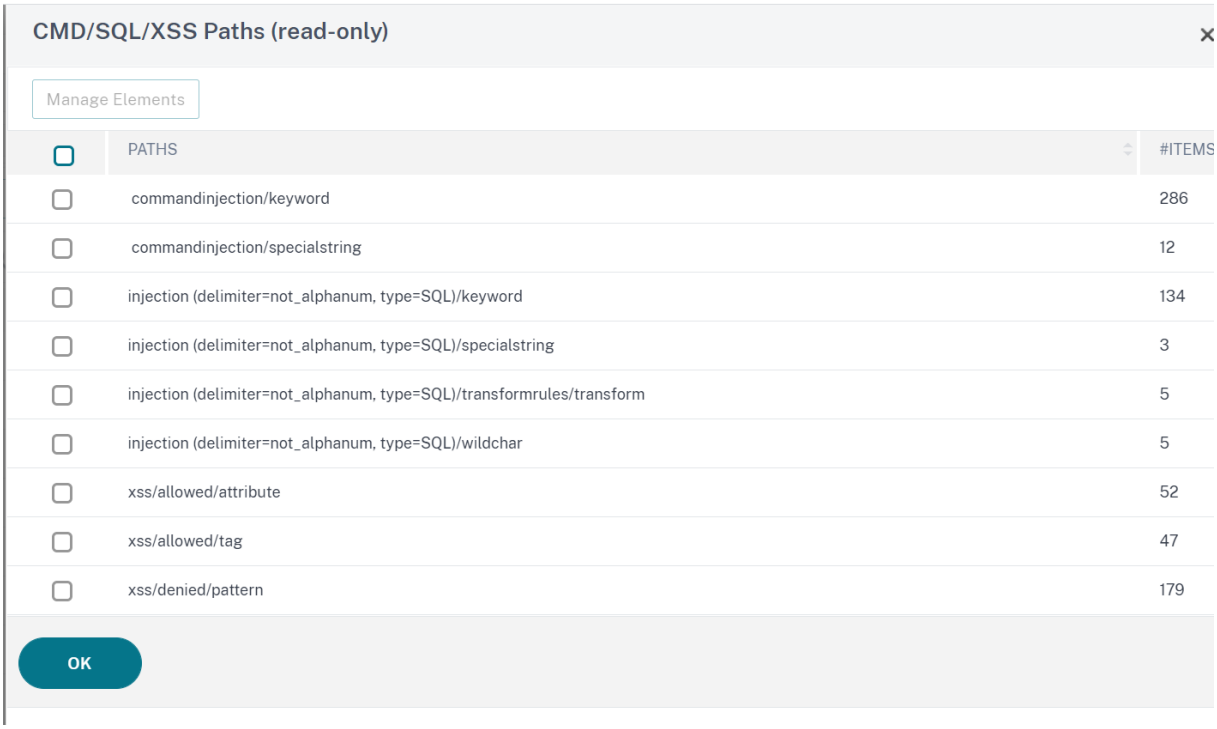
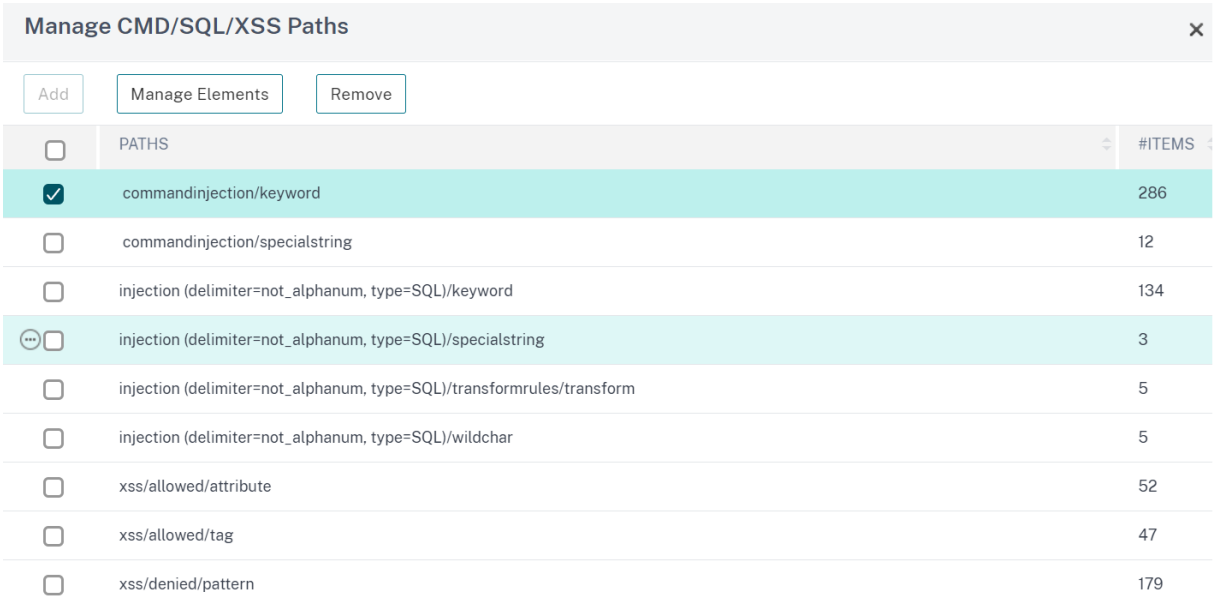
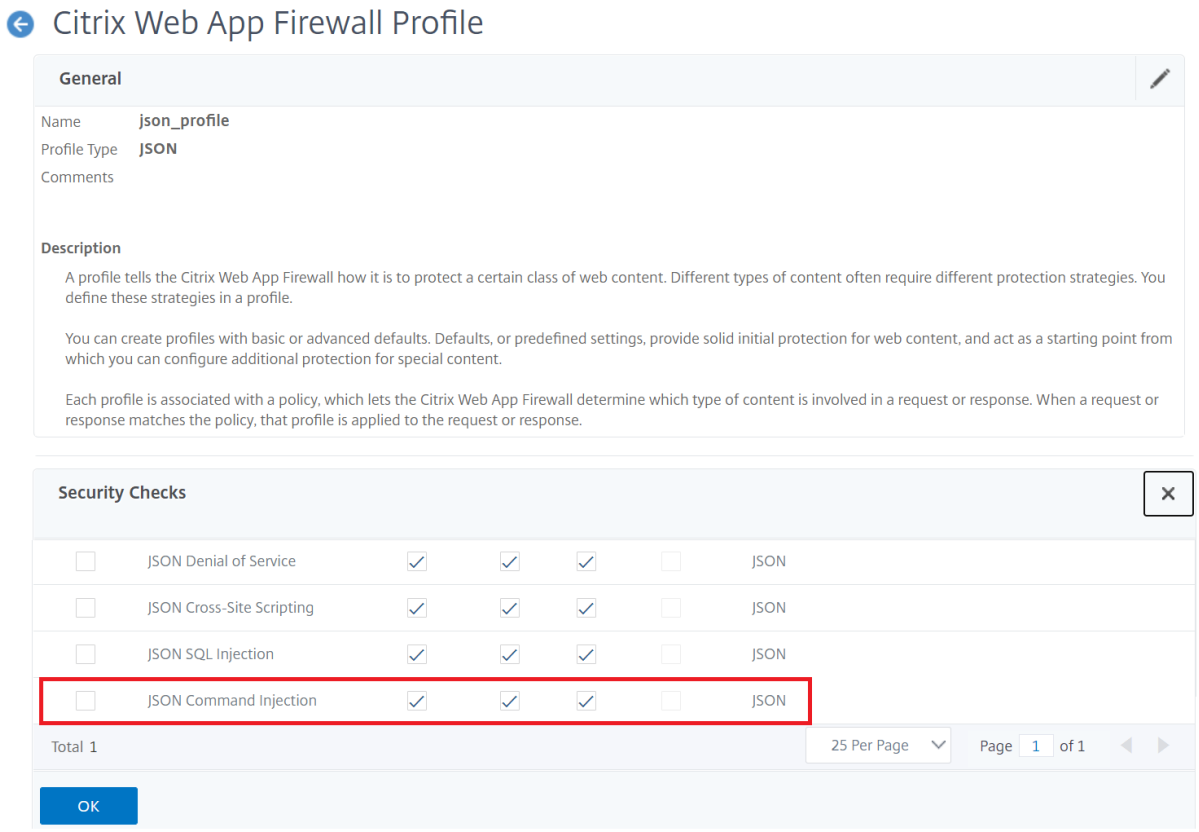
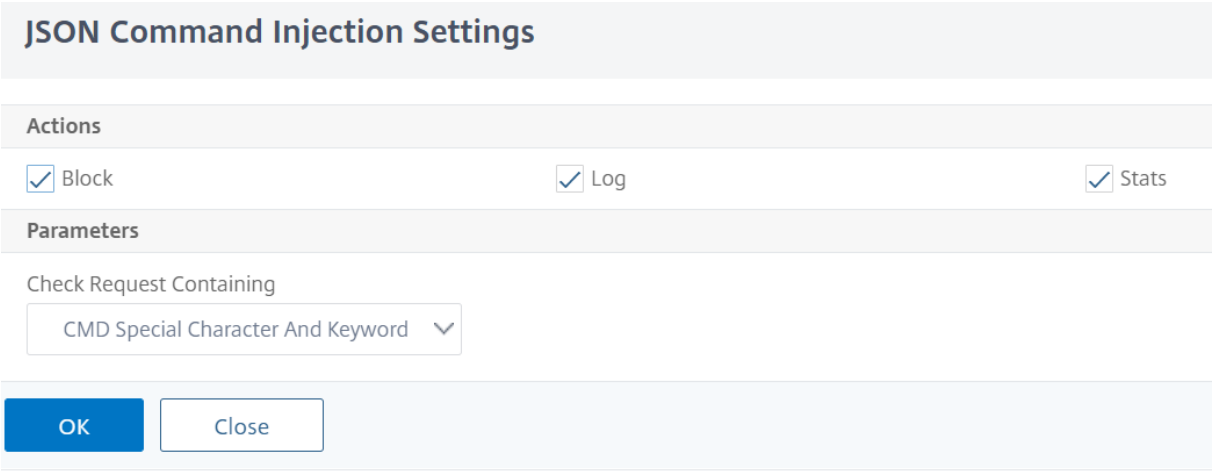
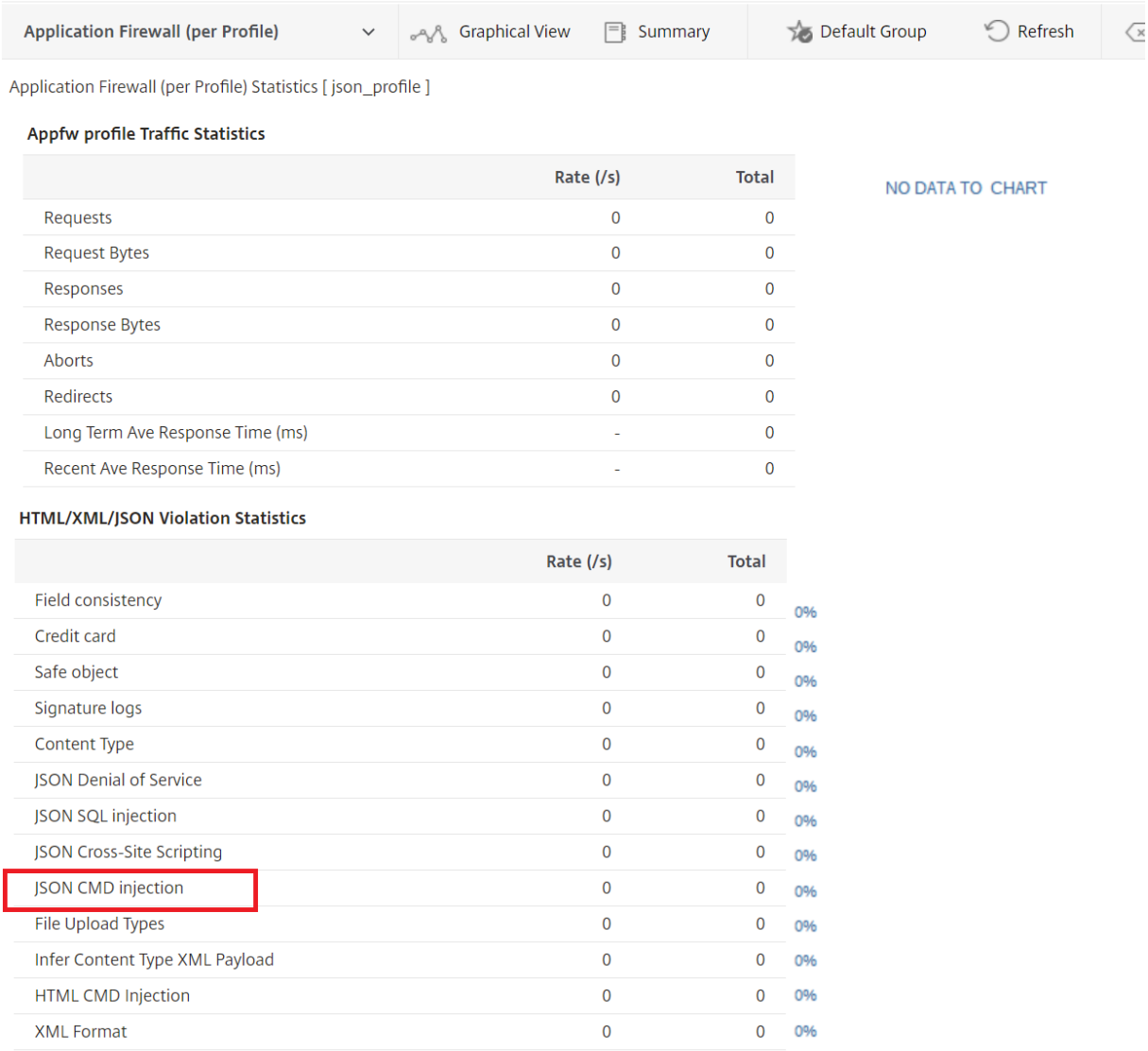
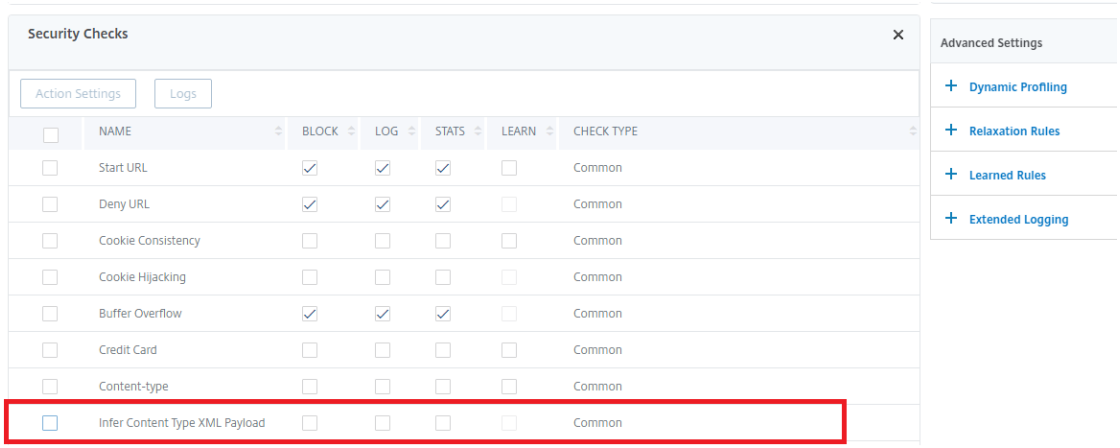
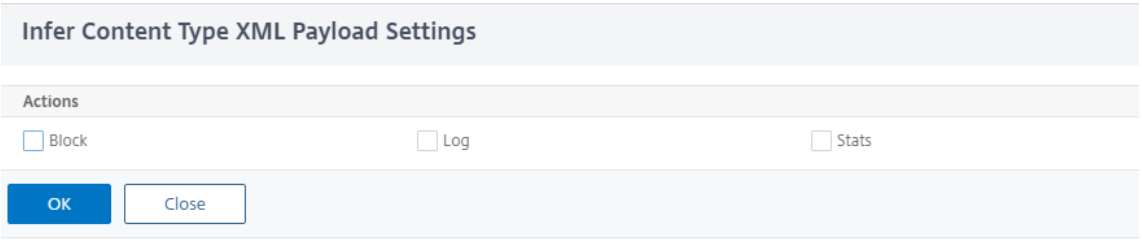
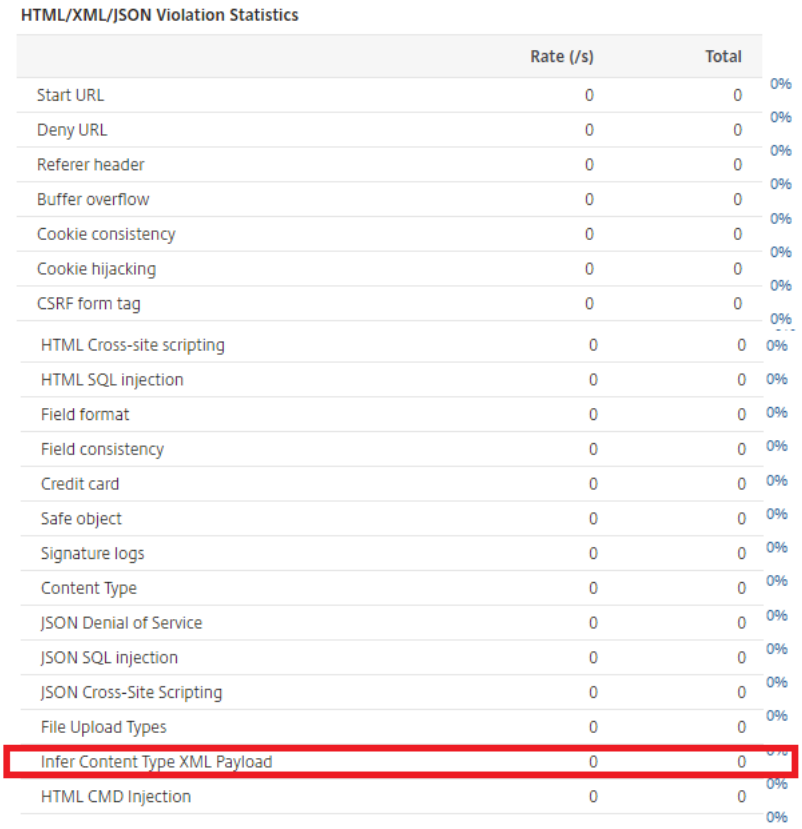
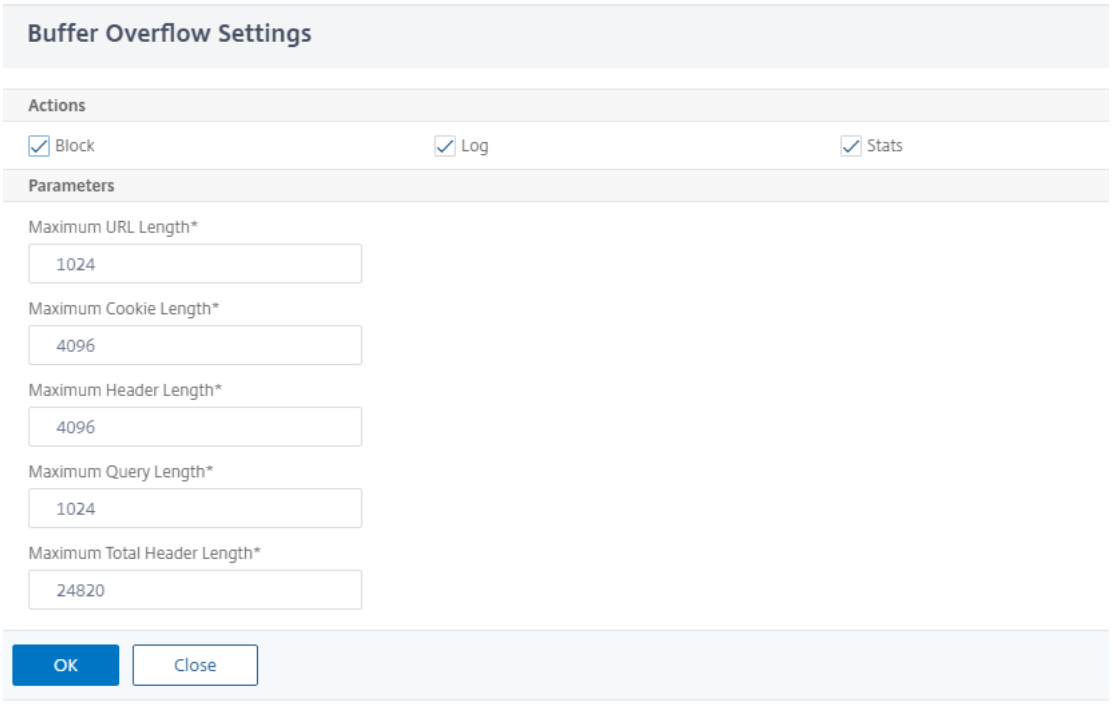
  </PatternList>

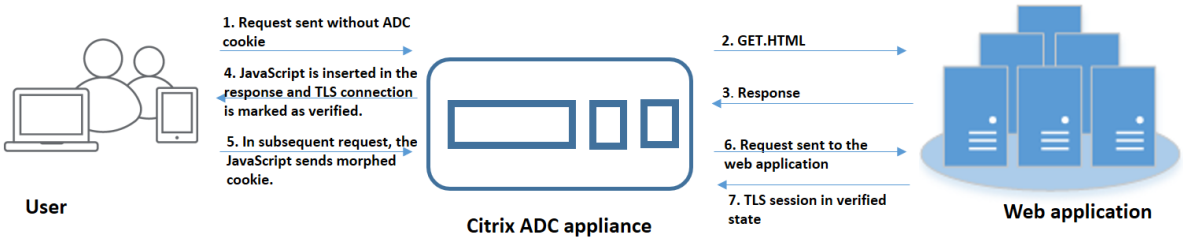
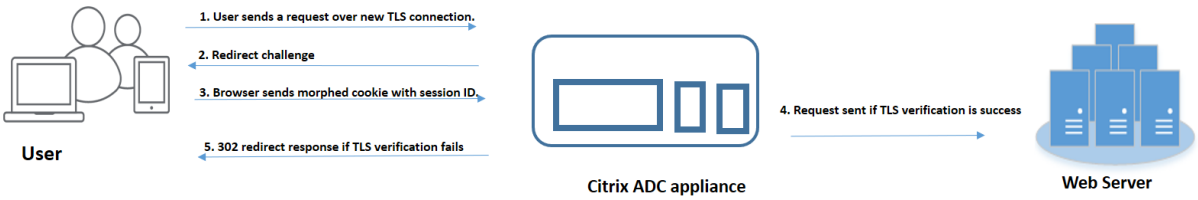
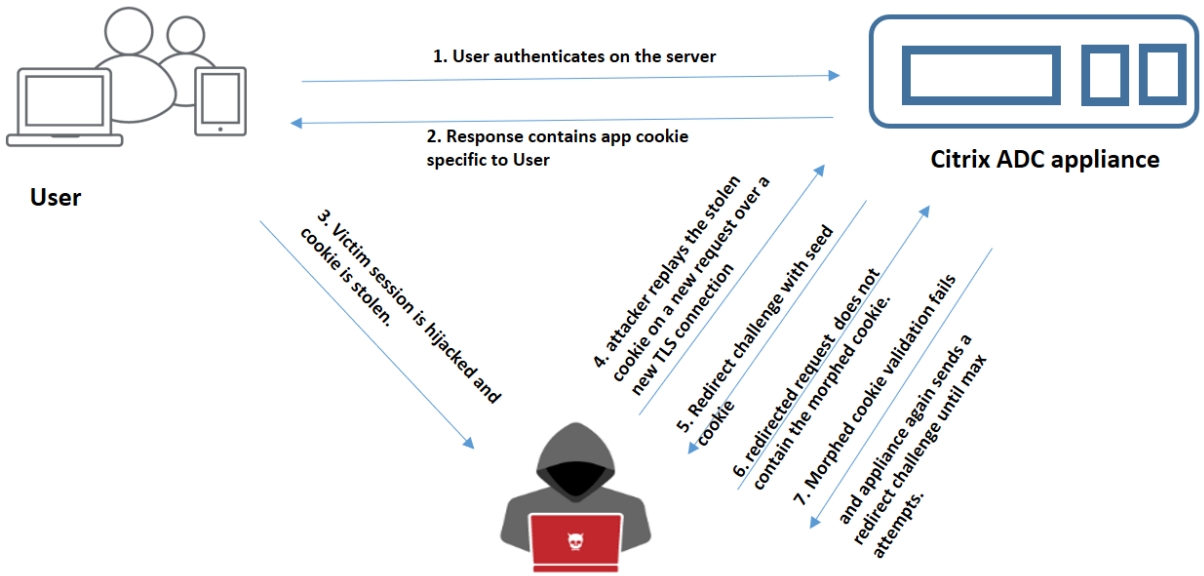
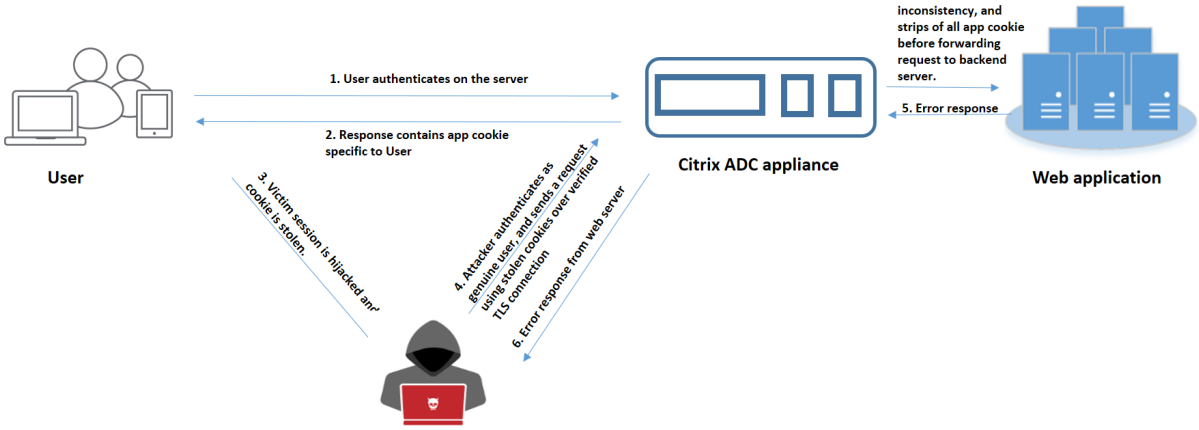
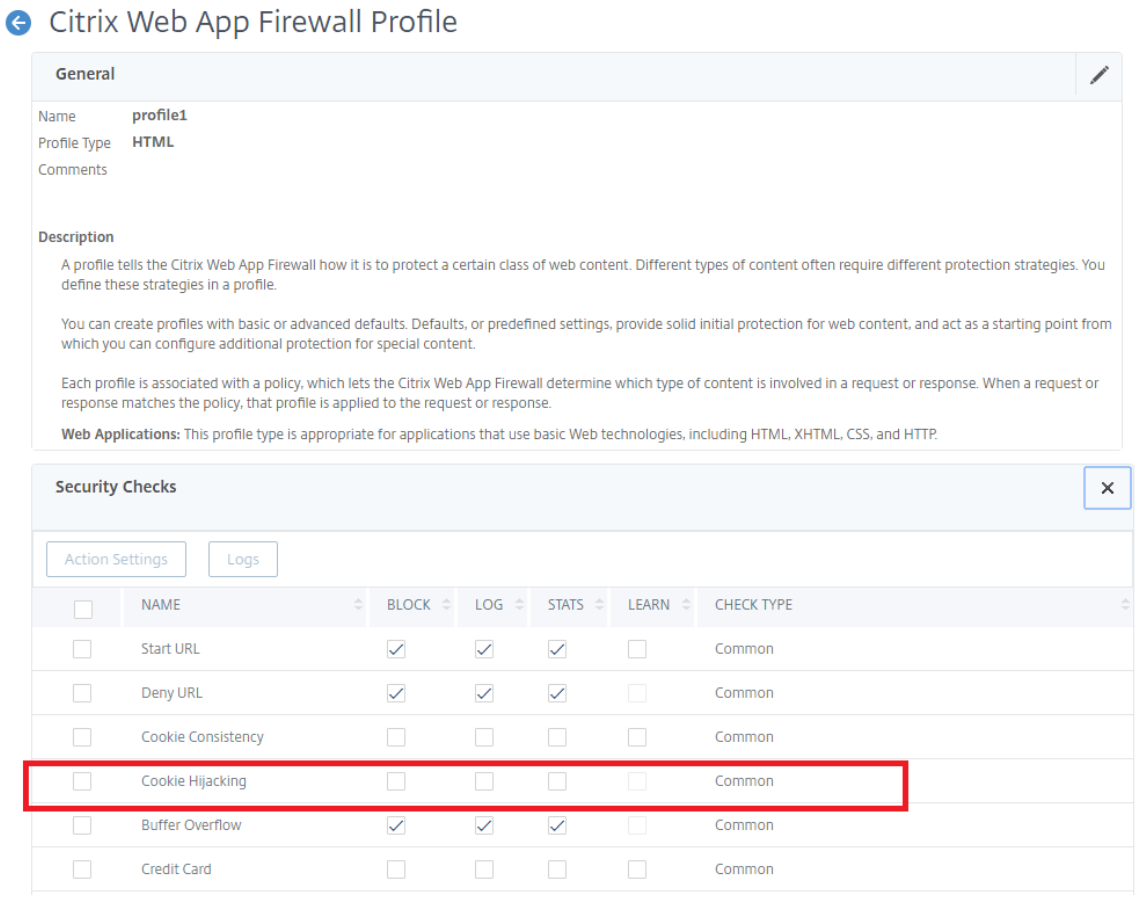
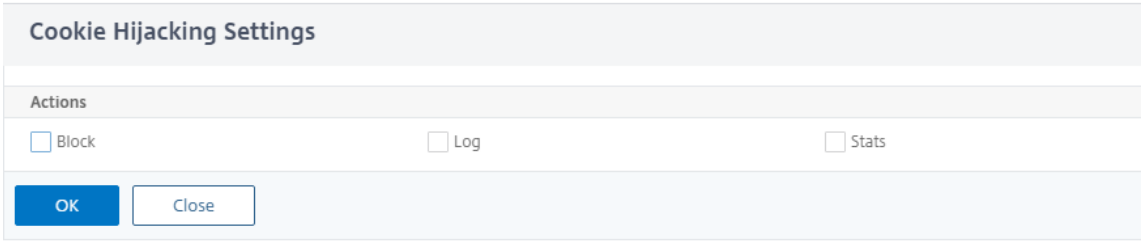
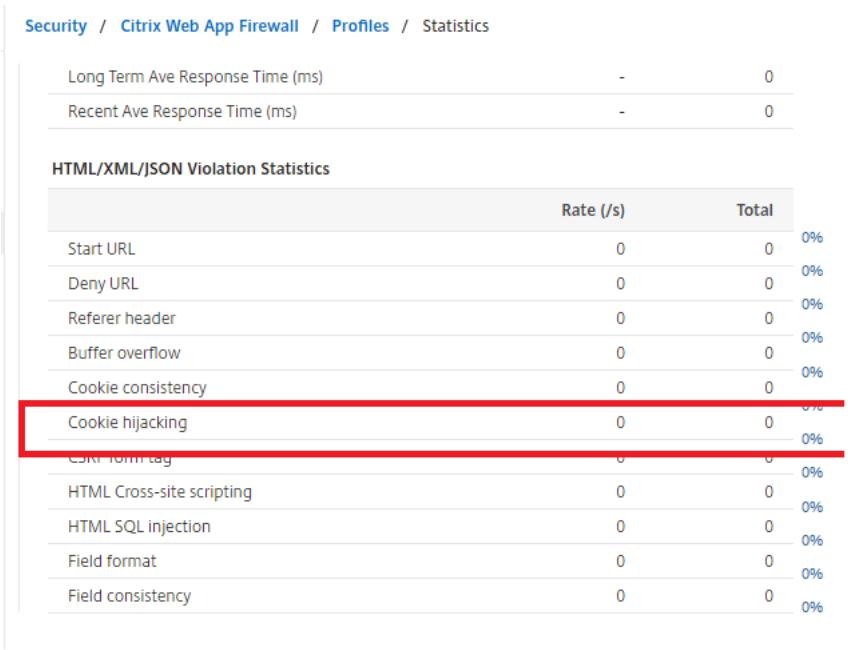
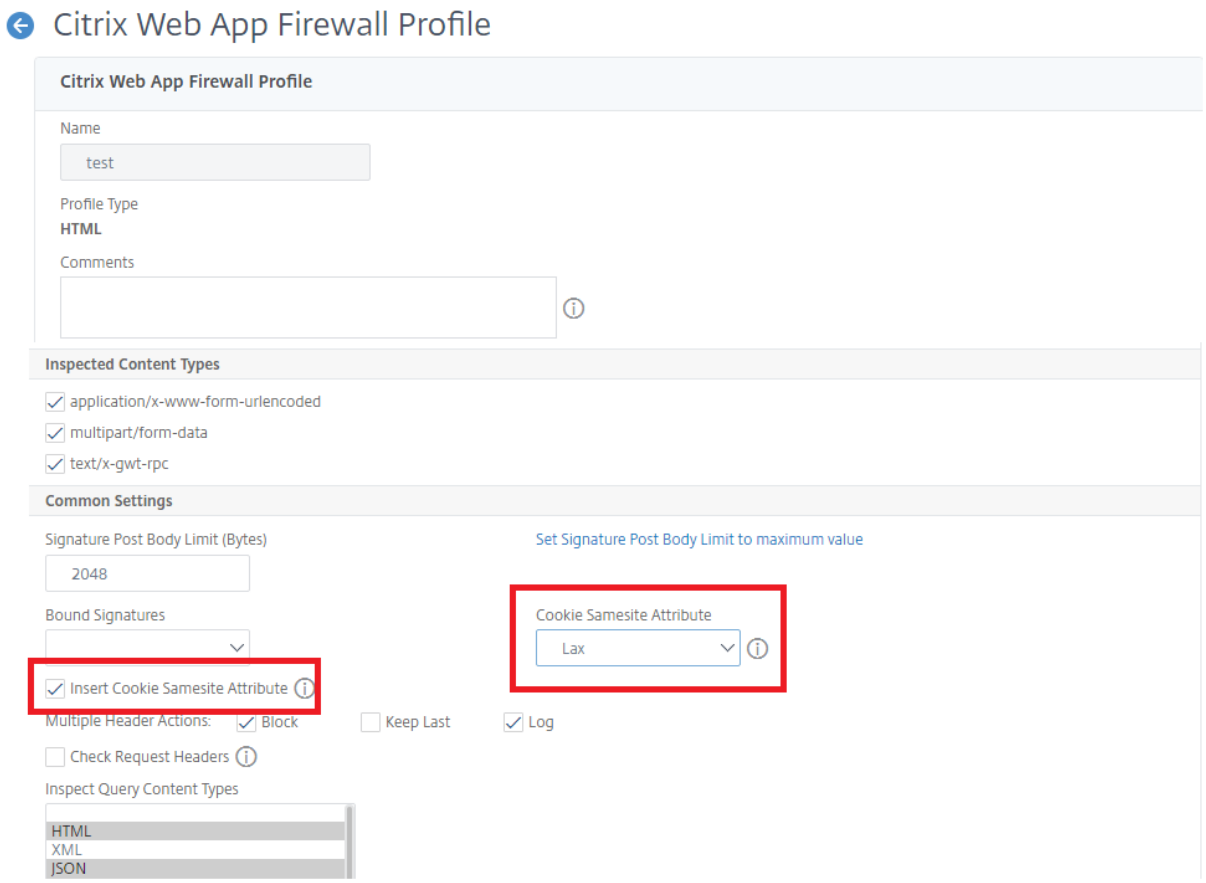
  <LogString>Cross-site scripting violation detected in json payload</
     LogString>

  <Comment/>

</SignatureRule>
<!--NeedCopy-->
```

Example of the payload

The following payload triggers the signature match, because it includes the cross-site scripting tag **<Gotcha!!>**.

```
{
 "glossary": {
 "title": "example glossary","GlossDiv": {
 "title": "S","GlossList": {
 "GlossEntry": {
 "ID": "SGML","SortAs": "SGML","GlossTerm": "Standard Generalized
    Markup Language","Acronym": "SGML","Abbrev": "ISO 8879:1986","
    GlossDef": {
 "para": "A meta-markup language, used to create markup languages **<
    Gotcha!!>** such as DocBook.","GlossSeeAlso": ["GML", "XML"] }
 ,"GlossSee": "markup" }
  }
  }
  }
  }

<!--NeedCopy-->
```

Example of the log message

```
Aug 21 12:21:42 <local0.info> 10.217.31.239 08/21/2015:23:21:42 GMT ns
   0-PPE-1 : APPFW APPFW_SIGNATURE_MATCH 1471 0 :  10.217.253.62 990-
   PPE0 NtJnVMNnvPeQJnaUzXYW/GTvAQsA010 prof1 http://10.217.31.212/FFC/
   login_post.php Signature violation rule ID 1000001: cross-site
   scripting violation detected in json payload  <not blocked>
<!--NeedCopy-->
```


> **Note**
>
> If you send the same payload after removing the cross-site script tag (**<Gotcha!!>**), the signature rule match is not triggered.

Highlights

- To protect JSON payload, use Web App Firewall signatures to detect cross-site scripting, SQL and other violations.
- Verify that the JSON content type is configured on the appliance as the allowed content type.
- Make sure that the content type in the payload matches the configured JSON content type.
- Make sure that all the patterns configured in the signature rule match for the signature violation to be triggered.
- When you add a signature rule, it MUST have at least one Rule pattern to match the Expression in the JSON payload. All the PI expressions in signature rules must start with the prefix TEXT. and must be Boolean.

Protect application or JSON content-type with SQL and cross-site scripting encoded payload using policies and signatures

Citrix Web App Firewall can protect application or JSON content type using policies and signatures.

Inspect application or JSON content type for SQL injection using policies

You must add the following policies and bind it to virtual server globally for supporting SQL injection.

```
add appfw policy sqli_1 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##(((\\A)
|(?<=[^a-zA-Z0-9_])))(select|insert|delete|update|drop|create|alter|grant
|revoke|commit|rollback|shutdown|union|intersect|minus|case|decode|where
|group|begin|join|exists|distinct|add|modify|constraint|null|like|exec|
execute|char|or|and|sp_sdidebug)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK
```

```
add appfw policy sqli_2 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_]))(xp_availablemedia|xp_cmdshell|xp_deletemail|xp_dirtree
|xp_dropwebtask|xp_dsninfo|xp_enumdsn|xp_enumerrorlogs|xp_enumgroups|
xp_enumqueuedtasks|xp_eventlog|xp_findnextmsg|xp_fixeddrives|xp_getfiledetails
|xp_getnetname|xp_grantlogin|xp_logevent|xp_loginconfig|xp_logininfo|
xp_makewebtask|xp_msver|xp_regread|xp_perfend|xp_perfmonitor|xp_perfsample
```



```
|xp_perfstart|xp_readerrorlog|xp_readmail|xp_revokelogin|xp_runwebtask|
xp_schedulersignal|xp_sendmail|xp_servicecontrol|xp_snmp_getstate|xp_snmp_raisetrap
|xp_sprintf|xp_sqlinventory|xp_sqlregister|xp_sqltrace|xp_sscanf|xp_startmail
|xp_stopmail|xp_subdirs|xp_unc_to_drive)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK
```

```
add appfw policy sqli_3 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|MSysACEs|MSysObjects|MSysQueries
|MSysRelationships)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK
```

```
add appfw policy sqli_4 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A
)|(?<=[^a-zA-Z0-9_]))(SYS\.USER_OBJECTS|SYS\.TAB|SYS\.USER_TABLES|SYS\.
USER_VIEWS|SYS\.ALL_TABLES|SYS\.USER_TAB_COLUMNS|SYS\.USER_CONSTRAINTS|SYS
\.USER_TRIGGERS|SYS\.USER_CATALOG|SYS\.ALL_CATALOG|SYS\.ALL_CONSTRAINTS|SYS
\.ALL_OBJECTS|SYS\.ALL_TAB_COLUMNS|SYS\.ALL_TAB_PRIVS|SYS\.ALL_TRIGGERS|SYS
\.ALL_USERS|SYS\.ALL_VIEWS|SYS\.USER_ROLE_PRIVS|SYS\.USER_SYS_PRIVS|SYS\.
USER_TAB_PRIVS)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK
```

Inspect application or JSON content type using signatures

You can add the following signature rules to the signature object in the application firewall profile to support SQL injection for JSON content-type.

> **Note:**
>
> Post body signatures are cpu-intensive.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2013-2018 Citrix Systems, Inc. All rights reserved. -->
<SignaturesFile schema_version="6" version="0" minor_schema_version="0"
   >
    <Signatures>
        <SignatureRule id="4000000" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                           IGNORECASE).SET_TEXT_MODE(URLENCODED).
                           DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((((\\A
```



```
                                )|(?<=[^a-zA-Z0-9_])))(select|insert|delete|
                                update|drop|create|alter|grant|revoke|commit
                                |rollback|shutdown|union|intersect|minus|
                                case|decode|where|group|begin|join|exists|
                                distinct|add|modify|constraint|null|like|
                                exec|execute|char|or|and|sp_sdidebug)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                        </Pattern>
                        <Pattern type="fastmatch">
                            <Location area="HTTP_METHOD"/>
                            <Match type="LITERAL">T</Match>
                        </Pattern>
                    </RequestPatterns>
                </PatternList>
                <LogString>sql Injection</LogString>
                <Comment/>
            </SignatureRule>
            <SignatureRule id="4000001" enabled="ON" actions="log,block"
               category="sql" source="" severity="" type="" version="1"
               sourceid="" harmscore="">
                <PatternList>
                    <RequestPatterns>
                        <Pattern>
                            <Location area="HTTP_POST_BODY"/>
                            <Match type="Expression">TEXT.SET_TEXT_MODE(
                               IGNORECASE).SET_TEXT_MODE(URLENCODED).
                               DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\\A)
                               |(?<=[^a-zA-Z0-9_]))(xp_availablemedia|
                               xp_cmdshell|xp_deletemail|xp_dirtree|
                               xp_dropwebtask|xp_dsninfo|xp_enumdsn|
                               xp_enumerrorlogs|xp_enumgroups|
                               xp_enumqueuedtasks|xp_eventlog|
                               xp_findnextmsg|xp_fixeddrives|
                               xp_getfiledetails|xp_getnetname|
                               xp_grantlogin|xp_logevent|xp_loginconfig|
                               xp_logininfo|xp_makewebtask|xp_msver|
                               xp_regread|xp_perfend|xp_perfmonitor|
                               xp_perfsample|xp_perfstart|xp_readerrorlog|
                               xp_readmail|xp_revokelogin|xp_runwebtask|
                               xp_schedulersignal|xp_sendmail|
                               xp_servicecontrol|xp_snmp_getstate|
                               xp_snmp_raisetrap|xp_sprintf|xp_sqlinventory
                               |xp_sqlregister|xp_sqltrace|xp_sscanf|
                               xp_startmail|xp_stopmail|xp_subdirs|
                               xp_unc_to_drive)((
```



```
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                    </Pattern>
                    <Pattern type="fastmatch">
                        <Location area="HTTP_METHOD"/>
                        <Match type="LITERAL">T</Match>
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
        <SignatureRule id="4000002" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                            IGNORECASE).SET_TEXT_MODE(URLENCODED).
                            DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\\A)
                            |(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|
                            MSysACEs|MSysObjects|MSysQueries|
                            MSysRelationships)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                    </Pattern>
                    <Pattern type="fastmatch">
                        <Location area="HTTP_METHOD"/>
                        <Match type="LITERAL">T</Match>
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
        <SignatureRule id="4000003" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                            IGNORECASE).SET_TEXT_MODE(URLENCODED).
                            DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\\A)
```



```
                                |(?<=[^a-zA-Z0-9_]))(SYS\.USER_OBJECTS|SYS\.
                                TAB|SYS\.USER_TABLES|SYS\.USER_VIEWS|SYS\.
                                ALL_TABLES|SYS\.USER_TAB_COLUMNS|SYS\.
                                USER_CONSTRAINTS|SYS\.USER_TRIGGERS|SYS\.
                                USER_CATALOG|SYS\.ALL_CATALOG|SYS\.
                                ALL_CONSTRAINTS|SYS\.ALL_OBJECTS|SYS\.
                                ALL_TAB_COLUMNS|SYS\.ALL_TAB_PRIVS|SYS\.
                                ALL_TRIGGERS|SYS\.ALL_USERS|SYS\.ALL_VIEWS|
                                SYS\.USER_ROLE_PRIVS|SYS\.USER_SYS_PRIVS|SYS
                                \.USER_TAB_PRIVS)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                    </Pattern>
                    <Pattern type="fastmatch">
                        <Location area="HTTP_METHOD"/>
                        <Match type="LITERAL">T</Match>
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
    </Signatures>
</SignaturesFile>

<!--NeedCopy-->
```

Updating a signature object

September 14, 2021

You must update your signatures objects frequently to ensure that your Web App Firewall is providing protection against current threats. You must regularly update both the default Web App Firewall signatures and any signatures that you import from a supported vulnerability scanning tool.

Citrix regularly updates the default signatures for the Web App Firewall. You can update the default signatures manually or automatically. In either case, ask your Citrix representative or Citrix reseller for the URL to access the updates. You can enable automatic updates of the Citrix native format signatures in the “Engine Settings” and “Signature Auto Update Settings” dialog boxes.

Most makers of vulnerability scanning tools regularly update the tools. Most websites also change frequently. You must update your tool and rescan your websites regularly, exporting the resulting signatures to a file and importing them into your Web App Firewall configuration.

> **Tip**
>
> When you update the Web App Firewall signatures from the Citrix ADC command line, you must first update the default signatures, and then issue more update commands to update each custom signatures file that is based on the default signatures. If you do not update the default signatures first, a version mismatch error prevents updating of the custom signatures files.
>
> **Note**
>
> The following applies to merging a third-party signature object with a user-defined signature object with Native rules and user-added rules:
>
> When a version 0 signatures is merged with a new imported file, the resultant signatures remain as version 0.
>
> This means all native (or built-in) rules in the imported file will be ignored after the merge. This is to ensure that the version 0 signatures are maintained as is after a merge.
>
> To include the native rules in the imported file for merge, you must update the existing signatures from version 0 first before the merge. This means you need to abandon the version 0 nature of the existing signatures.
>
> When there is a Citrix ADC release upgrade, the file "default_signatures.xml" is added to the new build and the file "updated_signature.xml" is removed from the older build. After the upgrade, if the signature auto update feature is enabled, the appliance updates the existing signature to the latest version of the build and generates the "updated_signature.xml" file.

To update the Web App Firewall signatures from the source by using the command line

At the command prompt, type the following commands:

- `update appfw signatures <name> [-mergedefault]`
- `save ns config`

Example

The following example updates the signatures object named MySignatures from the default signatures object, merging new signatures in the default signatures object with the existing signatures. This command does not overwrite any user-created signatures or signatures imported from another source, such as an approved vulnerability scanning tool.

```
1 update appfw signatures MySignatures -mergedefault
2 save ns config
3 <!--NeedCopy-->
```


Updating a signatures object from a Citrix format file

Citrix regularly updates the signatures for the Web App Firewall. You must regularly update the signatures on your Web App Firewall to ensure that your Web App Firewall is using the most current list. Ask your Citrix representative or Citrix reseller for the URL to access the updates.

To update a signatures object from a Citrix format file by using the command line

At the command prompt, type the following commands:

- `update appfw signatures <name> [-mergeDefault]`
- `save ns config`

To update a signatures object from a Citrix format file by using the GUI

1. Navigate to **Security** > **Web App Firewall** > **Signatures**.
2. In the details pane, select the signatures object that you want to update.
3. In the **Action** drop-down list, select **Merge**.
4. In the **Update Signatures Object** dialog box, choose one of the following options.
   - **Import from URL**—Choose this option if you download signature updates from a web URL.
   - **Import from Local File**—Choose this option if you import signature updates from a file on your local hard drive, network hard drive, or other storage device.
5. In the text area, type the URL, or type or browse to the local file.
6. Click **Update**. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the **Modify Signatures Object** dialog box. The **Update Signatures Object** dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
7. Review and configure the new and modified signatures.
8. When you are finished, click **OK**, and then click **Close**.

Updating a signatures object from a supported vulnerability scanning tool

> **Note:**
>
> Before you update a signatures object from a file, you must create the file by exporting signatures from the vulnerability scanning tool.

To import and update signatures from a vulnerability scanning tool

1. Navigate to **Security** > **Web App Firewall** > **Signatures**.
2. In the details pane, select the signatures object that you want to update, and then click **Merge**.

3. In the **Update Signatures Object** dialog box, on the **External Format** tab, Import section, choose one of the following options.
   - **Import from URL**—Choose this option if you download signature updates from a Web URL.
   - **Import from Local File**—Choose this option if you import signature updates from a file on your local or a network hard drive or other storage device.
4. In the text area, type the URL, or browse or type the path to the local file.
5. In the XSLT section, choose one of the following options.
   - **Use Built-in XSLT File**—Choose this option if you want to use a built-in XSLT file.
   - **Use Local XSLT File**—Choose this option to use an XSLT file on your local computer.
   - **Reference XSLT from URL**—Choose this option to import an XSLT file from a web URL.
6. If you chose Use Built-in XSLT File, in the Built-In XSLT drop-down list select the file that you want to use from the following options:
   - **Cenzic.**
   - **Deep_Security_for_Web_Apps.**
   - **Hewlett_Packard_Enterprise_WebInspect.**
   - **IBM-AppScan-Enterprise.**
   - **IBM-AppScan-Standard.**
   - **Qualys.**
   - **Whitehat.**
7. Click **Update**. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the Modify Signatures Object dialog box, which is described in Configuring or Modifying a Signatures Object. The **Update Signatures Object** dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
8. Review and configure the new and modified signatures.
9. When you are finished, click **OK**, and then click **Close**.

Signature auto update

September 14, 2021

The Signature Auto Update functionality in the Web Application Firewall allows the user to get the latest signatures to protect the web application against new vulnerabilities. The auto update feature provides better protection without the need for ongoing manual intervention to get the latest updates.

The signatures are auto updated on an hourly basis and do not require regular check for the availability of the most recent update. Once you enable Signature Auto Update, then the Citrix ADC appliance connects to the server hosting the signatures to check if a newer version is available.

Customizable location

The latest Application Firewall signatures are hosted on Amazon which is configured as the default Signature URL to check for the latest update.

However, the user has an option to download these signature mapping files to their internal server. User can then configure a different Signature URL path to download the signature mapping files from a local server. For the auto update feature to work, you might need to configure the DNS server to access the external site.

Update signatures

All the user defined signature objects which are created using the appfw default signature object have a version greater than zero. If you enable Signature Auto Update, then all the signatures are updated automatically.

If the user has imported signatures with the external format such as Cenzic or Qualys, then the signatures are imported with the version as zero. Similarly, if the user has created a signature object using the blank template, then it is created as a zero version signature. These signatures are not automatically updated, because the user might not be interested in the overhead of managing the default signatures that is not used.

However, Web Application Firewall also allows the user the flexibility to manually select these signatures and update them to add the default signature rules to the existing rules. After the signatures are manually updated, the version changes and then the signatures will also get auto updated along with the other signatures.

Configure the signature auto update

To configure the signature auto update feature using the CLI:

At the command prompt, type:

```
1 set appfw settings SignatureAutoUpdate on
2 set appfw settings SignatureUrl https://s3.amazonaws.com/
     NSAppFwSignatures/SignaturesMapping.xml
3 <!--NeedCopy-->
```

To configure the signature auto update using the GUI:

1. Expand the Security node.
2. Expand the Application Firewall node.
3. Select the Signatures node.
4. Select **Auto Update Settings** from **Action**.

5. Enable the **Signatures Auto Update** option.
6. You can specify a customized path for the signature update URL, if necessary. Click **Reset** to reset to the default `s3.amazonaws.com server`.
7. Click **OK**.



Update signatures manually

To manually update a zero version signature or any other user defined signature, you must first get the latest update for the default signatures and then use this for updating the target user defined signature.

Run the following commands from the CLI to update a signature file:

```
update appfw signatures "*Default Signatures"
update appfw signatures cenzic  -mergedefault
<!--NeedCopy-->
```

> **Note:**
>
> `Default Signatures` Is case sensitive. Cenzic in the preceding command is the name of the

> signature file that is updated.

Import default signatures without internet access

It is recommended to configure a proxy server to point to Amazon (AWS) server to get the latest update. However, if the NetScaler appliance does not have an internet connection to the external sites, then the user can store the updated signature files on a local server. The appliance can then download the signatures from the local server. In this scenario, the user must constantly check the **Amazon site** to get the latest updates. You can download and verify the signature file against the corresponding sha1 file which were created by using the **Citrix public** key to protect against tampering.

To copy the Signatures files to a local server, complete the following procedure:

1. Create a local directory such as <MySignatures> on a local server.
2. Open the AWS site.
3. Copy the SignaturesMapping.xml file to the <MySignatures> folder.

If you open the SignaturesMapping.xml file, you can see all the xml files for signatures and their corresponding sha1 files for different supported versions. One such pair is highlighted in the following screenshot:

1. Create a subdirectory <sigs> in the <MySignatures> folder.
2. Copy all pairs of the *.xml files listed in the <file> tags and the *.xml.sha1 files listed in the corresponding <sha1> tags of the SignaturesMapping.xml file to the <sigs> folder. The following are a few sample files that is copied to the <sigs> folder:

```
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml.sha1
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml.sha1
```

> **Note:**
>
> You can give any name to the <MySignatures> folder and it can be in any location but the sub-directory <sigs> must be a subdirectory in the <MySignatures> folder where the mapping file is copied. In addition, ensure that as shown in the SignaturesMapping.xml, the subdirectory name <sigs> must have the exact name and is case sensitive. All Signature files and their corresponding sha1 files should be copied under this <sigs> directory.

After mirroring the contents from the hosted Amazon web server to the local server, change the path to the new local web server to set it as the SignatureUrl for auto update. For example, run the following command from the command line interface of the appliance:

```
1 set appfw settings SignatureUrl https://myserver.example.net/
     MySignatures/SignaturesMapping.xml
```



```
<!--NeedCopy-->
```

The update operation can take several minutes, depending on the number of signatures to be updated. Allow sufficient time for the update operation to complete.

If you face an error “Error in accessing URL!” while configuring, follow the steps to resolve it.

1. Add the url `https://myserver.example.net` to `/netscaler/ns_gui/admin_ui/php/application/controllers/common/utils.php` so that Content Security Policy (CSP) security does not block the url access. Please note that these settings do not persists in an upgrade. User has to add it again after the upgrade.

```
$configuration_view_connect_src = "connect-src 'self' https://app.pendo.io https://s3.amazonaws.comhttps://myserver.example.net;";
<!--NeedCopy-->
```

1. User must configure the webserver `https://myserver.example.net` such that it responds to the following CORS headers for `https://myserver.example.net/MySignatures/SignaturesMapping.xml`

```
Access-Control-Allow-Methods: GET
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 3000
<!--NeedCopy-->
```

Guidelines to update signatures

Following guidelines are used when updating signatures:

- The signatures are updated when the Signature update URL has a signature object which has the same or newer version.
- Each Signature Rule is associated with a rule ID and version number. For example: `<SignatureRule id="803"version="16"···>`
- Signature Rule from the incoming Signatures file with the same ID and version number as the existing one is ignored even if it has different patterns or log string.
- Signature Rule with a new ID is added. All the actions and enabled flag are used from the new file.

> **Note:**
>
> You might still must review the updated signatures periodically to enable these newly added rules and change other action settings as per the requirements of the application.

- Rules with the same ID but with a newer version number replace the existing one. All the actions and enabled flag from the existing rule is preserved.

> **Tip:**
>
> When you update the signatures from the CLI, you must first update the default signatures. You must then add update commands to update each custom signature file that is based on the default signatures. If you do not update the default signatures first, a version mismatch error prevents custom signatures file update.

Snort rule integration

September 14, 2021

With malicious attacks on web applications, it is important to protect your internal network. Malicious data not only affect your web applications at the interface level but malicious packets also reach the application layer. To overcome such attacks, it is important to configure an intrusion detection and prevention system that examines your internal network.

Snort rules are integrated into the appliance for examine malicious attacks in data packets at the application layer. You can download the snort rules and convert it into WAF signatures rules. The signatures have rule-based configuration that can detect malicious activities such as DOS attacks, buffer overflows, stealth port scans, CGI attacks, SMB probes, and OS Fingerprinting attempts. By integrating Snort rules, you can strengthen your security solution at the interface and at the application level.

Configure snort rules

The configuration begins by first downloading the Snort rules and then importing it into WAF signature rules. Once you have converted the rules into WAF signatures, the rules can be used as WAF security checks. The snort based signature rules examine the incoming data packet to detect if there are malicious attacks on your network.

A new parameter, “VendorType” is added to the import command to convert Snort rules to WAF signatures.

The parameter “VendorType” is set on SNORT only for Snort rules.

Download snort rules by using the command interface

You can download the Snort rules as a text file from the below URL:

```
https://www.snort.org/downloads/community/snort3-community-rules.tar.gz
```


Import snort rules by using the command interface

After you download, you can import the Snort rules into your appliance.

At the command prompt, type:

```
import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge [-preservedefactions]] [-sha1 <string>] [-VendorType Snort]
```

Example:

```
import appfw signatures http://www.example.com/ns/signatures.xml sig-snort -comment “signatures from snort rules” -VendorType snort
```

Arguments:

Src. URL (protocol, host, path, and file name) for the location at which to store the imported signatures object.

> **Note:**
>
> The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Mandatory argument of maximum length: 2047

Name. Name to assign to the signatures object on the Citrix ADC. Mandatory argument of maximum length: 31

Comment. Description of how to preserve information about the signatures object. Maximum Length: 255

overwrite. Overwrite any existing signatures object of the same name.

Merge. Merges existing Signature with new signature rules.

Preservedefactions. Preserves def actions of signature rules.

VendorType. Third-party vendor to generate the WAF signatures. Possible values: Snort.

Configure snort rules by using the Citrix ADC GUI

The GUI configuration for Snort rules is similar to configuring other external web application scanners like Cenzic, Qualys, Whitehat.

Follow the steps below to configure Snort:

1. Navigate to **Configuration > Security > Citrix Web App Firewall > Signatures**.
2. In the **Signatures** page, click **Add**.
3. In the **Add Signatures** page, set the following parameters to configure Snort rules.
    a) File format. Select the file format as external.

b) Import from. Select the import option as a snort file or URL to enter the URL.
c) Snort V3 Vendor. Select the check box to import Snort rules from a file or from a URL.

4. Click **Open**.



The appliance imports the Snort rules as snort-based WAF signature rules.



As a best practice, you must use filter actions to enable snort rules that you prefer to import as WAF signature rules on the appliance.



5. To confirm, click **Yes**.



6. The selected rules are enabled on the appliance.



7. Click **OK**.

Exporting a signatures object to a file

September 14, 2021

You export a signatures object to a file so that you can import it to another Citrix ADC.

To export a signatures object to a file

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to configure.
3. In the **Actions** drop-down list, select **Export**.
4. In the **Export Signatures Object** dialog box, **Local File** text box, type the path and name of the file to which you want to export the signatures object, or use the **Browse** dialog to designate a path and name.
5. Click **OK**.

Signatures editor

September 14, 2021

You can use the signatures editor to add or modify a user-defined (local) signature rule to an existing signatures object. A local signature rule has the same attributes as a default signature rule from Citrix, and it functions in the same way. You enable or disable it, and configure the signature actions for it, just as you do for a default signature.

Add a local rule if you need to protect your websites and services from a known attack that the existing signatures do not match. For example, you might discover a new type of attack and determine its characteristics by examining the logs on your web server, or you might obtain third-party information about a new type of attack.

At the heart of a signature rule are the rule *patterns*, which collectively describe the characteristics of the attack that the rule is designed to match. Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting patterns.

You might want to modify a signature rule by adding a new pattern or modifying an existing pattern to match an attack. For example, you might find out about changes to an attack, or you might determine a better pattern by examining the logs on your web server, or from third-party information.

To add or modify a local signature rule by using the Signatures Editor

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.

2. In the details pane, select the signatures object that you want to edit, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results** window, do one of the following:
   - To add a new local signature rule, click Add.
   - To modify an existing local signature rule, select that rule, and then click **Open**.
4. In the **Add Local Signature Rule** or the **Modify Local Signature Rule** dialog box, configure the actions for a signature by selecting the appropriate check boxes.
   - **Enabled.** Enables the new signature rule. If you do not select this, this new signature rule is added to your configuration, but is inactive.
   - **Block.** Blocks connections that violate this signature rule.
   - **Log.** Logs violations of this signature rule to the Citrix ADC log.
   - **Stat.** Includes violations of this signature rule in the statistics.
   - **Remove.** Strips information that matches the signature rule from the response. (Applies only to response rules.)
   - **X-Out.** Masks information that matches the signature rule with the letter X. (Applies only to response rules.)
   - **Allow Duplicates.** Allows duplicates of this signature rule in this signatures object.
5. Choose a category for the new signature rule from the **Category** drop-down list.

   You can also create a category by clicking the icon to the right of the list and using the Add Signature Rule Category dialog box to add a new category to the list. The rule you are modifying is automatically added to the new category. For instructions, see To add a signature rule category.
6. In the **LogString** text box, type a brief description of the signature rule to be used in the logs.
7. In the **Comment** text box, type a comment. (Optional)
8. Click More…, and modify the advanced options.
   a) To strip HTML comments before applying this signature rule, in the Strip Comments drop-down list choose All or Exclude Script Tag.
   b) To set CSRF Referer Header checking, in the CSRF Referrer Header checking radio button array, select either the If Present or Always radio button.
   c) To manually modify the Rule ID assigned to this local signature rule, modify the number in the Rule ID text box. The ID must be a positive integer between 1000000 and 1999999 that has not already been assigned to a local signature rule.
   d) To assign a version number to the new signature rule, modify the number in the Version Number text box.
   e) To assign a Source ID, modify the string in the Source ID text box.
   f) To specify the source, choose Local or Snort from the Source drop-down list, or click the Add icon to the right of the list and add a new source.

g) To assign a harm score to violations of this local signature rule, type a number between 1 and 10 in the Harm Score text box.
h) To assign a severity rating to this local signature rule, in the Severity drop-down list choose High, Medium, or Low, or click the Add icon to the right of the list and add a new severity rating.
i) To assign a violation type to this local signature rule, in the Type drop-down list choose Vulnerable or Warning, or click the Add icon to the right of the list and add a new violation type.

9. In the **Patterns** list, add or edit a pattern.
   - To add a pattern, click **Add**. In the **Create New Signature Rule Pattern** dialog box, add one or more patterns for your signature rule, and then click **OK**.
   - To edit a pattern, select the pattern, and then click **Open**. In the **Edit Signature Rule Pattern** dialog box, modify the pattern, and then click **OK**.

   For more information about adding or editing patterns, see Signature Rule Patterns.
10. Click **OK**.

To add a signature rule category

September 14, 2021

Putting signature rules into a category enables you to configure the actions for a group of signatures instead of for each individual signature. You might want to do so for the following reasons:

- **Ease of selection.** For example, assume that all of signature rules in a particular group protect against attacks on a specific type of web server software or technology. If your protected websites use that software or technology, you want to enable them all. If they do not, you do not want to enable any of them.
- **Ease of initial configuration.** It is easiest to set defaults for a group of signatures as a category, instead of one-by-one. You can then make any changes to individual signatures as needed.
- **Ease of ongoing configuration.** It is easier to configure signatures if you can display only those that meet specific criteria, such as belonging to a specific category.

1. Navigate to **Security** > **Web App Firewall** > **Signatures**.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen, beneath the **Filtered Results** window, click **Add**.
4. In the **Add Local Signature Rule** dialog box, click the icon to the right of the Category drop-down list.

5. In the **Add Signature Rule Category** dialog box, **New Category** text box, type a name for your new signature category. The name can consist of from one to 64 characters.
6. Click **OK**.

Signature rule patterns

September 14, 2021

You can add a pattern or modify an existing pattern to specify a string or expression that characterize an attack if the signature matches. To detect the patterns an attack exhibits, you can examine the logs on your web server. You can use a tool to observe connection data in real time, or obtain the string or expression from a third-party report about the attack.

> **Caution:**
>
> Any new pattern that you add to a signature rule is in an AND relationship with the existing patterns. Do not add a pattern to an existing signature rule if you do not want a potential attack to have to match all patterns to match the signature.

Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting pattern. Before you attempt to add a pattern that is based on a regular expression, you must make sure that you understand PCRE-format regular expressions. PCRE expressions are complex and powerful. If you do not understand how they work, you can unintentionally create a pattern that matches something that you did not want (a *false positive*) or that fails to match something that you did want (a *false negative*).

Custom signature pattern for non-default content types

The Citrix ADC Web App Firewall (WAF) now supports new location to inspect canonicalized content. By default, WAF does not block encoded payload with non-default content types. When these content types are whitelisted, and no configured action is applied, the SQL and cross-site scripting protection check do not filter SQL or cross-site scripting attacks in the encoded payloads. To resolve the issue, a user can create custom signature rule with this new location (HTTP_CANON_POST_BODY) that examines the encoded payloads for non-default content-types and if there is any SQL or cross-site scripting attack, it blocks the traffic after canonicalization of the post body.

> **Note:**
>
> That support is applicable only for HTTP requests.

If you are not already familiar with PCRE-format regular expressions, you can use the following resources to learn the basics, or for help with some specific issue:

- “Mastering Regular Expressions,” Third Edition. Copyright (c) 2006 by Jeffrey Friedl. O’Reilly Media, ISBN: 9780596528126.
- “Regular Expressions Cookbook”. Copyright (c) 2009 by Jan Goyvaerts and Steven Levithan. O’Reilly Media, ISBN: 9780596520687
- **PCRE Man page/Specification** (text/official): http://www.pcre.org/pcre.txt
- **PCRE Man Page/Specification**

http://www.gammon.com.au/pcre/index.html

- **Wikipedia PCRE entry**: http://en.wikipedia.org/wiki/PCRE
- **PCRE Mailing List**
  Pcre-dev - PCRE Development

If you need to encode non-ASCII characters in a PCRE-format regular expression, the Citrix ADC platform supports encoding of hexadecimal UTF-8 codes. For more information, see PCRE Character Encoding Format.

To configure a signature rule pattern

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results** window, either click **Add** to create a signature rule, or select an existing signature rule and click **Open**.

   > **Note:**
   >
   > You can modify only signature rules that you added. You cannot modify the default signature rules.

   Depending on your action, either the Add Local Signature Rule or the Modify Local Signature Rule dialog box appears. Both dialog boxes have the same contents.
4. Under the **Patterns window** in the dialog box, either click **Add** to add a new pattern, or select an existing pattern from the list beneath the **Add** button and click **Open**. Depending on your action, either the **Create New Signature Rule Pattern** or the **Edit Signature Rule Pattern** dialog box appears. Both dialog boxes have the same contents.
5. In the **Pattern Type** drop-down list, choose the type of connection that the pattern is intended to match.
   - If the pattern is intended to match request elements or features, such as injected SQL code, attacks on web forms, cross-site scripts, or inappropriate URLs, choose **Request**.
   - If the pattern is intended to match response elements or features, such as credit card numbers or safe objects, choose **Response**.

6. In the Location area, define the elements to examine with this pattern.
   The Location area describes what elements of the HTTP request or response to examine for this pattern. The choices that appear in the Location area depend upon the chosen pattern type. If you chose Request as the pattern type, items relevant to HTTP requests appear. If you chose Response, items relevant to HTTP responses appear.
   In addition, as you choose a value from the Area drop-down list, the remaining parts of the Location area change interactively. Following are all configuration items that might appear in this section.

   - Area. Drop-down list of elements that describe a particular portion of the HTTP connection. The choices are as follows:
     - **HTTP_ANY**. All parts of the HTTP connection.
     - **HTTP_COOKIE**. All cookies in the HTTP request headers after any cookie transformations are performed.
       Note: Does not search HTTP response "Set-Cookie:" headers.
     - **HTTP_FORM_FIELD**. Form fields and their contents, after URL decoding, percent decoding, and removal of excess whitespace. You can use the <Location> tag to further restrict the list of form field names to be searched.
     - **HTTP_HEADER**. The value portions of the HTTP header after any cross-site scripting or URL decoding transformations.
     - **HTTP_METHOD**. The HTTP request method.
     - **HTTP_ORIGIN_URL**. The origin URL of a web form.
     - **HTTP_POST_BODY**. The HTTP post body and the web form data that it contains.
     - **HTTP_RAW_COOKIE**. All HTTP request cookie, including the "Cookie:" name portion.
       Note: Does not search HTTP response "Set-Cookie:" headers.
     - **HTTP_RAW_HEADER**. The entire HTTP header, with individual headers separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
     - **HTTP_RAW_RESP_HEADER**. The entire response header, including the name and value parts of the response header after URL transformation has been done, and the complete response status. As with HTTP_RAW_HEADER, individual headers are separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
     - **HTTP_RAW_SET_COOKIE**. The entire Set-Cookie header after any URL transformations have been performed
       Note: URL transformation can change both the domain and path parts of the Set-Cookie header.
     - **HTTP_RAW_URL**. The entire request URL before any URL transformations is performed, including any query or fragment parts.
     - **HTTP_RESP_HEADER**. The value part of the complete response headers after any URL transformations have been performed.
     - **HTTP_RESP_BODY**. The HTTP response body

- **HTTP_SET_COOKIE**. All “Set-Cookie” headers in the HTTP response headers.
- **HTTP_STATUS_CODE**. The HTTP status code.
- **HTTP_STATUS_MESSAGE**. The HTTP status message.
- **HTTP_URL**. The value portion of the URL in the HTTP headers, excluding any query or fragment ports, after conversion to the UTF-* character set, URL decoding, stripping of whitespace, and conversion of relative URLs to absolute. Does not include HTML entity decoding.
- URL. Examines any URLs found in the elements specified by the Area setting. Select one of the following settings.
- **Any.** Checks all URLs.
- Literal. Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
- PCRE. Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.
- Expression. Checks URLs that match a Citrix ADC default expression.
- Field Name. Examines any form field names found in the elements specified by the Area selection. **Any.** Checks all URLs.
- Literal. Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
- PCRE. Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements or you can use the Regex Editor for assistance in constructing a regular expression that you want.
- Expression. Checks URLs that match a Citrix ADC default expression.

7. In the Pattern area, define the pattern. A pattern is a literal string or PCRE-format regular expression that defines the pattern that you want to match. The Pattern area contains the following elements:

- Match. A drop-down list of search methods that you can use for the signature. This list differs depending on whether the pattern type is Request or Response.

Request Match Types

PCRE. A PCRE-format regular expression.

> **Note:**

> When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.

- **Injection**. Directs the Web App Firewall to look for injected SQL in the specified location. The Pattern window disappears, because the Web App Firewall already has the patterns for SQL injection.
- **CrossSiteScripting**. Directs the Web App Firewall to look for cross-site scripts in the specified location. The Pattern window disappears, because the Web App Firewall already has the patterns for cross-site scripts.
- **Expression**. An expression in the Citrix ADC default expressions language is the same expression language for creating Web App Firewall policies on the Citrix ADC appliance. Although the Citrix ADC expressions language was originally developed for policy rules, it is a highly flexible general purpose language that can also be used to define a signature pattern.

When you choose Expression, the Citrix ADC Expression Editor appears beneath Pattern window. For more information about the Expression Editor and instructions on how to use it, see To add a firewall rule (expression) by using the Add Expression dialog box

Response Match Types:

```
1 -  Literal. A literal string
2 -  PCRE. A PCRE-format regular expression.
```

> Note
>
> When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.

- **Credit Card**. A built-in pattern to match one of the six supported types of credit card number.

> Note:
>
> The Expression match type is not available for Response-side signatures.

- Pattern Window (unlabeled)

In this window, type the pattern that you want to match, and fill in any additional data.

- Literal. Type the string you want to search for in the text area.
- CRE. Type the regular expression in the text area. Use the Regex Editor for more assistance in constructing the regular expression that you want, or the Regex Tokens to insert common regular expression elements at the cursor. To enable UTF-8 characters, click UTF-8.
- Expression. Type the Citrix ADC advanced expression in the text area. Use Prefix to choose the first term in your expression, or Operator to insert common operators at the cursor.

Click Add to open the Add Expression dialog box for more assistance in constructing the regular expression that you want. Click Evaluate to open the Advanced Expression Evaluator to help determine what effect your expression has.

- Offset. The number of characters to skip over before starting to match on this pattern. You use this field to start examining a string at some point other than the first character.
- Depth. How many characters from the starting point to examine for matches. You use this field to limit searches of a large string to a specific number of characters.
- Min-Length. The string to be searched must be at least the specified number of bytes in length. Shorter strings are not matched.
- Max-Length. The string to be searched must be no longer than the specified number of bytes in length. Longer strings are not matched.
- Search method. A check box labeled fastmatch. You can enable fastmatch only for a literal pattern, to improve performance.

8. Click **OK**.
9. Repeat the previous four steps to add or modify more patterns.
10. When finished adding or modifying patterns, click **OK** to save your changes and return to the Signatures pane.

> **Caution:**
>
> Until you click **OK** in the **Add Local Signature Rule** or **Modify Local Signature Rule** dialog box, your changes are not saved. Do not close either of these dialog boxes without clicking **OK** unless you want to discard your changes.

To Import and merge rules

September 14, 2021

When using the signature editor to perform an import and merge operation from the GUI, you can now see the new, updated, duplicate, and invalid rules.

The signature editor displays the following four new rows:

1. New Rules
2. Updated Rules
3. Duplicate Rules
4. Invalid Rules

The output of the New Rules Only and Updated Rules Only filters also appears in the Category filter pane of the Edit window in signature editor.

You will need to import the files from GUI to see the corresponding links for New, duplicate, invalid and updated rules.

Procedure to import signature rules:

1. In the Citrix ADC web GUI, go to **Configuration > Security > Citrix Web App Firewall Signatures**. In the Signatures window, click **Add**. Then select **File Format** > **Native**, **Import From** > **URL** and in the URL field, add the above link. If you are unable to access the URL, you can download the XML data in a text file format.
2. After you click **Open**, the signature file will open and you can see links for new rule and invalid rules.
3. If you import a 3rd party signature rule, you can see 90 new rules and 9 duplicate rules in the imported .xml file. If you are unable to access the URL, you can download the XML data in a text file format.

Signature updates in high availability deployment and build upgrades

September 14, 2021

The signature update occurs on the primary node. While the signatures are updated on the primary node, in parallel the updated files are simultaneously synchronized with the secondary node.

The Default signature is always updated first and then the rest of the user-defined signatures are updated.

Connecting to Amazon AWS

The default route NSIP is used to connect to the Amazon AWS. If there is a specific use case scenario where SNIP is used, and if there are multiple SNIPs, the first one to receive the ARP response from the hosting site will hold the route.

Signature updates during version upgrades

In case of an upgrade, if the NS has an older base version for the signatures, *Default signature is automatically updated if a newer signature version is available.

If the schema has changed, the schema version of all the signature objects gets updated when the version is upgraded.

However, for the base version of the user-defined signatures, the behavior is different in release 10.5 versus release 11.0.

In release 10.5, only the default signature was updated and the base version of the rest of the signatures remained unchanged after the build upgrade.

In release 11.0, this behavior has changed. When the appliance is upgraded to install a new build, not only the *Default signature object but all the other user-defined signatures that currently exist in the appliance are also updated and will have the same version after the build upgrade.

In both 10.5 and 11.0 release builds, if auto-update is configured, the *Default Signatures as well as all non-zero version signatures get auto-updated to the latest released signature version and will have the same base version.

Overview of security checks

September 14, 2021

The Web App Firewall advanced protections (security checks) are a set of filters designed to catch complex or unknown attacks on your protected websites and web services. The security checks use heuristics, positive security, and other techniques to detect attacks that may not be detected by signatures alone. You configure the security checks by creating and configuring an Web App Firewall profile, which is a collection of user-defined settings that tell the Web App Firewall which security checks to use and how to handle a request or response that fails a security check. A profile is associated with a signatures object and with a policy to create a security configuration.

The Web App Firewall provides twenty security checks, which differ widely in the types of attacks that they target and how complex they are to configure. The security checks are organized into the following categories:

- **Common security checks.** Checks that apply to any aspect of web security that either does not involve content or is equally applicable to all types of content.
- **HTML security checks.** Checks that examine HTML requests and responses. These checks apply to HTML-based websites and to the HTML portions of Web 2.0 sites, which contain mixed HTML and XML content.
- **XML security checks.** Checks that examine XML requests and responses. These checks apply to XML-based web services and to the XML portions of Web 2.0 sites.

The security checks protect against a wide range of types of attack, including attacks on operation system and web server software vulnerabilities, SQL database vulnerabilities, errors in the design and coding of websites and web services, and failures to secure sites that host or can access sensitive information.

All security checks have a set of configuration options, the check actions, which control how the Web App Firewall handles a connection that matches a check. Three check actions are available for all security checks. They are:

- **Block.** Block connections that match the signature. Disabled by default.
- **Log.** Log connections that match the signature, for later analysis. Enabled by default.
- **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.

A fourth check action, **Learn**, is available for more than half of the check actions. It observes traffic to a protected website or web service and uses connections that repeatedly violate the security check to generate recommended exceptions (relaxations) to the check, or new rules for the check. In addition to the check actions, certain security checks have parameters that control the rules that the check uses to determine which connections violate that check, or that configure the Web App Firewall's response to connections that violate the check. These parameters are different for each check, and they are described in the documentation for each check.

To configure security checks, you can use the Web App Firewall wizard, as described in The Web App Firewall Wizard, or you can configure the security checks manually, as described in Manual Configuration By Using the GUI. Some tasks, such as manually entering relaxations or rules or reviewing learned data, can be done only by using the GUI, not the command line. Using the wizard is usually best configuration method, but in some cases manual configuration might be easier if you are thoroughly familiar with it and simply want to adjust the configuration for a single security check.

Regardless of which method you use to configure the security checks, each security check requires that certain tasks be performed. Many checks require that you specify exceptions (relaxations) to prevent blocking of legitimate traffic before you enable blocking for that security check. You can do this manually, by observing the log entries after a certain amount of traffic has been filtered and then creating the necessary exceptions. However, it is usually much easier to enable the learning feature and let it observe the traffic and recommend the necessary exceptions.

Web App Firewall uses packet engines (PE) during processing the transactions. Each packet engine has a limit of 100K sessions which is sufficient for most deployment scenarios. However, when Web App Firewall is processing heavy traffic and the session timeout is configured at a higher value, the sessions might get accumulated. If the number of alive Web App Firewall sessions exceed the 100K per PE limit, the Web App Firewall security check violations might not be sent to the Security Insight appliance. Lowering the session timeout to a smaller value, or using sessionless mode for the security checks with sessionless URL closure or sessionless field consistency might help in preventing the sessions getting accumulated. If this is not a viable option in scenarios where transactions might require longer sessions, upgrading to a higher-end platform with more packet engine is recommended.

Support for cached AppFirewall is added, and the max session setting through the CLI per core is set to 50K sessions.

Top level protections

September 14, 2021

Four of the Web App Firewall protections are especially effective against common types of Web attacks, and are therefore more commonly used than any of the others. They are:

- **HTML Cross-Site Scripting.** Examines requests and responses for scripts that attempt to access or modify content on a different website than the one on which the script is located. When this check finds such a script, it either renders the script harmless before forwarding the request or response to its destination, or it blocks the connection.
- **HTML SQL Injection.** Examines requests that contain form field data for attempts to inject SQL commands into an SQL database. When this check detects injected SQL code, it either blocks the request or renders the injected SQL code harmless before forwarding the request to the Web server.

  Note: If both of the following conditions apply to your configuration, you must make certain that your Web App Firewall is correctly configured:

  - If you enable the HTML Cross-Site Scripting check or the HTML SQL Injection check (or both), and
  - Your protected websites accept file uploads or contain Web forms that can contain large POST body data.

  For more information about configuring the Web App Firewall to handle this case, see Configuring the Application Firewall.
- **Buffer Overflow.** Examines requests to detect attempts to cause a buffer overflow on the Web server.
- **Cookie Consistency.** Examines cookies returned with user requests to verify that they match the cookies your Web server set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the Web server.

The Buffer Overflow check is simple; you can usually enable blocking for it immediately. The other three top-level checks are considerably more complex and require configuration before you can safely use them to block traffic. Citrix strongly recommends that, rather than attempting to configure these checks manually, you enable the learning feature and allow it to generate the necessary exceptions.

HTML cross-site scripting check

September 14, 2021

The HTML Cross-Site Scripting (cross-site scripting) check examines both the headers and the POST bodies of user requests for possible cross-site scripting attacks. If it finds a cross-site script, it either modifies (*transforms*) the request to render the attack harmless, or blocks the request.

> **Note:**
>
> The HTML Cross-Site Scripting (cross-site scripting) check works only for content type, content length, and so forth. It does not work for the cookie. Also ensure to have the 'checkRequestHeaders' option enabled in your Web Application Firewall profile.

You can prevent misuse of the scripts on your protected websites by using the HTML Cross-Site Scripting scripts that violate the *same origin rule*, which states that scripts must not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of JavaScript-enhanced web content that violates the same origin rule. If you enable the HTML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity.

The Web App Firewall offers various action options for implementing HTML Cross-Site Scripting protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, you also have the option to **Transform cross-site scripts** to render an attack harmless by entity encoding the script tags in the submitted request. You can configure Check complete URLs for cross-site scripting parameter to specify if you want to inspect not just the query parameters but the entire URL to detect cross-site scripting attack. You can configure **InspectQueryContentTypes** parameter to inspect request query portion for the cross-site scripting attack for the specific content-types.

You can deploy relaxations to avoid false positives. The Web App Firewall learning engine can provide recommendations for configuring relaxation rules.

To configure an optimized HTML Cross-Site Scripting protection for your application, configure one of the actions:

- **Block**—If you enable block, the block action is triggered if the cross-site scripting tags are detected in the request.
- **Log**—If you enable the log feature, the HTML Cross-Site Scripting check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each header or form field in which the cross-site scripting violation was detected. However, only one message is generated when the request is blocked. Similarly, 1 log message per request is generated for the transform operation, even when cross-site scripting tags are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests

are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

- **Stats**—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you must configure new relaxation rules or modify the existing ones.
- **Learn**—If you are not sure which relaxation rules might be ideally suited for your application, you can use the learn feature to generate HTML Cross-Site Scripting rule recommendations based on the learned data. The Web App Firewall learning engine monitors the traffic and provides learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.
- **Transform cross-site scripts**—If enabled, the Web App Firewall makes the following changes to requests that match the HTML Cross-Site Scripting check:
  - Left angle bracket (<) to HTML character entity equivalent (<)
  - Right angle bracket (>) to HTML character entity equivalent (>)

This ensures that browsers do not interpret unsafe html tags, such as `<script>`, and thereby run malicious code. If you enable both request-header checking and transformation, any special characters found in request headers are also modified. If the scripts on your protected website contain cross-site scripting features, but your website does not rely upon those scripts to operate correctly, you can safely disable blocking and enable transformation. This configuration ensures that no legitimate web traffic is blocked, while stopping any potential cross-site scripting attacks.

- **Check complete URLs for cross-site scripting**. If checking of complete URLs is enabled, the Web App Firewall examines entire URLs for HTML cross-site scripting attacks instead of checking just the query portions of URLs.
- **Check Request headers**. If Request header checking is enabled, the Web App Firewall examines the headers of requests for HTML cross-site scripting attacks, instead of just URLs. If you use the GUI, you can enable this parameter in the Settings tab of the Web App Firewall profile.
- **InspectQueryContentTypes**. If Request query inspection is configured, the App Firewall examines the query of requests for cross-site scripting attacks for the specific content-types. If you use the GUI, you can configure this parameter in the Settings tab of the App Firewall profile.

> **Important:**
>
> As part of the streaming changes, the Web App Firewall processing of the cross-site scripting tags has changed. This change is applicable to 11.0 builds onwards. This change is also pertinent for the enhancement builds of 10.5.e that support request side streaming. In earlier releases, presence of either open bracket (<), or close bracket (>), or both open and close brackets (<>) was flagged as cross-site scripting Violation. The behavior has changed in the builds that include

> support for request side streaming. Only the close bracket character (>) is no longer considered as an attack. Requests are blocked even when an open bracket character (<) is present, and is considered as an attack. The Cross-site scripting attack gets flagged.

Cross-site scripting Fine grained Relaxations

The Web App Firewall gives you an option to exempt a specific form field, header, or Cookie from cross-site scripting inspection check. You can completely bypass the inspection for one or more of these fields by configuring relaxation rules.

The Web App Firewall allows you to implement tighter security by fine-tuning the relaxation rules. An application might require the flexibility to allow specific patterns, but configuring a relaxation rule to bypass the security inspection might make the application vulnerable to attacks, because the target field is exempted from inspection for any cross-site scripting attack patterns. Cross-site scripting fine grained relaxation provides the option to allow specific attributes, tags, and patterns. The rest of the attributes, tags, and patterns are blocked. For example, the Web App Firewall currently has a default set of more than 125 denied patterns. Because hackers can use these patterns in Cross-site script attacks, the Web App Firewall flags them as potential threats. You can relax one or more patterns that are considered safe for the specific location. The rest of the potentially dangerous cross-site scripting patterns are still checked for the target location and continue to trigger the security check violations. You now have much tighter control.

The commands used in relaxations have optional parameters for **Value Type** and **Value Expression**. The value type can be left blank or you have an option to select **Tag** or **Attribute** or **Pattern**. If you leave the value type blank, the configured field of the specified URL is exempted from the Cross-Site Scripting check inspection. If you select a value type, you must provide a value expression. You can specify whether the value expression is a regular expression or a literal string. When the input is matched against the allowed and denied list, only the specified expressions configured in the relaxation rules are exempted.

The Web App Firewall has the following cross-site scripting built-in lists:

1. **cross-site scripting Allowed Attributes**: There are 52 defaults allowed attributes, such as, **abbr**, **accesskey**, **align**, **alt**, **axis**, **bgcolor**, **border**, **cell padding**, **cell spacing**, **char**, **charoff**, **charset** and so forth
2. **cross-site scripting Allowed Tags**: There are 47 defaults allowed tags, such as, **address**, **basefont**, **bgsound**, **big**, **blockquote**, **bg**, **br**, **caption**, **center**, **cite**, **dd**, **del** and so forth
3. **cross-site scripting Denied Patterns**: There are 129 defaults denied patterns, such as, **FSCommand**, **javascript:**, **onAbort**, **onActivate** and so forth

> **Warning**
>
> Web App Firewall action URLs are regular expressions. When configuring HTML cross-site script-

> ing relaxation rules, you can specify **Name**, and **Value Expression** to be literal or RegEx. Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the rule that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML Cross-Site Scripting check would otherwise have blocked.

Points to Consider:

- Value expression is an optional argument. A field name might not have any value expression.
- A field name can be bound to multiple value expressions.
- Value expressions must be assigned a value type. The cross-site scripting value type can be: 1) Tag, 2) Attribute, or 3) Pattern.
- You can have multiple relaxation rules per field name/URL combination
- The form field names and the action URLs are not case sensitive.

Using the Command Line to Configure the HTML Cross-Site Scripting check

To configure HTML Cross-Site Scripting check actions and other parameters by using the command line

If you use the command-line interface, you can enter the following commands to configure the HTML Cross-Site Scripting Check:

- set appfw profile “Parameter descriptions provided at bottom of the page.”)
- `<name> -crossSiteScriptingAction (([block] [learn] [log] [stats])| [** none**])`
- set appfw profile “Parameter descriptions provided at bottom of the page.”)
- `<name> **-crossSiteScriptingTransformUnsafeHTML** (ON | OFF)`
- set appfw profile Parameter descriptions provided at bottom of the page.
- `<name> -crossSiteScriptingCheckCompleteURLs (ON | OFF)`
- set appfw profile Parameter descriptions provided at bottom of the page.
- `<name> - checkRequestHeaders (ON | OFF)` Parameter descriptions provided at bottom of the page.”
- `<name> - CheckRequestQueryNonHtml (ON | OFF)` Parameter descriptions provided at bottom of the page.”

To configure an HTML Cross-Site Scripting check relaxation rule by using the command line

Use the bind or unbind command to add or delete binding, as follows:

- `bind appfw profile <name> -crossSiteScripting <String> [isRegex (REGEX | NOTREGEX)] <formActionURL> [-location <location>] [-valueType (Tag|`


```
Attribute|Pattern)[<valueExpression>] [-isValueRegex (REGEX | NOTREGEX)
]]
```

- `unbind appfw profile <name> -crossSiteScripting <String> <formActionURL > [-location <location>] [-valueType (Tag |Attribute|Pattern)[<valueExpression >]]`

Using the GUI to configure the HTML cross-site scripting check

In the GUI, you can configure the HTML Cross-Site Scripting check in the pane for the profile associated with your application.

To configure or modify the HTML Cross-Site Scripting check by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a. If you want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for the HTML Cross-Site Scripting, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the **Security Check** pane.

b. If you want to configure more options for this security check, double-click **HTML Cross-Site Scripting**, or select the row and click **Action Settings**, to display the following options:

Transform cross-site scripts—Transform unsafe script tags.

Check complete URLs for Cross-site scripting—Instead of checking just the query part of the URL, check the complete URL for cross-site script violations.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To enable or disable the **Check request Header** setting, in the **Advanced Settings** pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check Request Headers** check box. Click **OK**. You can either use the X icon at the top right hand side of the Profile Settings pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **Application Firewall** > **Profile**.

To enable or disable the **Check request Query Non HTML** setting, in the **Advanced Settings** pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check request Query Non HTML** check box. Click **OK**. You can either use the X icon at the top right hand side of the **Profile Settings**

pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **App Firewall > Profile**.

To configure an HTML Cross-Site Scripting relaxation rule by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **HTML Cross-Site Scripting** entry, or select it and click **Edit**.
4. In the **HTML Cross-Site Scripting Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

> **Note**
>
> When you add a new rule, the **Value Expression** field is not displayed unless you select **Tag** or **Attribute** or **Pattern** option in the **Value Type** Field.

To manage HTML Cross-Site Scripting relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **HTML Cross-Site Scripting** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the Cross-Site Scripting patterns by using the GUI

You can use the GUI to view or customize the default list of cross-site scripting allowed attributes or allowed tags. You can also view or customize the default list of cross-site scripting denied Patterns.

The default lists are specified in **Application Firewall** > **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default cross-site scripting allowed and denied list specified in the Default Signatures object will be used by the profile for the Cross-Site Scripting security check processing. The Tags, Attributes, and Patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the allowed or denied lists in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized allowed and denied lists.

1. To view default cross-site scripting patterns:

a. Navigate to **Application Firewall** > **Signatures**, select **Default Signatures**, and click **Edit**. Then click **Manage SQL/cross-site scripting Patterns**.

The **Manage SQL/cross-site scripting Paths** table shows following three rows pertaining to cross-site scripting:

```
xss/allowed/attribute
```

```
xss/allowed/tag
```


`xss/denied/pattern`

b. Select a row and click **Manage Elements** to display the corresponding cross-site scripting Elements (Tag, Attribute, Pattern) used by the Web App Firewall **Cross-Site Scripting** check.

1. **To customize cross-site scripting Elements**: You can edit the User-Defined signature object to customize the allowed Tag, allowed Attributes and denied Patterns. You can add new entries or remove the existing ones.

a. Navigate to **Application Firewall** > **Signatures**, highlight the target User-defined signature, and click **Edit**. Click **Manage SQL/cross-site scripting Patterns** to display the **Manage SQL/cross-site scripting paths** table.

b. Select the target cross-site scripting row.

i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding cross-site scripting element.

ii. Click **Remove** to remove the selected row.

> **Warning:**
>
> You must be careful before you remove or modify any default cross-site scripting element, or delete the cross-site scripting path to remove the entire row. The signature rules and the Cross-Site Scripting security check rely on these elements for detecting attacks to protect your applications. Customizing the cross-site scripting Elements can make your application vulnerable to Cross-Site Scripting attacks if the required pattern is removed during editing.

Using the Learn Feature with the HTML Cross-Site Scripting Check

When the “learn” action is enabled, the Citrix Web App Firewall learning engine monitors the traffic and learns the cross-site scripting URL violations. You can periodically inspect cross-site scripting URL rules and deploy it for false positive scenarios.

> **Note:**
>
> In a cluster configuration, all nodes must be of the same version to deploy the cross-site scripting URL rules.

HTML Cross-Site Scripting Learning enhancement. An Web App Firewall learning enhancement was introduced in release 11.0 of the Citrix ADC software. To deploy fine grained HTML Cross-Site Scripting relaxation, the Web App Firewall offers fine grained HTML Cross-Site Scripting learning. The learning engine makes recommendations regarding the observed Value Type (Tag, Attribute, Pattern) and the corresponding Value expression observed in the input fields. In addition to checking the blocked requests to determine whether the current rule is too restrictive and needs to be relaxed, you can review the rules generated by the learning engine to determine which value type and value expressions are triggering violations and need to be addressed in the relaxation rules.

> **Note:**
>
> The Web App Firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields from HTML Cross-Site Scripting inspection.
>
> **Tip**
>
> Cross-site scripting tags which are longer than 12 characters are not learned or logged correctly.
>
> If you need a larger tag length for learning, you can add a large non-appearing tag in the **AS_cross-site scripting_ALLOWED_TAGS_LIST** for length 'x'.

To view or use learned data by using the command line interface

At the command prompt, type one of the following commands:

- `show appfw learningdata <profilename> crossSiteScripting`
- `rm appfw learningdata <profilename> -crossSiteScripting <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> **crossSiteScripting*`

To view or use learned data by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the **HTML Cross-Site Scripting** entry in the Learned Rules table and double-click it to access the learned rules. The table displays the **Field Name**, A**ction URL**, **Value Type**, **Value**, and **Hits** columns. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the **HTML Cross-Site Scripting** entry in the Learned Rules table and clicking **Visualizer** to get a consolidated view of all the learned violations. The visualizer makes it easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the log feature with the HTML Cross-Site Scripting check

When the log action is enabled, the HTML Cross-Site Scripting security check violations are logged in the audit log as **APPFW_cross-site scripting** violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the HTML Cross-Site Scripting violations:

```
Shell
tail -f /var/log/ns.log | grep APPFW_cross-site scripting
```

Example of a Cross-Site Scripting security check violation log message in CEF log format:

```
1 Jul 11 00:45:51 <local0.info> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
     .0|APPFW|**APPFW_cross-site scripting**|6|src=10.217.253.62
     geolocation=Unknown spt=4840 method=GET request=http://aaron.
     stratum8.net/FFC/CreditCardMind.html?abc\=%3Cdef%3E msg=**Cross-site
      script check failed for field abc=\"Bad tag: def"** cn1=133 cn2=294
      cs1=pr_ffc cs2=PPE1 cs3=eUljypvLa0BbabwfGVE52Sewg9U0001 cs4=ALERT
     cs5=2015 act=**not blocked**
2 <!--NeedCopy-->
```

Example of a Cross-Site Scripting security check violation log message in Native log format showing transform action

```
1 Jul 11 01:00:28 <local0.info> 10.217.31.98 07/11/2015:01:00:28 GMT ns
     0-PPE-0 : default APPFW **APPFW_cross-site scripting** 132 0 :
     10.217.253.62 392-PPE0 eUljypvLa0BbabwfGVE52Sewg9U0001 pr_ffc http:
     //aaron.stratum8.net/FFC/login.php?login_name=%3CBOB%3E&passwd=&
     drinking_pref=on &text_area=&loginButton=ClickToLogin&as_sfid=
     AAAAAAVFqmYL68IGvkrcn2pzehjfIkm5E6EZ9FL8YLvIW_41AvAATuKYe9N7uGThSpEAxbb0iBx55j
     -FC4llF **Cross-site script special characters seen in fields <
     transformed>**
2 <!--NeedCopy-->
```

Access the log messages by using the GUI

The Citrix GUI includes a useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **HTML Cross-Site Scripting** row and click **Logs**. When you access the

logs directly from the HTML Cross-Site Scripting check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.

- You can also access the Syslog Viewer by navigating to **Citrix ADC** > **System** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **HTML Cross-Site Scripting** check, filter by selecting **APPFW** in the drop-down list options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_cross-site scripting** check box and click the **Apply** button, only log messages pertaining to the HTML Cross-Site Scripting security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** and so forth appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click to Deploy functionality is available only in the GUI. You can use the Syslog Viewer to not only view the logs but also to deploy HTML Cross-Site Scripting relaxation rules based on the log messages for the Web App Firewall security check violations. The log messages must be in CEF log format for this operation. Click to deploy functionality is available only for log messages that are generated by the block (or not block) action. You cannot deploy a relaxation rule for a log message about the transform operation.

To deploy a relaxation rule from the Syslog Viewer, select the log message. A check box appears in the upper right corner of the Syslog Viewer box of the selected row. Select the check box, and then select an option from the **Action** list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options.

The HTML Cross-Site Scripting rules that are deployed by using the **Click to Deploy** option do not include the fine grain relaxation recommendations.

Configure click to deploy feature by using the GUI

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Select the **APP_cross-site scripting** as the **Event Type** to filter corresponding log messages.
3. Select the check box to identify the rule to deploy.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Statistics for the HTML Cross-Site Scripting violations

When the stats action is enabled, the counter for the HTML Cross-Site Scripting check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 HTML Cross-Site Scripting violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display HTML Cross-Site Scripting check statistics by using the command line

At the command prompt, type:

```
> sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> **stat appfw profile** <profile name>
```

Display HTML Cross-Site Scripting statistics by using the GUI

1. Navigate to **Security** > **Application Firewall > Profiles > Statistics**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about HTML Cross-Site Scripting violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

- **Built-in Support for HTML Cross-Site Scripting attack Protection**—The Citrix Web App Firewall protects against Cross-Site Scripting attacks by monitoring a combination of allowed attributes and tags, and denied patterns in the received payload. All the built-in default allowed tags, allowed attributes and denied patterns used by the cross-site scripting check are specified in the /netscaler/default_custom_settings.xml file.
- **Customizatio**n—You can change the default list of tags, attributes, and patterns to customize the Cross-Site Scripting security check inspection for the specific needs of your application. Make a copy of the default signature object, modify existing entries, or add new ones. Bind this signature object to your profile to make use of the customized configuration.
- **Hybrid Security Model**—Both signatures and deep security protections use the SQL/cross-site scripting patterns specified in the signature object that is bound to the profile. If no signature object is bound to the profile, the SQL/cross-site scripting patterns present in the default signature object are used.

- **Transform**—Note the following about the transform operation:

The transform operation works independently of the other Cross-Site Scripting action settings. If transform is enabled and block, log, stats, and learn are all disabled, cross-site scripting tags are transformed.

If the block action is enabled, it takes precedence over the transform action.

- **Fine Grained Relaxation and Learning**. Fine tune the relaxation rule to relax a subset of cross-site scripting elements from security check inspection but detect the rest. The learning engine recommends a specific value type and value expressions based on the observed data.
- **Click to Deploy**—Select one, or multiple cross-site scripting violation log messages in the syslog viewer and deploy them as relaxation rules.
- **Charset**—The default charset for the profile must be set based on the need of the application. By default, the profile charset is set to English US (ISO-8859-1). If a request is received without the specified charset, the Web App Firewall processes the request as if it is ISO-8859-1. The open bracket character (<) or the close bracket character (>) will not get interpreted as cross-site scripting tags if these characters are encoded in other charsets. For example, if a request contains a UTF-8 character string "**%uff1cscript%uff1e**" but the charset is not specified on the request page, the cross-site scripting violation might not get triggered unless the default charset for the profile is specified as Unicode.

HTML SQL injection check

September 14, 2021

Many web applications have web forms that use SQL to communicate with relational database servers. Malicious code or a hacker can use an insecure web form to send SQL commands to the web server. The Web App Firewall HTML SQL Injection check provides special defenses against injection of unauthorized SQL code that might break security. If the Web App Firewall detects unauthorized SQL code in a user request, it either transforms the request, to render the SQL code inactive, or blocks the request. The Web App Firewall examines the request payload for injected SQL code in three locations: 1) POST body, 2) headers, and 3) cookies. To examine a query portion in requests for injected SQL code, please configure an application firewall profile setting 'InspectQueryContentTypes' for the specific content-types.

A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch SQL attacks. You can add new patterns, and you can edit the default set to customize the SQL check inspection. The Web App Firewall offers various action options for implementing SQL Injection protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, the Web App Firewall profile also offers the option to **transform SQL special characters** to render an

attack harmless.

In addition to actions, there are several parameters that can be configured for SQL injection processing. You can check for **SQL wildcard characters**. You can change the SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the payload. The **SQL Comments Handling parameter** gives you an option to specify the type of comments that need to be inspected or exempted during SQL Injection detection.

You can deploy relaxations to avoid false positives. The Web App Firewall learning engine can provide recommendations for configuring relaxation rules.

The following options are available for configuring an optimized SQL Injection protection for your application:

Block—The block action is triggered only if the input matches the SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the SQL injection type, a request is not blocked if it contains no key words, even if SQL special characters are detected in the input. Such a request is blocked if the SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.

Log—If you enable the log feature, the SQL Injection check generates log messages indicating the actions that it takes. If the block action is disabled, a separate log message is generated for each input field in which the SQL violation was detected. However, only one message is generated when the request is blocked. Similarly, one log message per request is generated for the transform operation, even when SQL special characters are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

Stats—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

Learn—If you are not sure which SQL relaxation rules might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The Web App Firewall learning engine monitors the traffic and provides SQL learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.

Transform SQL special characters—The Web App Firewall considers three characters, Single straight quote (‘), Backslash (), and Semicolon (;) as special characters for SQL security check processing. The SQL Transformation feature modifies the SQL Injection code in an HTML request to ensure that the request is rendered harmless. The modified HTML request is then sent to the server. All default transformation rules are specified in the /netscaler/default_custom_settings.xml file.

The transform operation renders the SQL code inactive by making the following changes to the request:

- Single straight quote (‘) to double straight quote (“).
- Backslash () to double backslash ().
- Semicolon (;) is dropped completely.

These three characters (special strings) are necessary to issue commands to an SQL server. Unless an SQL command is prefaced with a special string, most SQL servers ignore that command. Therefore, the changes that the Web App Firewall performs when transformation is enabled prevent an attacker from injecting active SQL. After these changes are made, the request can safely be forwarded to your protected website. When web forms on your protected website can legitimately contain SQL special strings, but the web forms do not rely on the special strings to operate correctly, you can disable blocking and enable transformation to prevent blocking of legitimate web form data without reducing the protection that the Web App Firewall provides to your protected websites.

The transform operation works independently of the **SQL Injection Type** setting. If transform is enabled and the SQL Injection type is specified as the SQL keyword, SQL special characters are transformed even if the request does not contain any keywords.

> **Tip**
>
> You normally enable either transformation or blocking, but not both. If the block action is enabled, it takes precedence over the transform action. If you have blocking enabled, enabling transformation is redundant.

Check for SQL Wildcard Characters—Wild card characters can be used to broaden the selections of a SQL (SQL-SELECT) statement. These wild card operators can be used with **LIKE** and **NOT LIKE** operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

SELECT * from customer WHERE name like “%D%”:

The following example combines the operators to find any salary values that have 0 in the second and third place.

SELECT * from customer WHERE salary like ‘_00%’:

Different DBMS vendors have extended the wildcard characters by adding extra operators. The Citrix Web App Firewall can protect against attacks that are launched by injecting these wildcard characters.

The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening bracket ([), and closing bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures**:

- <wildchar type=" LITERAL" >%</wildchar>
- <wildchar type=" LITERAL" >_</wildchar>
- <wildchar type=" LITERAL" >^</wildchar>
- <wildchar type=" LITERAL" >[</wildchar>
- <wildchar type=" LITERAL" >]</wildchar>

Wildcard characters in an attack can be PCRE, like [^A-F]. The Web App Firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

> **Note:**
>
> The SQL wildcard character check is different from the SQL special character check. This option must be used with caution to avoid false positives.

Check Request Containing SQL Injection Type—The Web App Firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

- **SQL Special Character and Keyword**—Both an SQL keyword and an SQL special character must be present in the input to trigger SQL violation. This least restrictive setting is also the default setting.
- **SQL Special Character**—At least one of the special characters must be present in the input to trigger SQL violation.
- **SQL key word**—At least one of the specified SQL keywords must be present in the input to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.
- **SQL Special Character or Keyword**—Either the key word or the special character string must be present in the input to trigger the security check violation.

> **Tip:**
>
> If you configure the Web App Firewall to check for inputs that contain an SQL special character, the Web App Firewall skips web form fields that do not contain any special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this option can significantly reduce the load on the Web App Firewall and speed up processing without placing your protected websites at risk.

SQL comments handling—By default, the Web App Firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure

the Web App Firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:

- **ANSI**—Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases. For example:
  - – (Two Hyphens) - This is a comment that begins with two hyphens and ends with end of line.
  - {} - Braces (Braces enclose the comment. The { precedes the comment, and the } follows it. Braces can delimit single- or multiple-line comments, but comments cannot be nested)
  - `/**/: C style comments (Does not allow nested comments). Please note /*! <comment that begin with slash followed by asterisk and exclamation mark is not a comment > */`
  - MySQL Server supports some variants of C-style comments. These enable you to write code that includes MySQL extensions, but is still portable, by using comments of the following form: `/*! MySQL-specific code */`
  - . #: Mysql comments: This is a comment that begins with # character.
- **Nested**—Skip nested SQL comments, which are normally used by Microsoft SQL Server. For example; – (Two Hyphens), and **/* */** (Allows nested comments)
- **ANSI/Nested**—Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.
- **Check all Comments**—Check the entire request for injected SQL without skipping anything. This is the default setting.

> **Tip**
>
> Usually, you must not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they might indicate an attempt to breach security on that server.

Check Request headers—Enable this option if, in addition to examining the input in the form fields, you want to examine the request headers for HTML SQL Injection attacks. If you use the GUI, you can enable this parameter in the **Advanced Settings** -> **Profile Settings** pane of the Web App Firewall profile.

> **Note:**
>
> If you enable the Check Request header flag, you might have to configure a relaxation rule for the **User-Agent** header. Presence of the SQL keyword **like** and SQL special character semi-colon (**;**) might trigger false positive and block requests that contain this header.
> **Warning**

> If you enable both request header checking and transformation, any SQL special characters found in the headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;). Enabling both Request header checking and transformation simultaneously might cause errors.

InspectQueryContentTypes — Configure this option if you want to examine the request query portion for SQL Injection attacks for the specific content-types. If you use the GUI, you can configure this parameter in the **Advanced Settings** -> **Profile Settings** pane of the App Firewall profile.

SQL Fine grained Relaxations

The Web App Firewall gives you an option to exempt a specific form field, header, or Cookie from the SQL Injection inspection check. You can completely bypass the inspection for one or more of these fields by configuring the relaxation rules for the SQL Injection check.

The Web App Firewall allows you to implement tighter security by fine-tuning the relaxation rules. An application might require the flexibility to allow specific patterns, but configuring a relaxation rule to bypass the security inspection might make the application vulnerable to attacks, because the target field is exempted from inspection for any SQL attack patterns. SQL fine grained relaxation provides the option to allow specific patterns and block the rest. For example, the Web App Firewall currently has a default set of more than 100 SQL keywords. Because hackers can use these keywords in SQL Injection attacks, the Web App Firewall flags them as potential threats. You can relax one or more keywords that are considered safe for the specific location. The rest of the potentially dangerous SQL keywords are still checked for the target location and continue to trigger the security check violations. You now have much tighter control.

The commands used in relaxations have optional parameters for **Value Type** and **Value Expression**. You can specify whether the value expression is a regular expression or a literal string. The value type can be left blank or you have an option to select **Keyword** or **SpecialString** or **WildChar**.

> **Warning:**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter or wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML SQL Injection check would otherwise have blocked.

Points to Consider:

- Value expression is an optional argument. A field name might not have any value expression.
- A field name can be bound to multiple value expressions.

- Value expressions must be assigned a value type. The SQL value type can be: 1) Keyword, 2) SpecialString, or 3) WildChar.
- You can have multiple relaxation rules per field name/URL combination.

Using the Command Line to Configure the SQL Injection Check

To configure SQL Injection actions and other parameters by using the command line:

In the command line interface, you can use either the **set appfw profile** command or the **add appfw profile** command to configure the SQL Injection protections. You can enable the block, learn, log, stats actions and specify whether you want to transform the special characters used in SQL Injection attack strings to disable the attack. Select the type of SQL attack pattern (key words, wildcard characters, special strings) you want to detect in the payloads, and indicate whether you want the Web App Firewall to also inspect the request Headers for SQL Injection violations. Use the **unset appfw profile** command to revert the configured settings back to their defaults. Each of the following commands sets only one parameter, but you can include multiple parameters in a single command:

- set application firewall profile "Parameter descriptions provided at the bottom of the page."
- `<name> -SQLInjectionAction ([block] [learn] [log] [stats])| [none])`
- set application firewall profile "Parameter descriptions provided at the bottom of the page."
- `<name> -SQLInjectionTransformSpecialChars (**ON** | OFF)`
- set application firewall profile "Parameter descriptions provided at the bottom of the page."
- `<name> -**SQLInjectionCheckSQLWildChars** (**ON** |**OFF**)`
- set application firewall profile "Parameter descriptions provided at the bottom of the page."
- `**<name> -**SQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])`
- **set application firewall profile** "Parameter descriptions provided at the bottom of the page."
- `<name> -**SQLInjectionParseComments** ([**checkall**] | [**ansi|nested **] | [**ansinested**])`
- **set application firewall profile "Parameter descriptions provided at the bottom of the page."
- `<name> -CheckRequestHeaders (ON | OFF)` Parameter descriptions provided at the bottom of the page.
- `<name> - CheckRequestQueryNonHtml (ON | OFF)` Parameter descriptions provided at the bottom of the page.

To configure a SQL Injection relaxation rule by using the command interface

Use the bind or unbind command to add or delete binding, as follows:

- `bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX| NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor |SpecialString|Wildchar)[<valueExpression>][-isValueRegex (REGEX | NOTREGEX)]]`

- `unbind appfw profile <name> -SQLInjection <String> <formActionURL> [-location <location>] [-valueTyp (Keyword|SpecialString|Wildchar)[<valueExpression>]]`

> **Note:**
>
> You can find the list of SQL keywords from the default signature file contents by viewing the view signature object, which has list of SQL key words and SQL special characters.

Using the GUI to Configure the SQL Injection Security Check

In the GUI, you can configure the SQL Injection security check in the pane for the profile associated with your application.

To configure or modify the SQL Injection check by using the GUI

1. Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a. If you want to enable or disable Block, Log, Stats, and Learn actions for HTML SQL Injection, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the **Security Check** pane.

b. If you want to configure more options for this security check, double click HTML SQL Injection, or select the row and click **Action Settings**, to display the following options:

Transform SQL Special character—Transform any SQL Special characters in the request.

Check for SQL Wildcard Characters—Consider SQL Wildcard characters in the payload to be attack patterns.

Check Request Containing—Type of SQL injection (SQLKeyword, SQLSplChar, SQLSplCharANDKeyword, or SQLSplCharORKeyword) to check.

SQL Comments Handling—Type of comments (Check All Comments, ANSI, Nested, or ANSI/Nested) to check.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure an SQL Injection relaxation rule by using the GUI

- Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.

- In the **Advanced Settings** pane, click **Relaxation Rules**.
- In the Relaxation Rules table, double-click the **HTML SQL Injection** entry, or select it and click **Edit**.
- In the **HTML SQL Injection Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

> **Note**
>
> When you add a new rule, the **Value Expression** field is not displayed unless you select **Keyword** or **SpecialString** or **WildChar** option in the **Value Type** Field.

To manage SQL injection relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **HTML SQL Injection** row and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

View or customize injection patterns by using the GUI

You can use the GUI to view or customize the injection patterns.

The default SQL patterns are specified in the default signatures file. If you do not bind any signature object to your profile, the default Injection patterns specified in the default signatures object will be used by the profile for the command injection security check processing. The rules and patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these patterns, make a copy of the default sSignatures object to create a User-Defined signature object. Make changes in the command injection patterns in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized patterns.

For more information, see Signatures

To view the default injection patterns by using the GUI:

1. Navigate to **Application Firewall > Signatures**, select ***Default Signatures**, and click **Edit**.



1. Click **Manage CMD/SQL/XSS patterns**. The **Manage SQL/cross-site scripting paths** table shows patterns pertaining to CMD/SQL/XS injection:



1. Select a row and click **Manage Elements** to display the corresponding injection patterns (keywords, special strings, transformation rules or the wildcard characters) used by the Web App Firewall command injection check.

Using the Learn Feature with the SQL Injection Check

When the learn action is enabled, the Web App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as an SQL Injection relaxation rule.

SQL Injection Learning enhancement—An Web App Firewall learning enhancement was introduced in release 11.0 of the Citrix ADC software. To deploy fine grained SQL Injection relaxation, the Web App Firewall offers fine grained SQL Injection learning. The learning engine makes recommendations regarding the observed Value Type (keyword, SpecialString, Wildchar) and the corresponding Value expression observed in the input fields. In addition to checking the blocked requests to determine whether the current rule is too restrictive and needs to be relaxed, you can review the rules generated by the learning engine to determine which value type and value expressions are triggering violations and need to be addressed in the relaxation rules.

> **Important**
>
> The Web App Firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields from SQL Injection inspection.
>
> **Note** To bypass the SQL check-in the User-Agent header, use the following relaxation rule:
>
> ```
> bind appfw profile your_profile_name -SQLInjection User-Agent ".*"-
> location HEADER
> ```

To view or use learned data by using the command line interface

At the command prompt, type one of the following commands:

- `show appfw learningdata <profilename> SQLInjection`
- `rm appfw learningdata <profilename> -SQLInjection <string> <formActionURL > [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> SQLInjection`

To view or use learned data by using the GUI

1. Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the **HTML SQL Injection** entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the **HTML SQL Injection** entry in the Learned Rules table and clicking **Visualizer** to get a consolidated view of

all the learned violations. The visualizer makes it easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the Log Feature with the SQL Injection Check

When the log action is enabled, the HTML SQL Injection security check violations are logged in the audit log as **APPFW_SQL** violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the **/var/log/** folder to access the log messages pertaining to the SQL Injection violations:

```
> Shell
```

```
## tail -f /var/log/ns.log | grep APPFW_SQL
```

Example of an HTML SQL Injection log message when the request is transformed

```
1 Jun 26 21:08:41 <local0.info> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
      .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=54001
       method=GET request=http://aaron.stratum8.net/FFC/login.php?
      login_name\=%27+or&passwd\=and+%3B&drinking_pref\=on&text_area\=
      select+*+from+%5C+%3B&loginButton\=ClickToLogin&as_sfid\=
      AAAAAAXjnGN5gLH-hvhTOpIySEIqES7BjFRs5Mq0fwPp-3ZHDi5yWlRWByj0cVbMyy-
      Ens2vaaiULKOcUri4OD4kbXWwSY5s7I3QkDsrvIgCYMC9BMvBwY2wbNcSqCwk52lfE0k
      %3D&as_fid\=feeec8758b41740eedeeb6b35b85dfd3d5def30c msg= Special
      characters seen in fields cn1=74 cn2=762 cs1=pr_ffc cs2=PPE1 cs3=9
      ztIlf9p1H7p6Xtzn6NMygTv/QM0002 cs4=ALERT cs5=2015 act=transformed
2 <!--NeedCopy-->
```

Example of an HTML SQL Injection log message when the post request is blocked

```
1 Jun 26 21:30:34 <local0.info> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
      .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=9459
      method=POST request=http://aaron.stratum8.net/FFC/login_post.php msg
      =SQL Keyword check failed for field text_area\="(')" cn1=78 cn2=834
      cs1=pr_ffc cs2=PPE1 cs3=eVJMMPtZ2XgylGrHjkx3rZLfBCI0002 cs4=ALERT
      cs5=2015 act=blocked
2 <!--NeedCopy-->
```


> **Note**
>
> As part of the streaming changes in 10.5.e build (enhancement builds) and 11.0 build onwards, we now process the input data in blocks. RegEx pattern matching is now restricted to 4K for contiguous character string matching. With this change, the SQL violation log messages might include different information compared to the earlier builds. The keyword and special character in the input can be separated by many bytes. We now keep track of the SQL keywords and special strings when processing the data, instead of buffering the entire input value. In addition to the field name, the log message now includes the SQL keyword, or the SQL special character, or both the SQL keyword and the SQL special character, as determined by the configured setting. The rest of the input is no longer included in the log message, as shown in the following example:
>
> **Example:**
>
> In 10.5, when the Web App Firewall detects the SQL violation, the entire input string might be included in the log message, as shown below:
>
> ```
> SQL Keyword check failed for field text=\"select a name from testbed1
> ;(;)\".*<blocked>
> ```
>
> In the enhancement builds of 10.5.e that support request side streaming and 11.0 build onwards, we log only the field name, keyword, and special character (if applicable) in the log message, as shown below:
>
> ```
> SQL Keyword check failed for field **text="select(;)"<blocked>
> ```
>
> This change is applicable to requests that contain application/x-www-form-urlencoded, or multipart/form-data, or text/x-gwt-rpc content-types. Log messages generated during processing of **JSON** or **XML** payloads are not affected by this change.

To access the log messages by using the GUI

The Citrix GUI includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **HTML SQL Injection** row and click **Logs**. When you access the logs directly from the HTML SQL Injection check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **Citrix ADC** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **HTML SQL Injection** check, filter by selecting **APPFW** in the drop-down list options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_SQL** check box and click the **Apply** button, only log messages pertaining to the **SQL Injection** security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** and so forth. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click to Deploy functionality is available only in the GUI. You can use the Syslog Viewer to not only view the logs but also to deploy HTML SQL Injection relaxation rules based on the log messages for the Web App Firewall security check violations. The log messages must be in CEF log format for this operation. Click to deploy functionality is available only for log messages that are generated by the block (or not block) action. You cannot deploy a relaxation rule for a log message about the transform operation.

To deploy a relaxation rule from the Syslog Viewer, select the log message. A check box appears in the upper right corner of the **Syslog Viewer** box of the selected row. Select the check box, and then select an option from the Action list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options.

The SQL Injection rules that are deployed by using the Click to Deploy option do not include the fine grain relaxation recommendations.

To use Click to Deploy functionality in the GUI:

1. In the Syslog Viewer, select **Application Firewall** in the **Module** options.
2. Select the **APP_SQL** as the **Event Type** to filter corresponding log messages.
3. Select the check box to identify the rule to deploy.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Statistics for the SQL Injection violations

When the stats action is enabled, the counter for the SQL Injection check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 SQL Injection violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if the block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display SQL Injection check statistics by using the command line:

At the command prompt, type:

```
sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> stat appfw profile <profile name>
```

To display HTML SQL Injection statistics by using the GUI

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about HTML SQL Injection violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the SQL Injection check:

- **Built-in Support for SQL Injection Protection**—The Citrix Web App Firewall protects against SQL Injection by monitoring a combination of SQL keywords and special characters in the form parameters. All SQL keywords, special characters, wildcard characters, and default transformation rules are specified in the /netscaler/default_custom_settings.xml file.
- **Customization**—You can change the default key words, special characters, wildcard characters, and transformation rules to customize the SQL security check inspection for the specific needs of your application. Make a copy of the default signature object, modify existing entries, or add new ones. Bind this signature object to your profile to make use of the customized configuration.
- **Hybrid Security Model**—Both signatures and deep security protections use the SQL/cross-site scripting patterns specified in the signature object that is bound to the profile. If no signature object is bound to the profile, the SQL/cross-site scripting patterns present in the default signature object are used.
- **Transform**—Note the following about the transform operation:
  - The transform operation works independently of the other SQL Injection action settings. If transform is enabled and the block, log, stats, and learn are all disabled, SQL special characters are transformed.
  - When SQL Transformation is enabled, user requests are sent to the back end servers after the SQL special characters are transformed in non-block mode. If the block action is enabled, it takes precedence over the transform action. If the injection type is specified as SQL special character and the block is enabled, the request is blocked despite the transform action.

- **Fine Grained Relaxation and Learning**—Fine-tune the relaxation rule to relax a subset of SQL elements from security check inspection but detect the rest. The learning engine recommends a specific value type and value expressions based on the observed data.
- **Click to Deploy**—Select one, or multiple SQL violation log messages in the syslog viewer and deploy them as relaxation rules.

SQL grammar-based protection for HTML and JSON payload

September 14, 2021

Citrix Web App Firewall uses a pattern match approach for detecting SQL injection attacks in HTTP and JSON payloads. The approach uses a set of pre-defined key-words and (or) special characters to detect an attack and flag it as a violation. Although this approach is effective, it can result in many false positives resulting in adding one or more relaxation rules. Especially when commonly used words such as “Select” and “From” are used in an HTTP or JSON request. We can reduce false positives by implementing the SQL grammar protection check for HTML and JSON payload.

In the existing pattern match approach, an SQL injection attack is identified if a pre-defined keyword and or a special character is present in an HTTP request. In this case, the statement need not be a valid SQL statement. But in the grammar-based approach, an SQL injection attack is detected only if a keyword or a special character is present in a SQL statement or is part of a SQL statement thereby reducing false positive scenarios.

SQL grammar-based protection usage scenario

Consider a statement, “Select my tickets and let’s meet at union station” present in an HTTP request. Although, the statement is not a valid SQL statement, the existing pattern match approach detects the request as an SQL injection attack because the statement uses keywords such as “Select”, “and” and “Union”. But, in the case of the SQL grammar approach, the statement is not detected as a violation attack because the keywords are not present in a valid SQL statement or not part of a valid SQL statement.

The grammar-based approach can also be configured for detecting SQL injection attacks in JSON payloads. For adding a relaxation rule, you can reuse the existing relaxation rules. Fine grained relaxation rules are also applicable for SQL grammar, for rules with “valueType” “keyword”. In JSON SQL grammar, the existing URL-based method can be reused.

Configure SQL grammar-based protection by using the CLI

To implement SQL grammar based detection, you must configure the “SQLInjectionGrammar” parameter in the Web App Firewall profile. By default, the parameter is disabled. All existing SQL Injection actions are supported except learning. Any new profile created after an upgrade supports SQL injection grammar and it continues to have default type as “special character or keyword” and it must be explicitly enabled.

At the command prompt, type:

```
add appfw profile <profile-name> -SQLInjectionAction <action-name> -
    SQLInjectionGrammar ON/OFF
<!--NeedCopy-->
```

Example:

```
add appfw profile profile1 -SQLInjectionAction Block -SQLInjectionGrammar ON
```

Configure SQL pattern-match protection and grammar-based protection by using the CLI

If you have enabled both grammar-based and pattern-match approaches, then the appliance performs grammar-based detection first and if there is SQL injection detection with the action type set to block, the request is blocked (without verifying detection using pattern-match).

At the command prompt, type:

```
add appfw profile <profile-name> -SQLInjectionAction <action-name> -
    SQLInjectionGrammar ON -SQLInjectionType <Any action other than 'None' : SQLSplCharANDKeyword/ SQLSplCharORKeyword/ SQLSplChar/
    SQLKeyword>
<!--NeedCopy-->
```

Example:

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar ON -
SQLInjectionType SQLSplChar
```

Configure SQL Injection check only with grammar-based protection by using the CLI

At the command prompt, type:

```
add appfw profile <profile-name> -SQLInjectionAction <action-name> -
    SQLInjectionGrammar ON -SQLInjectionType None
<!--NeedCopy-->
```


Example:

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar ON -
SQLInjectionType None
```

Bind relaxation rules for SQL grammar-based protection by using the CLI

If your application requires you to bypass the SQL injection check for a specific "ELEMENT" or "ATTRIBUTE" in the payload, you must configure a relaxation rule.

> **Note:**
>
> Relaxation rules with valueType "keyword" are evaluated only when the appliance performs detection using SQL grammar.

The SQL command Injection inspection relaxation rules have the following syntax. At the command prompt, type:

```
1 bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|
     NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor
     |SpecialString|Wildchar) [<valueExpression>][-isValueRegex (REGEX |
     NOTREGEX) ]]
2 <!--NeedCopy-->
```

Example:

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -isregex
regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
Keyword 'selec[a-z]+' -isvalueRegex regEX
```

Configure SQL grammar-based protection for JSON payload by using the CLI

To implement SQL grammar-based detection for the JSON payload, you must configure the "JSONSQLInjectionGrammar" parameter in the Web App Firewall profile. By default, the parameter is disabled. All existing SQL Injection actions are supported except learning. Any new profile created after an upgrade supports SQL injection grammar and it continues to have default type as "special character or keyword" and you must explicitly enable it.

At the command prompt, type:

```
1 add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
     action-name> -JSONSQLInjectionGrammar ON/OFF
2 <!--NeedCopy-->
```


Example:

```
add appfw profile profile1 -type JSON -JSONSQLInjectionAction Block -JSONSQLInjectionG
 ON
```

Configure SQL pattern match protection and grammar-based protection by using the CLI

If you have enabled both grammar-based and pattern-match checks, then the appliance performs grammar-based detection first and if there is SQL injection detection with the action type set to block, the request is blocked (without verifying detection using pattern-match).

> **Note:**
>
> Relaxation rules with valueType “keyword” are evaluated only when the appliance performs detection using SQL grammar.

At the command prompt, type:

```
1 add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
      action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType <Any
       action other than  'None' : SQLSplCharANDKeyword/
      SQLSplCharORKeyword/ SQLSplChar/ SQLKeyword>
2 <!--NeedCopy-->
```

Example:

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block - JSONSQLInjectionGramma
 ON -JSONSQLInjectionType SQLSplChar
```

Configure SQL grammar-based protection for JSON payload by using the CLI

At the command prompt, type:

```
1 add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
      action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType None
      `\
2 <!--NeedCopy-->
```

Example:

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block - JSONSQLInjectionGramma
 ON -JSONSQLInjectionType None
```


Bind url-based relaxation rules for JSON SQL grammar-based protection by using the CLI

If your application requires you to bypass the JSON command injection inspection for a specific "ELEMENT" or "ATTRIBUTE" in the payload, you can configure a relaxation rule.
The JSON command Injection inspection relaxation rules have the following syntax. At the command prompt, type:

```
bind appfw profile <profile name>  - JSONCMDURL <expression> -comment <
    string> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED ) -state (
    ENABLED | DISABLED )
<!--NeedCopy-->
```

Example:

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -isregex
regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
Keyword 'selec[a-z]+' -isvalueRegex regEX
```

Configure SQL grammar based protection by using the GUI

Complete the GUI procedure to configure grammar based HTML SQL Injection detection.

1. On the navigation pane, navigate to **Security > Profiles**.
2. In the **Profiles** page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Security Checks** under **Advanced Settings**.
4. In the **Security Checks** section, go to **HTML SQL Injection** settings.
5. Click the executable icon near the check box.
6. Click **Action Settings** to access the **HMTL SQL Injection Settings** page.



7. Select the **Check using SQL Grammar** check box.
8. Click **OK**.

Configure SQL grammar based protection for JSON payload by using the GUI

Complete the GUI procedure to configure grammar based JSON SQL Injection detection.

1. On the navigation pane, navigate to **Security > Profiles**.
2. In the **Profiles** page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Security Checks** under **Advanced Settings**.
4. In the **Security Checks** section, go to **JSON SQL Injection** settings.
5. Click the executable icon near the check box.
6. Click **Action Settings** to access the **JSON SQL Injection Settings** page.
7. Select the **Check using SQL Grammar** check box.
8. Click **OK**.



Relaxation and deny rules for handling HTML SQL injection attacks

September 14, 2021

When there is an incoming traffic, the violation detection logic checks for traffic violations. If no HTML SQL injection attacks are detected, the traffic is allowed to pass. But if a violation is detected, the relaxation (allow) and deny rules define how to handle the violations. If the security check is configured in the allow mode (default mode), the detected violation is blocked unless the user has explicitly configured a relaxation or allow rule.

In addition to allow mode, the security check can also be configured in deny mode and use deny rules for handling violations. If the security check is configured in this mode, the detected violations are blocked if a user has explicitly configured a deny rule. If there are no deny rules configured, then the user configured action is applied.

The following illustration explains how to allow and deny modes of operation work:



1. When a violation is detected, the relaxation (allow) and deny rules define how to handle the violations.
2. If the security check is configured in deny mode (if configured in allow mode, jump to step 5), the violation is blocked unless you have explicitly configured a deny rule.
3. If the violation matches a deny rule, the appliance blocks the traffic.
4. If the traffic violation does not match a rule, the appliance applies a user-defined action (block, reset, or drop).
5. If the security check is configured in allow mode, the Web App Firewall module checks if there are any allow rule configured.
6. If the violation matches an allow rule, the appliance allows the traffic to bypass otherwise, it is blocked.

Configure security check-in relaxation and enforcement mode

At the command prompt, type:

```
set appfw profile <name>   - SQLInjectionAction [block stats learn]  -
    SQLInjectionRuleType [ALLOW DENY]
<!--NeedCopy-->
```

Example:

```
set appfw profile prof1 sqlInjectionAction block -sqlInjectionRuleType
ALLOW DENY
```

Bind relaxation and enforcement rules to Web Application Firewall profile

At the command prompt, type:


```
bind appfw profile <name> -SQLInjection <string> <formActionURL>
<!--NeedCopy-->
```

Example:

```
bind appfw profile p1 -SQLInjection field_f1 "/login.php"-RuleType ALLOW
bind appfw profile p2 -SQLInjection field_f1 "/login.php"-RuleType ALLOW
```

HTML command injection protection check

September 14, 2021

The **HTML** command injection check examines if the incoming traffic has unauthorized commands that break the system security or modify the system. If the traffic has any malicious commands when detected, the appliance blocks the request or performs the configured action.

The Citrix Web App Firewall profile is now enhanced with a new security check for command injection attacks. When the command injection security check examines the traffic and detects any malicious commands, the appliance blocks the request or performs the configured action.

In a command injection attack, the attacker aims to run unauthorized commands on the Citrix ADC operation system. To achieve this, the attacker injects operating system commands using a vulnerable application. A Citrix ADC appliance is vulnerable to injection attacks if the application passes any unsafe data (forms, cookies, or header) to the system shell.

How command injection protection works

1. For an incoming request, WAF examines the traffic for keywords or special characters. If the incoming request has no patterns that match any of the denied keywords or special characters, the request is allowed. Otherwise, the request is blocked, dropped, or redirected based on the configured action.
2. If you prefer to exempt a keyword or a special character from the list, you can apply a relaxation rule to bypass the security check under specific conditions.
3. You can enable logging to generate log messages. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
4. You can also enable the statistics feature to gather statistical data about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If

legitimate requests are getting blocked, you might have to revisit the configuration to see if you must configure the new relaxation rule or modify the existing one.

Keywords and special characters denied for command injection check

To detect and block command injection attacks, the appliance has a set of patterns (keywords and special characters) defined in the default signature file. Following is a list of keywords blocked during command injection detection.

```
    <commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>
<!--NeedCopy-->
```

Special characters defined in the signature file are:
`| ; & $ > < '\ ! >> ##`

Configuring command injection check by using the CLI

In the command line interface, you can use either the set the profile command or the add the profile command to configure the command injection settings. You can enable the block, log, and stats actions. You must also set the key words and string characters that you want to detect in the payloads.

At the command prompt, type:

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -CMDInjectionType
 <CMDInjectionType>]
```

> **Note:**
>
> By default, the command injection action is set as “None.” Also, the default command injection type is set as `CmdSplCharANDKeyWord`.

Example:

```
set appfw profile profile1 -cmdInjectionAction block -CMDInjectionType
CmdSplChar
```

Where, the available command injection actions are:

- None - Disable command injection protection.
- Log - Log command injection violations for the security check.
- Block - blocks traffic that violates the command injection security check.

- Stats - Generates statistics for command injection security violations.

Where, the available command injection types are:

- Cmd SplChar. Checks special characters
- CmdKeyWord. Checks command injection Keywords
- CmdSplCharANDKeyWord. Checks special characters and command injection. Keywords and blocks only if both are present.
- CmdSplCharORKeyWord. Checks special characters and command injection Keywords and blocks if either of them is found.

Configuring relaxation rules for command injection protection check

If your application requires you to bypass the command injection inspection for a specific ELEMENT or ATTRIBUTE in the payload, you can configure a relaxation rule.

The command Injection inspection relaxation rules have the following syntax:

```
bind appfw profile <profile name> -cmdInjection <string> <URL> -isregex <
REGEX/NOTREGEX>
```

Example for relaxation rule for Regex in header

```
bind appfw profile sample -CMDInjection hdr "http://10.10.10.10/"-location
heaDER -valueType Keyword '[a-z]+grep'-isvalueRegex REGEX
```

As a result, the injection exempts the command injection check allows header `hdr` containing variants of “grep.”

Example for relaxation rule with valueType as regex in cookie

```
bind appfw profile sample -CMDInjection ck_login "http://10.10.10.10/"-
location cookie -valueType Keyword 'pkg[a-z]+'-isvalueRegex REGEX
```

Configuring command injection check by using Citrix ADC GUI

Complete the following steps to configure the command injection check.

1. Navigate to **Security > Citrix Web App Firewall and Profiles**.
2. On the **Profiles** page, select a profile and click **Edit**.
3. On the **Citrix Web App Firewall Profile** page, go to the **Advanced Settings** section and click **Security Checks**.



1. In the **Security Checks** section, select **HTML Command Injection** and click **Action** settings.
2. In the **HTML Command Injection Settings** page, set the following parameters:
   a) Actions. Select one or more actions to perform for command injection security check.
   b) Check Request Containing. Select a command injection pattern to check if the incoming request has the pattern.
3. Click **OK**.



View or customize command injection patterns by using the GUI

You can use the GUI to view or customize the **HTML** command injection patterns.

The default command injection patterns are specified in default signatures file. If you do not bind any signature object to your profile, the default HTML command injection patterns specified in the default signatures object will be used by the profile for the command injection security check processing. The rules and patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these patterns, make a copy of the default sSignatures object to create a User-Defined signature object. Make changes in the command injection patterns in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized patterns.

For more information, see Signatures

To view the default command injection patterns by using the GUI:

1. Navigate to **Application Firewall > Signatures**, select ***Default Signatures**, and click **Edit**.

1. Click **Manage CMD/SQL/XSS patterns**. The **CMD/SQL/XSS Paths (read-only)** table shows patterns pertaining to `CMD`/`SQL`/`XSS` injection:



1. Select a row and click **Manage Elements** to display the corresponding command injection patterns (keywords, special strings, transformation rules, or wildcard characters) used by the Web App Firewall command injection check.

To customize a command injection pattern by using the GUI

You can edit the user-defined signature object to customize the **CMD** key words, special strings, and wildcard characters. You can add new entries or remove the existing ones. You can modify the transformation rules for the command injection special strings.

1. **Navigate to Application Firewall > Signatures**, highlight the target User-defined signature, and click **Add**. Click **Manage CMD/SQL/XSS patterns**.
2. In the **Manage CMD/SQL/XSS paths** page, select the target CMD injection row.
3. Click **Manage Elements**, **Add**, or **Remove** a command injection element.

> **Warning:**
>
> You must be careful before you remove or modify any default command injection element, or delete the CMD path to remove the entire row. The signature rules and the command injection security check rely on these elements for detecting command injection attacks to protect your applications. Customizing the SQL patterns can make your application vulnerable to command injection attacks if the required pattern is removed during editing.



Viewing command injection traffic and violation statistics

The **Citrix Web App Firewall Statistics** page shows security traffic and security violation details in a tabular or graphical format.

To view security statistics by using the command interface.

At the command prompt, type:

```
stat appfw profile profile1
```

Appfw profile Traffic Statistics	Rate (/s)	Total
Requests	0	0
Request Bytes	0	0
Responses	0	0
Response Bytes	0	0
Aborts	0	0
Redirects	0	0
Long Term Ave Response Time (ms)	–	0
Recent Ave Response Time (ms)	–	0

HTML/XML/JSON Violation Statistics	Rate (/s)	Total
Start URL	0	0
Deny URL	0	0
Referer header	0	0
Buffer overflow	0	0
Cookie consistency	0	0
Cookie hijacking	0	0
CSRF form tag	0	0
HTML Cross-site scripting	0	0
HTML SQL injection	0	0
Field format	0	0
Field consistency	0	0
Credit card	0	0
Safe object	0	0
Signature Violations	0	0
Content Type	0	0
JSON Denial of Service	0	0
JSON SQL injection	0	0
JSON Cross-Site Scripting	0	0
File Upload Types	0	0
Infer Content Type XML Payload	0	0
HTML CMD Injection	0	0
XML Format	0	0
XML Denial of Service (XDoS)	0	0
XML Message Validation	0	0
Web Services Interoperability	0	0
XML SQL Injection	0	0
XML Cross-Site Scripting	0	0
XML Attachment	0	0

HTML/XML/JSON Violation Statistics	Rate (/s)	Total
SOAP Fault Violations	0	0
XML Generic Violations	0	0
Total Violations	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
Start URL logs	0	0
Deny URL logs	0	0
Referer header logs	0	0
Buffer overflow logs	0	0
Cookie consistency logs	0	0
Cookie hijacking logs	0	0
CSRF from tag logs	0	0
HTML cross-site scripting logs	0	0
HTML cross-site scripting transform logs	0	0
HTML SQL Injection logs	0	0
HTML SQL transform logs	0	0
Field format logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
File upload types logs	0	0
Infer Content Type XML Payload L	0	0
HTML Command Injection logs	0	0
XML Format logs	0	0
XML Denial of Service(XDoS) logs	0	0
XML Message Validation logs	0	0
WSI logs	0	0
XML SQL Injection logs	0	0
XML cross-site scripting logs	0	0
XML Attachment logs	0	0
SOAP Fault logs	0	0
XML Generic logs	0	0
Total log messages	0	0

Server Error Response Statistics Rate (/s) > Total		
HTTP Client Errors (4xx Resp)	0	0
HTTP Server Errors (5xx Resp)	0	0

Viewing HTML command injection statistics by using the Citrix ADC GUI

Complete the following steps to view the command injection statistics:

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select a Web App Firewall profile and click **Statistics**.
3. The **Citrix Web App Firewall Statistics** page displays the HTML command injection traffic and violation details.
4. You can select **Tabular View** or switch to **Graphical View** to display the data in a tabular or graphical format.

HTML command injection traffic statistics

HTML SQL Injection logs	0	0
HTML SQL transform logs	0	0
Field format logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload Logs	0	0
HTML Command Injection logs	0	0
XML Format logs	0	0
XML Denial of Service(XDoS) logs	0	0
XML Message Validation logs	0	0
WSI logs	0	0
XML SQL Injection logs	0	0
XML XSS logs	0	0
XML Attachment logs	0	0

HTML command injection violation statistics

HTML/XML/JSON Violation Statistics

	Rate (/s)	Total	
Start URL	0	0	0%
Deny URL	0	0	0%
Referer header	0	0	0%
Buffer overflow	0	0	0%
Cookie consistency	0	0	0%
Cookie hijacking	0	0	0%
CSRF form tag	0	0	0%
HTML Cross-site scripting	0	0	0%
HTML SQL injection	0	0	0%
Field format	0	0	0%
Field consistency	0	0	0%
Credit card	0	0	0%
Safe object	0	0	0%
Signature logs	0	0	0%
Content Type	0	0	0%
JSON Denial of Service	0	0	0%
JSON SQL injection	0	0	0%
JSON Cross-Site Scripting	0	0	0%
File Upload Types	0	0	0%
Infer Content Type XML Payload	0	0	0%
HTML CMD Injection	0	0	0%
XML Format	0	0	0%
XML Denial of Service (XDoS)	0	0	
XML Message Validation	0	0	
Web Services Interoperability	0	0	

JSON command injection protection check

September 14, 2021

The JSON command injection check examines the incoming JSON traffic for unauthorized commands that break the system security or modify the system. When examining the traffic, if any malicious commands are detected, the appliance blocks the request or performs the configured action.

In a command injection attack, the attacker aims to run unauthorized commands on the Citrix ADC operating system or the back-end server. To achieve this, the attacker injects operating system commands using a vulnerable application. The back-end application is vulnerable to injection attacks if the appliance simply forwards a request without any security check. Therefore, it is highly important to configure a security check, so the Citrix ADC appliance can protect your web application by blocking unsafe data.

How command injection protection works

1. For an incoming JSON request, WAF examines the traffic for keywords or special characters. If the JSON request has no patterns that match any of the denied keywords or special characters, the request is allowed. Otherwise, the request is blocked, dropped, or redirected based on the configured action.
2. If you prefer to exempt a keyword or a special character from the list, you can create a relaxation rule to bypass the security check under specific conditions.
3. You can enable logging to generate log messages. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
4. You can also enable the statistics feature to gather statistical data about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you must configure the new relaxation rule or modify the existing one.

Keywords and special characters denied for command injection check

To detect and block JSON command injection attacks, the appliance has a set of patterns (keywords and special characters) defined in the default signature file. Following is a list of keywords blocked during command injection detection.

```
<commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
```



```
5  ...
6  </commandinjection>
7
8  <!--NeedCopy-->
```

Special characters defined in the signature file are:
| ; & $ > < '\ ! >> ##

Configuring JSON command injection check by using the CLI

In the command line interface, you can use either the set appfw profile command or add an appfw profile command to configure the JSON command injection settings. You can enable the block, log, and stats actions. You must also set the command injection type such as key words and string characters that you want to detect in the payloads.

At the command prompt, type:

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -CMDInjectionType
 <CMDInjectionType>]
```

> **Note:**
>
> By default, the command injection action is set as “block log stats”. Also, the default command injection type is set as `CmdSplCharANDKeyWord`. After an upgrade, the existing Web app Firewall profiles have the action set as “None.”.

Example:

```
set appfw profile profile1 -JSONCMDInjectionAction block -JSONCMDInjectionType
 CmdSplChar
```

Where, the available JSON command injection actions are:

None - Disable command injection protection.
Log - Log command injection violations for the security check.
Block - blocks traffic that violates the command injection security check.
Stats - Generates statistics for command injection security violations.

Where, the available JSON command injection types are:

`Cmd SplChar` - Checks special characters
`CmdKeyWord` - Checks command injection Keywords
`CmdSplCharANDKeyWord` - This is the default action. The action checks special characters and command injection. Keywords and blocks only if both are present.
`CmdSplCharORKeyWord` - Checks special characters and command injection Keywords and blocks if either of them is found.

Configuring relaxation rules for JSON command injection protection check

If your application requires you to bypass the JSON command injection inspection for a specific ELEMENT or ATTRIBUTE in the payload, you can configure a relaxation rule.

The JSON command Injection inspection relaxation rules have the following syntax.

```
bind appfw profile <profile name> -JSONCMDURL <expression> -comment <string
> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED )-state ( ENABLED |
DISABLED )
```

Example for relaxation rule for Regex in header

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/hello.html
```

Whereas, the following relaxes requests from all URLs hosted on 1.1.1.1:

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/*"
```

To remove the relaxation, use 'unbind'.

```
unbind appfw profile abc_json -jsoncmDURL  " http://1.1.1.1/*"
```

Configure JSON command injection check by using the GUI

Complete the following steps to configure the JSON command injection check.

1. Navigate to **Security > Citrix Web App Firewall and Profiles**.
2. On the **Profiles** page, select a profile and click **Edit**.
3. On the **Citrix Web App Firewall Profile** page, go to **Advanced Settings** section and click **Security Checks**.



1. In **Security Checks** section, select **JSON Command Injection** and click **Action** settings.
2. In the **JSON Command Injection Settings** page, set the following parameters
   a) Actions. Select one or more actions to perform for JSON command injection security check.
   b) Check Request Containing. Select a command injection pattern to check if the incoming request has the pattern.
3. Click **OK**.



Viewing command injection traffic and violation statistics

The **Citrix Web App Firewall Statistics** page shows security traffic and security violation details in a tabular or graphical format.

To view security statistics by using the command interface.

At the command prompt, type:

```
stat appfw profile profile1
```

Appfw profile Traffic Statistics	Rate (/s)	Total
Requests	0	0
Request Bytes	0	0
Responses	0	0
Response Bytes	0	0
Aborts	0	0
Redirects	0	0
Long Term Ave Response Time (ms)	–	0
Recent Ave Response Time (ms)	–	0

HTML/XML/JSON Violation Statistics	Rate (/s)	Total
Start URL	0	0
Deny URL	0	0
Referer header	0	0
Buffer overflow	0	0
Cookie consistency	0	0
Cookie hijacking	0	0
CSRF form tag	0	0
HTML Cross-site scripting	0	0
HTML SQL injection	0	0
Field format	0	0

HTML/XML/JSON Violation Statistics	Rate (/s)	Total
Field consistency	0	0
Credit card	0	0
Safe object	0	0
Signature Violations	0	0
Content Type	0	0
JSON Denial of Service	0	0
JSON SQL injection	0	0
JSON Cross-Site Scripting	0	0
File Upload Types	0	0
Infer Content Type XML Payload	0	0
HTML CMD Injection	0	0
XML Format	0	0
XML Denial of Service (XDoS)	0	0
XML Message Validation	0	0
Web Services Interoperability	0	0
XML SQL Injection	0	0
XML Cross-Site Scripting	0	0
XML Attachment	0	0
SOAP Fault Violations	0	0
XML Generic Violations	0	0
Total Violations	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
Start URL logs	0	0
Deny URL logs	0	0
Referer header logs	0	0
Buffer overflow logs	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
Cookie consistency logs	0	0
Cookie hijacking logs	0	0
CSRF from tag logs	0	0
HTML cross-site scripting logs	0	0
HTML cross-site scripting transform logs	0	0
HTML SQL Injection logs	0	0
HTML SQL transform logs	0	0
Field format logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload L	0	0
JSON CMD Injection	0	0
HTML Command Injection logs	0	0
XML Format logs	0	0
XML Denial of Service(XDoS) logs	0	0
XML Message Validation logs	0	0
WSI logs	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
XML SQL Injection logs	0	0
XML cross-site scripting logs	0	0
XML Attachment logs	0	0
SOAP Fault logs	0	0
XML Generic logs	0	0
Total log messages	0	0

Server Error Response Statistics Rate (/s)	Total	
HTTP Client Errors (4xx Resp)	0	0
HTTP Server Errors (5xx Resp)	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
JSON Command Injection logs	0	0
XML format logs	0	0

Viewing JSON command injection statistics by using the Citrix ADC GUI

Complete the following steps to view the command injection statistics:

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select a Web App Firewall profile and click **Statistics**.
3. The **Citrix Web App Firewall Statistics** page displays the JSON command injection traffic and violation details.
4. You can select **Tabular View** or switch to **Graphical View** to display the data in a tabular or graphical format.

JSON command injection traffic statistics

HTML/XML/JSON Log Statistics

	Rate (/s)	Total
Start URL logs	0	0
Deny URL logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0
JSON CMD injection logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload Logs	0	0

JSON CMD injection logs: X
Number of JSON Command Injection security check log messages generated by the Application Firewall.

JSON command injection violation statistics



XML external entities (XXE) Attack Protection

September 14, 2021

The XML external entities (XXE) attack protection examines if an incoming payload has any unauthorized XML input regarding entities outside the trusted domain where the web application resides. The XXE attack occurs if you have a weak XML parser that parses an XML payload with input containing references to external entities.

In a Citrix ADC appliance, if the XML parser is improperly configured, the impact of exploiting the vulnerability can be dangerous. It allows an attacker to read sensitive data on the web server. Perform the denial of service attack and so forth. Therefore, it is important protect the appliance from XXE

attacks. Web Application Firewall is able to protect the appliance from XXE attacks as long as the content-type is identified as XML. To prevent a malicious user from bypassing this protection mechanism, WAF blocks an incoming request if the "inferred" content-type in the HTTP headers does not match with the content-type of the body. This mechanism prevents the XXE attack protection bypass when a whitelisted default or non-default content-type is used.

Some of the potential XXE threats that affect a Citrix ADC appliance are:

- Confidential data leaks
- Denial-of-service (DOS) attacks
- server side forgery requests
- Port scanning

Configure XML external entities (XXE) injection protection

To configure XML external entities (XXE) check by using the command interface:
In the command line interface, you can add or modify the application firewall profile command to configure the **XXE** settings. You can enable the block, log, and stats actions.

At the command prompt, type:

```
set appfw profile <name> [-inferContentTypeXmlPayloadAction <inferContentTypeXmlPayloa
 <block | log | stats | none>]
```

> Note:
>
> By default, the XXE action is set as "none."

Example:

```
set appfw profile profile1 -inferContentTypeXmlPayloadAction Block
```

Where, action types are:

Block: The request is blocked without any exception to the urls in the request.

Log: If a mismatch between content-type in an HTTP request header and payload occurs, information about the violating request must be contained in the log message.

Stats: If a mismatch in the content-types is detected, the corresponding statistics for this violation type is incremented.

None: No action is taken if mismatch in content-types is detected. None cannot be combined with any other action type. Default action is set to None.

Configure XXE injection check by using Citrix ADC GUI

Complete the following steps to configure the XXE injection check.

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. On the **Profiles** page, select a profile and click **Edit**.
3. On the **Citrix Web App Firewall Profile** page, go to the **Advanced Settings** section and click **Security Checks**.



4. In the **Security Checks** section, select **Infer Content Type XML Payload** and click **Action** settings.
5. In the Infer Content Type XML Payload Settings page, set the following parameters:
   a) Actions. Select one or more actions to perform for XXE injection security check.
6. Click **OK**.



Viewing XXE injection traffic and violation statistics

The Citrix Web App Firewall Statistics page shows security traffic and security violation details in a tabular or graphical format.

To view security statistics by using the command interface.

At the command prompt, type:

```
stat appfw profile profile1
```


Viewing XXE injection statistics by using the Citrix ADC GUI

Complete the following steps to view the XXE injection statistics:

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select a Web App Firewall profile and click **Statistics**.
3. The **Citrix Web App Firewall Statistics** page displays the XXE command injection traffic and violation details.
4. You can select **Tabular View** or switch to **Graphical View** to display the data in a tabular or graphical format.



HTML/XML/JSON Violation Statistics

	Rate (/s)	Total	
Start URL	0	0	0%
Deny URL	0	0	0%
Referer header	0	0	0%
Buffer overflow	0	0	0%
Cookie consistency	0	0	0%
Cookie hijacking	0	0	0%
CSRF form tag	0	0	0%
HTML Cross-site scripting	0	0	0%
HTML SQL injection	0	0	0%
Field format	0	0	0%
Field consistency	0	0	0%
Credit card	0	0	0%
Safe object	0	0	0%
Signature logs	0	0	0%
Content Type	0	0	0%
JSON Denial of Service	0	0	0%
JSON SQL injection	0	0	0%
JSON Cross-Site Scripting	0	0	0%
File Upload Types	0	0	0%
Infer Content Type XML Payload	0	0	0%
HTML CMD Injection	0	0	0%

Buffer overflow check

September 14, 2021

The Buffer Overflow check detects attempts to cause a buffer overflow on the web server. If the Web App Firewall detects that the URL, cookies, or header are longer than the configured length, it blocks the request because it can cause a buffer overflow.

The Buffer Overflow check prevents attacks against insecure operating-system or web-server software

that can crash or behave unpredictably when it receives a data string that is larger than it can handle. Proper programming techniques prevent buffer overflows by checking incoming data and either rejecting or truncating overlong strings. Many programs, however, do not check all incoming data and are therefore vulnerable to buffer overflows. This issue especially affects older versions of web-server software and operating systems, many of which are still in use.

The Buffer Overflow security check allows you to configure the **Block**, **Log**, and **Stats** actions. In addition, you can also configure the following parameters:

- **Maximum URL Length**. The maximum length the Web App Firewall allows in a requested URL. Requests with longer URLs are blocked. **Possible Values**: 0–65535. **Default**: 1024
- **Maximum Cookie Length**. The maximum length the Web App Firewall allows for all cookies in a request. Requests with longer cookies trigger the violations. **Possible Values**: 0–65535. **Default**: 4096
- **Maximum Header Length**. The maximum length the Web App Firewall allows for HTTP headers. Requests with longer headers are blocked. **Possible Values**: 0–65535. **Default**: 4096
- **Query string length**. Maximum length allowed for query string in an incoming request. Requests with longer queries are blocked. Possible Values: 0–65535. Default: 1024
- **Total request length**. Maximum request length allowed for an incoming request. Requests with longer length are blocked. Possible Values: 0–65535. Default: 24820

Using the command line to configure the Buffer Overflow security check

To configure Buffer Overflow security check actions and other parameters by using the command line

At the command prompt, type:

```
add appfw profile <name> -bufferOverflowMaxURLLength <positive_integer> -
bufferOverflowMaxHeaderLength <positive_integer> - bufferOverflowMaxCookieLength
 <positive_integer> -bufferOverflowMaxQueryLength <positive_integer> -
bufferOverflowMaxTotalHeaderLength <positive_integer>
```

Example:

```
add appfw profile profile1 -bufferOverflowMaxURLLength 7000 -bufferOverflowMaxHeaderLe
 7250 - bufferOverflowMaxCookieLength 7100 -bufferOverflowMaxQueryLength
7300 -bufferOverflowMaxTotalHeaderLength 7300
```

Configure buffer overflow security check by using the Citrix ADC GUI

1. Navigate to **Security** > **Web App Firewall** and **Profiles**.
2. On the **Profiles** page, select a profile and click **Edit**.

3. On the **Citrix Web App Firewall Profile** page, go to **Advanced Settings** section and click **Security Checks**.
4. In **Security Checks** section, select **Buffer Overflow** and click **Action Settings**.
5. In the **Buffer Overflow Settings** page, set the following parameters.
   a. Actions. Select one or more actions to perform for command injection security check.
   b. Maximum URL Length. Maximum length, in characters, for URLs on your protected websites. Requests with longer URLs are blocked.
   c. Maximum Cookie Length. Maximum length, in characters, for cookies sent to your protected websites. Requests with longer cookies are blocked.
   d. Maximum Header Length. Maximum length, in characters, for HTTP headers in requests sent to your protected websites. Requests with longer headers are blocked.
   e. Maximum Query Length. Maximum length, in bytes, for query string sent to your protected websites. Requests with longer query strings are blocked.
   f. Maximum Total Header Length. Maximum length, in bytes, for the total HTTP header length in requests sent to your protected websites. The minimum value of this and maxHeaderLen in httpProfile will be used. Requests with longer length are blocked.
6. Click **OK** and **Close**.



Using the Log Feature with the Buffer Overflow Security Check

When the log action is enabled, the Buffer Overflow security check violations are logged in the audit log as **APPFW_BUFFEROVERFLOW_URL**, **APPFW_BUFFEROVERFLOW_COOKIE**, and **APPFW_BUFFEROVERFLOW_HDR** violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

If you use the GUI to review the logs, you can use the click-to-deploy feature to apply relaxations indicated by the logs.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the **/var/log/** folder to access the log messages pertaining to the Buffer overflow violations:

```
> **Shell**
> **tail -f /var/log/ns.log | grep APPFW_BUFFEROVERFLOW**
<!--NeedCopy-->
```

Example of a CEF log message showing bufferOverflowMaxCookieLength violation in non-block mode

```
Oct 22 17:35:20 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|**APPFW_BUFFEROVERFLOW_COOKIE**|6|src=10.217.253.62
    geolocation=Unknown spt=41198 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html **msg=Cookie header length(43) is greater
     than maximum allowed(16).** cn1=119 cn2=465 cs1=owa_profile cs2=
    PPE1 cs3=wvOOOb+cJ2ZRbstZpyeNXIqLj7Y0001 cs4=ALERT cs5=2015 **act=
    not blocked**
<!--NeedCopy-->
```

Example of a CEF log message showing bufferOverflowMaxURLLength violation in non-block mode

```
Oct 22 18:39:56 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|**APPFW_BUFFEROVERFLOW_URL**|6|src=10.217.253.62
    geolocation=Unknown spt=19171 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html **msg=URL length(39) is greater than
    maximum allowed(20).** cn1=707 cn2=402 cs1=owa_profile cs2=PPE0 cs3=
    kW49GcKbnwKByByi3+jeNzfgWa80000 cs4=ALERT cs5=2015 **act=not blocked
    **
<!--NeedCopy-->
```

Example of a Native Format Log message showing bufferOverflowMaxHeaderLength violation in block mode

```
Oct 22 18:44:00 <local0.info> 10.217.31.98 10/22/2015:18:44:00 GMT ns
    0-PPE-2 : default APPFW **APPFW_BUFFEROVERFLOW_HDR** 155 0 :
```



```
      10.217.253.62 374-PPE2 khhBEeY4DB8V2D3H2sMLkXmfWnA0002 owa_profile
      **Header(User-Agent) length(82) is greater than maximum allowed(10)
      ** : http://aaron.stratum8.net/ **<blocked>**
<!--NeedCopy-->
```

To access the log messages by using the GUI

The Citrix GUI includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **Buffer Overflow** row and click **Logs**. When you access the logs directly from the Buffer Overflow Security Check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **Buffer Overflow** check, filter by selecting **APPFW** in the drop-down list options for **Module**. The **Event Type** list offers three options, **APPFW_BUFFEROVERFLOW_URL**, **APPFW_BUFFEROVERFLOW_COOKIE**, and A**PPFW_BUFFEROVERFLOW_HDR**, to view all the log messages pertaining to buffer overflow security check. You can select one or more options to further refine your selection. For example, if you select the **APPFW_BUFFEROVERFLOW_COOKIE** check box and click the **Apply** button, only log messages pertaining to the **Buffer Overflow security** check violations for the Cookie header appear in the Syslog Viewer. If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, and **Client IP**, appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click-to-Deploy: The GUI provides click-to-deploy functionality, which is currently supported only for the buffer overflow log messages pertaining to the **URL Length** violations. You can use the Syslog Viewer to not only view the triggered violations, but also run informed decisions based on the observed lengths of the blocked messages. If the current value is too restrictive and is triggering false positives, you can select a message and deploy it to replace the current value with the URL length value seen in the message. The log messages must be in CEF log format for this operation. If the relaxation can be deployed for a log message, a check box appears at the right edge of the **Syslog Viewer** box in the row. Select the check box, and then select an option from the **Action** list to deploy the

relaxation. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options. You can use the **APPFW_BUFFEROVERFLOW_URL** filter to isolate all the log messages pertaining to the configured URL length violations.

If you select an individual log message, all three action options **Edit & Deploy**, **Deploy**, and **Deploy All** are available. If you select **Edit & Deploy**, the **Buffer Overflow settings** dialogue is displayed. The new URL length that was observed in the request is inserted into the **Maximum URL length input** field. If you click **Close** without any edits, the current configured values remain unchanged. If you click the **OK** button, the new value of the Maximum URL length replaces the previous value.

> **Note**
>
> The **block**, **log** and **stats** action check boxes are unchecked in the displayed **Buffer Overflow settings** dialogue, and need to be reconfigured if you select the **Edit & Deploy** option. Make sure to enable these check boxes before clicking **OK**, otherwise the new URL length gets configured but the actions are set to **none**.

If you select the check boxes for multiple log messages, you can use the **Deploy** or **Deploy All** option. If the deployed log messages have different URL lengths, the configured value gets replaced by the highest URL Length value observed in the selected messages. Deploying the rule results only in changing the **bufferOverflowMaxURLLength** value. Configured actions are retained and remain unchanged.

To use Click-to-Deploy functionality in the GUI

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Enable the **APPFW_BUFFEROVERFLOW_URL** check box as the **Event Type** to filter corresponding log messages.
3. Enable the check box to select the rule.
4. Use the **Action** drop-down list of options to deploy the relaxation.
5. Navigate to **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks** to access the **Buffer Overflow** settings pane to verify that the **Maximum URL Length** value is updated.

Statistics for the Buffer Overflow violations

When the stats action is enabled, the counter for the Buffer Overflow Security Check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three Buffer Overflow violations increments the stats counter by one, because the page is blocked when the first violation is detected. However, if block is disabled, processing the same request increments the stat counter for violations because each violation generates a separate log message.

To display Buffer Overflow Security Check statistics by using the command line

At the command prompt, type:

```
> sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> stat appfw profile <profile name>
```

To display Buffer Overflow statistics by using the GUI

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about Buffer Overflow violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

- The buffer overflow security check allows you to configure limits to enforce the maximum length of allowed URLs, Cookies, and Headers.
- **Block**, **Log** and **Stats** actions enable you to monitor the traffic and configure optimal protection for your application.
- Syslog viewer enables you to filter and view all the log messages pertaining to buffer overflow violations.
- **Click-to-Deploy** functionality is supported for the **bufferOverflowMaxURLLength** violations. You can select and deploy an individual rule, or you can select multiple log messages to tweak and relax the current configured value of the maximum allowed length of the URL. The highest value of the URL from the selected group is set as the new value, to allow all these requests that are currently flagged as violations.
- The Web App Firewall now evaluates individual cookies when inspecting the incoming request. If length of any one cookie received in the Cookie header exceeds the configured **BufferOverflowMaxCookieLength**, the Buffer Overflow violation is triggered.

> **Important**
>
> In release 10.5.e (in a few interim enhancements builds prior to 59.13xx.e build) and in the 11.0 release (in builds prior to 65.x), Web App Firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. ** In some situations, the total cookie size forwarded

> to the server might be larger than the accepted value, and the server might respond with “400 Bad Request”.
>
> This change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds in addition to 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.

Web App Firewall support for Google web toolkit

September 14, 2021

Note: This feature is available in Citrix ADC release 10.5.e.

Web servers following Google Web Toolkit (GWT) Remote Procedure Call (RPC) mechanisms can be secured by the Citrix Web App Firewall without a need for any specific configuration to enable the GWT support.

What is GWT

The GWT is used for building and optimizing complex high-performance web applications by people who do not have expertise in XMLHttpRequest, and JavaScript. This open source, free development toolkit is used extensively for developing small and large scale applications and is quite frequently used for displaying browser based data such as search results for flights, hotels, and so on. The GWT provides a core set of Java APIs and widgets for writing optimized JavaScript scripts that can run on most browsers and mobile devices. The GWT RPC framework makes it easy for the client and server components of the web application to exchange Java objects over HTTP. GWT RPC services are not the same as web services based on SOAP or REST. They are simply a lightweight method for transferring data between the server and the GWT application on the client. GWT handles serialization of the Java objects exchanging the arguments in the method calls and the return value.

For popular websites that use GWT, see

https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29

How a GWT request works

The GWT RPC request is pipe delimited and has variable number of arguments. It is carried as a payload of HTTP POST and has the following values:

1. Content-type = text/x-gwt-rpc. Charset can be any value.

2. Method = POST.

Both GET and POST HTTP requests are considered valid GWT requests if content-type is “text/x-gwt-rpc”. Query strings are now supported as part of GWT requests. Configure the “InspectQueryContent-Types” parameter of the App Firewall profile to “OTHER” to examine the request query portion for conent-type “text/x-gwt-rpc”.

The following example shows a valid payload for a GWT request:

```
5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com
    .test.client.TestService|testMethod|java.lang.String|java.lang.
    Integer| myInput1|java.lang.Integer/3438268394|1|2|3|4|2|5|6|7|8|1|
<!--NeedCopy-->
```

The request can be divided into three parts:

a) Header: 5|0|8|

The first 3 digits `5|0|8|` in the above request, represent “version, subversion, and size of table”, respectively. These must be positive integers.

b) String Table:

`http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.TestService|testMethod|java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394|`

The members of the above pipe delimited string table contain the user-provided inputs. These inputs are parsed for the Web App Firewall checks and are identified as follows:

- 1st : `http://localhost:8080/test/`

  This is the Request URL.

- 2nd : `16878339F02B83818D264AE430C20468`

  Unique HEX identifier. A request is considered malformed if this string has non-hex characters.

- 3rd : `com.test.client.TestService`

  Service Class name

- 4th : `testMethod`

  Service method name

- 5th onwards: `java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394`

  Data-types and data. Non-primitive data-types are specified as

  `<container>.<sub-cntnr>.name/<integer><identifier>`

c) `Payload: 1|2|3|4|2|5|6|7|8|1|`

The payload consists of references to the elements in the string table. These integer values cannot be larger than the number of elements in the string table.

Web App Firewall protection for GWT applications

The Web App Firewall understands and interprets GWT RPC requests, inspects the payload for security check violations, and takes specified actions.

The Web App Firewall headers and cookies checks for GWT requests are similar to those for other request formats. After appropriate URL decoding and charset conversion, all the parameters in the string table are inspected. The GWT request body does not contain field names, just the field values. The input values can be validated against the specified format by using the Web App Firewall Field Format check, which can also be used to control the length of the input. The **Cross-site Scripting** and **SQL Injection** attacks in the inputs can be easily detected and thwarted by the Web App Firewall.

Learning and relaxation rules: Learning and deployment of relaxation rules are supported for GWT requests. Web App Firewall rules are in the form of <actionURL> <fieldName> mapping. The GWT request format does not have the field names and thus requires special handling. The Web App Firewall inserts dummy field names in the learned rules that can be deployed as relaxation rules. The -isRegex flag works as it does for non-GWT rules.

- Action URL:

  Multiple services responding to an RPC can be configured on the same web server. The HTTP request has the URL of the web server, not of the actual service handling the RPC. Therefore, relaxation is not applied on the basis of the HTTP request URL, because that would relax all the services on that URL for the target field. For GWT requests, the Web App Firewall uses the URL of the actual service found in the GWT payload, in the fourth field in the string table.

- Field name:

  Since the GWT request body contains only field values, the Web App Firewall inserts dummy field names such as 1, 2, and so on when recommending learned rules.

  Example of a GWT learned rule

```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157

5|0|8|http://localhost:8080/acdtest/|16878339
   F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.
   lang.Integer|onblur|
```



```
8
9     The learn data will be as follows:
10  > sh learningdata pr1 crossSiteScripting
11  Profile: pr1    SecurityCheck: crossSiteScripting
12  1)  Url:    http://localhost:8080/acdtest/  >> From GWT Payload.
13      Field:  10
14      Hits:   1
15   Done
16  <!--NeedCopy-->
```

Example of a GWT relaxation rule

```
bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex NOTREGEX
```

Log Messages: The Web App Firewall generates log messages for the security check violations that are detected in the GWT requests. A log message generated by a malformed GWT request contains the string "GWT" for easy identification.

Example of a Log message for malformed GWT request:

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT ns
0-PPE-0 : APPFW Message 696 0 : "GWT RPC request with malformed payload. <
blocked>"
```

Difference in processing of GWT vs non-GWT requests:

The same payload can trigger different Web App Firewall security check violations for different Content-types. Consider the following example:

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|com
.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|
select|
```

Content-type: application/x-www-form-urlencoded:

A request sent with this content type results in a SQL violation if the SQL Injection Type is configured to use any of the four available options: SQLSplCharANDKeyword, SQLSplCharORKeyword, SQLKeyword, or SQLSplChar. The Web App Firewall considers '&' to be the field separator and '=' to be the name-value separator when processing the above payload. Since neither of these characters appears anywhere in the post body, the entire content is treated as a single field name. The field name in this request contains both an SQL special character (;) and an SQL Keyword (select). Therefore violations are caught for all four SQL Injection type options.

Content-type: text/x-gwt-rpc:

A request sent with this content type triggers an SQL violation only if the SQL injection type is set to one of the following three options: SQLSplCharORKeyword,SQLKeyword, or SQLSplChar. No violation is triggered if the SQL injection type is set to SQLSplCharANDKeyword, which is the default option. The

Web App Firewall considers the vertical bar | to be the field separator for the above payload in the GWT request. Therefore, the post body is divided into various form-field values, and form-field names are added (in accordance with the convention described earlier). Because of this splitting, the SQL special character and SQL keyword become parts of separate form fields.

Form field 8: `java.lang.String%3b -\> %3b is the (;)char`

Form Field 10: `select`

As a result, when SQL Injection Type is set to **SQLSplChar**, field 8 indicates the SQL violation. For **SQLKeyword**, field 10 indicates the violation. Either of these two fields can indicate a violation if the SQL Inject type is configured with the **SQLSplCharORKeyword** option, which looks for the presence of either a keyword or a special character. No violation is caught for the default **SQLSplCharAND-Keyword** option, because there is no single field that has a value that contains both **SQLSplChar** and **SQLKeyword** together.

Tips:

- No special Web App Firewall configuration is needed to enable GWT support.
- The Content-type must be text/x-gwt-rpc.
- Learning and deploying of the relaxation rules for all the pertinent Web App Firewall security checks applied to GWT payload works the same as it does for the other supported content-types.
- Only POST requests are considered valid for GWT. All other request methods are blocked if the content-type is text/x-gwt-rpc.
- GWT requests are subject to the configured POST body limit of the profile.
- The sessionless setting for the security checks is not applicable and will be ignored.
- CEF log format is supported for the GWT log messages.

Cookie Protection

September 14, 2021

Cookie is a small packet data sent from a web server to a client browser. Cookies carry sensitive data such as passwords, user authentication details, and credentials over an HTTP connection and stored in a web browser. Hence it is highly important to protect cookies from attackers who steal information.

Cookie consistency check: Examines cookies returned with user requests to verify that they match the cookies your Web server set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the Web server. For more information, see Cookie consistency check topic.

Cookie hijacking protection: Hijacking refers to a situation where an attacker gains an unauthorized access to cookies. To protect cookie from authorized access, the Citrix ADC Web App Firewall (WAF)

challenges the TLS connection from the client along with WAF cookie consistency validation. For every new client request, the appliance validates the TLS connection and also verifies the consistency of application and session cookie in the request. For more information, see Cookie hijacking protection topic.

SameSite cookie attribute: The `SameSite` attribute in the Set-Cookie HTTP response allows you to declare if your cookie must be restricted to a first-party or same-site context. The cookie setting mitigates attacks and provides a secured web communication. For more information, see SameSite cookie attribute topic.

Cookie consistency check

September 14, 2021

The Cookie Consistency check examines cookies returned by users, to verify that they match the cookies that your website set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the web server. You can also configure the Cookie Consistency check to transform all of the server cookies that it processes, by encrypting the cookies, proxying the cookies, or adding flags to the cookies. This check applies to requests and responses.

An attacker would normally modify a cookie to gain access to sensitive private information by posing as a previously authenticated user, or to cause a buffer overflow. The Buffer Overflow check protects against attempts to cause a buffer overflow by using a long cookie. The Cookie Consistency check focuses on the first scenario.

If you use the wizard or the GUI, in the
Modify Cookie Consistency Check dialog box, on the
General tab you can enable or disable the following actions:

- Block
- Log
- Learn
- Statistics
- Transform. If enabled, the Transform action modifies all cookies as specified in the following settings:
  - **Encrypt Server Cookies.** Encrypt cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list, before forwarding the response to the client. Encrypted cookies are decrypted when the client sends a subsequent request, and the decrypted cookies are reinserted into the request before it is forwarded to the protected web server. Specify one of the following types of encryption:
    - **None.** Do not encrypt or decrypt cookies. The default.

    * **Decrypt only.** Decrypt encrypted cookies only. Do not encrypt cookies.
    * **Encrypt session only.** Encrypt session cookies only. Do not encrypt persistent cookies. Decrypt any encrypted cookies.
    * **Encrypt all.** Encrypt both session and persistent cookies. Decrypt any encrypted cookies.

      Note: When encrypting cookies, the Web App Firewall adds the **HttpOnly** flag to the cookie. This flag prevents scripts from accessing and parsing the cookie. The flag therefore prevents a script-based virus or trojan from accessing a decrypted cookie and using that information to breach security. This is done regardless of the Flags to Add in Cookies parameter settings, which are handled independently of the Encrypt Server Cookies parameter settings.
- **Proxy Server Cookies.** Proxy all non-persistent (session) cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list. Cookies are proxied by using the existing Web App Firewall session cookie. The Web App Firewall strips session cookies set by the protected web server and saves them locally before forwarding the response to the client. When the client sends a subsequent request, the Web App Firewall reinserts the session cookies into the request before forwarding it to the protected web server. Specify one of the following settings:
  - **None.** Do not proxy cookies. The default.
  - **Session only.** Proxy session cookies only. Do not proxy persistent cookies
    Note: If you disable cookie proxying after having enabled it (set this value to None after it was set to Session only), cookie proxying is maintained for sessions that were established before you disabled it. You can therefore safely disable this feature while the Web App Firewall is processing user sessions.
- **Flags to Add in Cookies**. Add flags to cookies during transformation. Specify one of the following settings:
  - **None.** Do not add flags to cookies. The default.
  - **HTTP only.** Add the HttpOnly flag to all cookies. Browsers that support the HttpOnly flag do not allow scripts to access cookies that have this flag set.
  - **Secure**. Add the Secure flag to cookies that are to be sent only over an SSL connection. Browsers that support the Secure flag do not send the flagged cookies over an insecure connection.
  - **All.** Add the HttpOnly flag to all cookies, and the Secure flag to cookies that are to be sent only over an SSL connection.

If you use the command-line interface, you can enter the following commands to configure the Cookie Consistency Check:

- `set appfw profile <name> -cookieConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`
- `set appfw profile <name> -cookieTransforms ([**ON**] | [**OFF**])`

- set appfw profile <name> -cookieEncryption ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])
- set appfw profile <name> -cookieProxying ([**none**] | [**sessionOnly**])
- set appfw profile <name> -addCookieFlags ([**none**] | [**httpOnly**] | [**secure**] | [**all**])

To specify relaxations for the Cookie Consistency check, you must use the GUI. On the Checks tab of the Modify Cookie Consistency Check dialog box, click Add to open the Add Cookie Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Cookie Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Cookie Consistency check relaxations:

- **Logon Fields.** The following expression exempts all cookie names beginning with the string logon_ followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- **Logon Fields (special characters).** The following expression exempts all cookie names beginning with the string türkçe-logon_ followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^txC3xBCrkxC3xA7e-logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- **Arbitrary strings.** Allow cookies that contain the string sc-item_, followed by the ID of an item that the user has added to his shopping cart ([0-9A-Za-z]+), a second underscore (_), and finally the number of these items he wants ([1-9][0-9]?), to be user-modifiable:

```
^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
<!--NeedCopy-->
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do

not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

> **Important**
>
> In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), Web App Firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".
>
> **Note** that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.**
>
> **Note**
>
> **Sessionless Cookie Consistency:** The cookie consistency behavior has changed in release 11.0. In earlier releases, the cookie consistency check invokes sessionization. The cookies are stored in the session and signed. A "wlt_" suffix is appended to transient cookies and a "wlf_" suffix is appended to the persistent cookies before they are forwarded to the client. Even if the client does not return these signed wlf/wlt cookies, the Web App Firewall uses the cookies stored in the session to perform the cookie consistency check.
>
> In release 11.0, the cookie consistency check is sessionless. The Web App Firewall now adds a cookie that is a hash of all the cookies tracked by the Web App Firewall. If this hash cookie or any other tracked cookie is missing or tampered with, the Web App Firewall strips the cookies before forwarding the request to the back end server and triggers a cookie-consistency violation. The server treats the request as a new request and sends new Set-Cookie header(s). The Cookie Consistency check in Citrix ADC version 13.0, 12.1, and NetScaler 12.0 and 11.1 does not have sessionless option.

Cookie hijacking protection

September 14, 2021

Cookie hijacking protection mitigates cookie stealing attacks from hackers. In the security attack, an attacker takes over a user session to gain unauthorized access to a web application. When a user browses a website, for example banking application, the website establishes a session with the browser. During the session, the application saves the user details such as login credentials, page visits in a cookie file. The cookie file is then sent to the client browser in the response. The browser stores the cookies to maintain active sessions. The attacker can steal these cookies either manually from the cookie store of the browser or through some rouge browser extension. The attacker then use these cookies to gain access into the user's web application sessions.

To mitigate cookie attacks, the Citrix ADC Web App Firewall (WAF) challenges the TLS connection from the client along with WAF cookie consistency validation. For every new client request, the appliance validates the TLS connection and also verifies the consistency of application and session cookie in the request. If an attacker tries to mix and match application cookies and session cookies stolen from the victim, the cookie consistency validation fails, and the configured cookie hijack action is applied. For more information about cookie consistency, see Cookie Consistency Check topic.

> **Note:**
>
> The Cookie hijacking feature supports logging and SNMP traps. For more information about logging, see ADM topic and for more information about SNMP configuration see SNMP topic.

Limitations

- JavaScript must be enabled in the client browser.
- Cookie Hijacking protection is not supported on TLS version 1.3.
- Limited support for the Internet Explorer (IE) browser because the browser does not reuse the SSL connections. Results in multiple redirects sent for a request eventually leading to a "MAX REDIRECTS EXCEEDED" error in the IE browser.

How cookie hijacking protection works

The following scenarios explain how cookie hijacking protection works in a Citrix ADC appliance.

Scenario 1: User accessing the first webpage without session cookie



1. The user attempts to authenticate into a web application and begins to access the first webpage without any ADC session cookie in the request.
2. When the request is received, the appliance creates an Application Firewall session with a session cookie ID.
3. This initiates a TLS connection for the session. Since the JavaScript is not sent and ran on the client browser, the appliance marks the TLS connection as validated and no challenge is required.

   > **Note:**
   >
   > Even if an attacker tries to send all the app cookie IDs from a victim without sending the session cookie, the appliance detects the issue and strips off all the app cookies in the request before forwarding the request to the back-end server. The back end server considers this request without no app cookie and takes necessary as per its configuration.

4. When the back-end server sends a response, the appliance receives the response and forwards it with a JavaScript session token and seed cookie. The appliance then marks the TLS connection as verified.
5. When the client browser receives the response, the browser runs the JavaScript and generates a morphed cookie ID using the session token and seed cookie.
6. When a user sends a subsequent request over the TLS connection, the appliance bypasses the morphed cookie validation. This is because the TLS connection is already validated.

Scenario 2: User accessing successive webpages over new TLS connection with session cookie



1. When a user sends an HTTP request for successive pages over a new TLS connection, the browser sends session cookie ID and morphed cookie ID.
2. Since this is a new TLS connection, the appliance detects the TLS connection and challenges the client with redirect response with seed cookie.
3. The client upon receiving the response from the ADC, calculates the morphed cookie using the session's token and new seed cookie.
4. The client then sends this newly calculated morphed cookie along with a session ID.
5. If the morphed cookie calculated within the ADC appliance and the one sent over the request matches, then the TLS connection is marked as verified.
6. If the calculated morphed cookie differs from the one present in the client request, then validation fails. After which, the appliance sends the challenge back to the client, to send a proper morphed cookie.

Scenario 3: Attacker impersonating as a non-authenticated user



1. When a user authenticates into the web application, the attacker uses different techniques to

steal the cookies and replay them.

2. Since this is a new TLS connection from the attacker, the ADC sends a redirect challenge along with a new seed cookie.
3. Since the attacker does not have JavaScript running, the response from the attacker for the redirected request does not contain the morphed cookie.
4. This results in morphed cookie validation failure at the ADC appliance side. The appliance again sends a redirect challenge to the client.
5. If the number of morphed cookie validation attempts exceeds the threshold limit, the appliance flags the status as cookie hijacking.
6. If the attacker tries to mix and match application cookies and session cookies stolen from the victim, the cookie consistency check fails, and the appliance applies the configured cookie hijack action.

Scenario 4: Attacker impersonating as an authenticated user



1. Attackers can also attempt to authenticate into a web application as a genuine user and replay the victim's cookies to gain access to the web session.
2. The ADC appliance detects such impersonated attackers also. Although a verified TLS connection is used by the attacker in replaying a victim's cookie, the ADC appliance still verifies if the session cookie and application cookie in the request are consistent. The appliance verifies the consistency of an application cookie using the session cookie in the request. Since the request contains an attacker's session cookie and a victim's app cookie, the cookie consistency validation fails.
3. As a result, the appliance applies the configured cookie hijack action. If the configured action is set as "block," then the appliance strips off all the application cookies and sends the request to the back-end Server.
4. The back-end server receives a request with no application cookie and so it responds an error response to the attacker, such as "User not logged in".

Configure cookie hijacking by using the CLI

You can select a specific application firewall profile and set one or more actions that prevent cookie hijacking.

At the command prompt, type:

```
set appfw profile <name> [-cookieHijackingAction <action-name> <block | log | stats | none>]
```

> **Note:**
>
> By default, the action is set to “none.”

Example:

```
set appfw profile profile1 - cookieHijackingAction Block
```

Where, action types are:

Block: Block connections that violate this security check.
Log: Log violations of this security check.
Stats: Generate statistics for this security check.
None: Disable all actions for this security check.

Configure cookie hijacking by using the Citrix ADC GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. On the **Profiles** page, select a profile and click **Edit**.
3. On the **Citrix Web App Firewall Profile** page, go to **Advanced Settings** section and click **Security Checks**.



4. In the **Security Checks** section, select **Cookie Hijacking** and then click **Action** settings.
5. In the **Cookie Hijacking Settings** page, select one or more actions to prevent cookie hijacking.
6. Click **OK**.



Add a relaxation rule for cookie consistency validation by using the Citrix ADC GUI

To handle false positives in cookie consistency validation, you can add a relaxation rule for cookies that can be exempted from cookie validation.

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. On the **Profiles** page, select a profile and click **Edit**.

3. On the **Citrix Web App Firewall Profile** page, go to **Advanced Settings** section and click **Relaxation rules**.
4. In the **Relaxation Rules** section, select **Cookie Consistency** and click **Action**.
5. In the **Cookie Consistency Relaxation Rule** page, set the following parameters.
   a) Enabled. Select if you want to enable the relaxation rule.
   b) Is Cookie Name Regex. Select if the cookie name is a regular expression.
   c) Cookie Name. Enter the name of the cookie that can be exempted from cookie validation.
   d) Regex Editor. Click this option to provide the regular expression details.
   e) Comments. A brief description about the cookie.
6. Click **Create** and **Close**.

View cookie hijacking traffic and violation statistics by using the CLI

View security traffic and security violation details in a tabular or graphical format.

To view security statistics:

At the command prompt, type:

```
stat appfw profile profile1
```

Appfw profile Traffic Statistics	Rate (/s)	Total
Requests	0	0
Request Bytes	0	0
Responses	0	0
Response Bytes	0	0
Aborts	0	0
Redirects	0	0
Long Term Ave Response Time (ms)	–	0
Recent Ave Response Time (ms)	–	0

HTML/XML/JSON Violation Statistic	Rate (/s)	Total
Start URL	0	0
Deny URL	0	0
Referer header	0	0
Buffer overflow	0	0
Cookie consistency	0	0
Cookie hijacking	0	0
CSRF form tag	0	0
HTML Cross-site scripting	0	0
HTML SQL injection	0	0
Field format	0	0
Field consistency	0	0
Credit card	0	0
Safe object	0	0
Signature Violations	0	0
Content Type	0	0
JSON Denial of Service	0	0
JSON SQL injection	0	0
JSON Cross-Site Scripting	0	0
File Upload Types	0	0
Infer Content Type XML Payload	0	0
HTML CMD Injection	0	0
XML Format	0	0
XML Denial of Service (XDoS)	0	0
XML Message Validation	0	0
Web Services Interoperability	0	0
XML SQL Injection	0	0
XML Cross-Site Scripting	0	0
XML Attachment	0	0

HTML/XML/JSON Violation Statistic	Rate (/s)	Total
SOAP Fault Violations	0	0
XML Generic Violations	0	0
Total Violations	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
Start URL logs	0	0
Deny URL logs	0	0
Referer header logs	0	0
Buffer overflow logs	0	0
Buffer overflow logs	0	0
Cookie consistency logs	0	0
Cookie hijacking logs	0	0
CSRF form tag logs	0	0
HTML cross-site scripting logs	0	0
HTML cross-site scripting transform logs	0	0
HTML SQL Injection logs	0	0
HTML SQL transform logs	0	0
Field format logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0

HTML/XML/JSON Log Statistics	Rate (/s)	Total
JSON Cross-Site Scripting logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload L	0	0
HTML Command Injection logs	0	0
XML Format logs	0	0
XML Denial of Service(XDoS) logs	0	0
XML Message Validation logs	0	0
WSI logs	0	0
XML SQL Injection logs	0	0
XML cross-site scripting logs	0	0
XML Attachment logs	0	0
SOAP Fault logs	0	0
XML Generic logs	0	0
Total log messages	0	0

Server Error Response Statistics	Rate (/s)	Total
HTTP Client Errors (4xx Resp)	0	0
HTTP Server Errors (5xx)	0	0

View cookie hijacking traffic and violation statistics by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select a **Web App Firewall** profile and click **Statistics**.
3. The **Citrix Web App Firewall Statistics** page displays the cookie hijacking traffic and violation details.
4. You can select **Tabular View** or switch to **Graphical View** to display the data in a tabular or

graphical format.



SameSite cookie attribute

September 14, 2021

For secure web communication, Google has mandated the usage of the `SameSite` cookie attribute. By complying with Google Chrome's new `SameSite` policy, the Citrix ADC appliance can manage third-party cookies with the `SameSite` attribute set in the `set-cookie` header. The cookie setting mitigates attacks and provides a secured web communication.

Until February 2020, the `SameSite` attribute was not explicitly set in the cookie. The browser took the default value as "None." However, with certain browser upgrade, such as Google Chrome 80, there is a change in the default cross-domain behavior in cookies.

Setting cookie attribute value

The `SameSite` attribute is set to one of the following values and for the Google Chrome browser, the default value is set as "Lax."

None. Indicates the browser to use the cookie for requests in the cross-site context only on secure connections.

Lax. Indicates the browser to use the cookie for requests in the same-site context. In the cross-site context, only safe HTTP methods like GET request can use the cookie.

Strict. Use the cookie only when the user is requesting for the domain explicitly.

> **Note:**
>
> If set-cookies (including firewall session cookies) have the `SameSite` attribute and if the `addcookiesamesite` attribute flag is enabled in the Web Application Firewall profile, then the `SameSite` attribute is overwritten according to the value configured in the profile.

Configure the SameSite attribute in the Web App Firewall profile by using the CLI

To configure the `SameSite` attribute, you must complete the following steps:

1. Enable the `SameSite` cookie attribute.
2. Set the cookie attribute for the appfw session cookies.

Enable the 'Samesite' cookie attribute

At the command prompt, type:

```
set appfw profile <profile-name> -insertCookieSameSiteAttribute ( ON | OFF)
```

Example:

```
set appfw profile p1 -insertCookieSameSiteAttribute ON
```

Set same site cookie attribute value for Web Application Firewall session cookies

At the command prompt, type:

```
set appfw profile <profile-name> - cookieSameSiteAttribute ( LAX | NONE |
STRICT )
```

Example:

```
set appfw profile p1 - cookieSameSiteAttribute LAX
```

Where attribute types are,

None. Cookie attribute SameSite is set to "none" and marked secure for all WAF and application cookies.

Lax. Cookie attribute SameSite is set to "Lax" for all WAF and application cookies.

Strict. Cookie attribute SameSite is set to "Lax" for all WAF and application cookies.

Configure the SameSite cookie attribute in the Web App Firewall profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.

2. In the details pane, select a profile and click **Edit**.
3. In the **Citrix Web App Firewall Profile** page, click **Profile Settings** under **Advanced Settings**.
4. In **Profile Settings** section set the following parameters:

   a. Insert the cookie `Samesite` attribute. Select the check box to enable the cookie `Samesite` attribute.

   b. Cookie Samesite Attribute. Select an option from the drop-down list to set the `Samesite` cookie value.

5. Click **OK** and **Done**.



Data leak prevention checks

September 14, 2021

The data-leak-prevention checks filter responses to prevent leaks of sensitive information, such as credit card numbers and social security numbers, to unauthorized recipients.

Credit card check

September 14, 2021

If you have an application that accepts credit cards, or your websites have access to database servers that store credit card numbers, you must use Data Leak Prevention (DLP) measures and configure protection for each type of credit card that you accept.

The Citrix Web App Firewall Credit Card check prevents attackers from exploiting Data Leak Prevention flaws to obtain credit card numbers of your customers. By following simple configuration steps, you can enforce protection of one or more of the following credit cards: 1) Visa, 2) Master Card, 3) Discover, 4) American Express (Amex), 5) JCB, and 6) Diners Club.

The Credit Card security check examines server responses to identify instances of the target credit card numbers, and applies a specified action when such a number is found. The action can be to transform the response by X'ing out all but the last group of digits in the credit card number, or to block the response if it contains more than a specified number of credit card numbers. If you specify both, the block action takes precedence. The Maximum credit cards allowed per page setting determines when the block action is invoked. The default setting, 0 (no credit card numbers allowed on the page), is the safest, but you can allow up to 255. Depending on where the violation is detected in the response and the block action gets triggered, you might get fewer than the maximum allowed number of credit cards in the response.

To avoid false positives, you can apply relaxations to exempt specific numbers from the Credit Card check. For example, a social security number, purchase order number, or Google account number might be similar to a credit card number. You can specify individual numbers or use a regular expression to indicate the string of digits to be bypassed when processing the response URL for credit card inspection.

If you're not sure which credit card numbers to exempt, you can use the learn feature to generate recommendations based on the learned data. To get optimal benefit without compromising performance, you might want to enable this option for a short time to get a representative sample of the rules, and then deploy the relaxations and disable learning.

If you enable the log feature, the Credit Card check generates log messages indicating the actions that it takes. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate thwarted attempts to gain access. By default, the doSecureCreditCardLogging parameter is ON, so the credit card number is not included in the log message generated by the safe commerce (Credit Card) violation.

The stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack.

To configure the Credit Card security check for protecting your application, configure the profile that

governs inspecting the traffic to and from this application.

> **Note**:
>
> A website that does not access a SQL database usually does not have access to sensitive private information such as credit card numbers.

Using the command line to configure the credit card check

In the command line interface, you can use either the set appfw profile command or the add appfw profile command to activate credit-card checking and specify which actions to perform. You can use the unset appfw profile command to revert back to the default settings. To specify relaxations, use the bind appfw command to bind credit card numbers to the profile.

To configure a credit card check by using the command line

Use either the set appfw profile command or the add appfw profile command, as follows:

- `set appfw profile <name> -creditCardAction ( ([block][learn] [log][ stats])| [none])`
- `set appfw profile <name> -creditCard (VISA | MASTERCARD | DISCOVER | AMEX | JCB | DINERSCLUB)`
- `set appfw profile <name> -creditCardMaxAllowed <integer>`
- `set appfw profile <name> -creditCardXOut ([ON] | [OFF])<name> -doSecureCreditCard ([ON] | [OFF])`
- To configure a Credit Card relaxation rule by using the command line

  Use the bind command to bind the credit card number to the profile. To remove a credit card number from a profile, use the unbind command, with the same arguments that you used for the bind command. You can use the show command to display the credit card numbers bound to a profile.

- To bind a credit card number a profile

  `bind appfw profile <profile-name> -creditCardNumber <any number/regex> “<url>”`

  Example: bind appfw profile test_profile -creditCardNumber 378282246310005 `http://www.example.com/credit\\_card\\_test.html`

  - To unbind a credit card number from a profile

    `unbind appfw profile <profile-name> -creditCardNumber <credit card number / regex> <url>`

- To show the list of credit card numbers bound to a profile.

  ```
  show appfw profile <profile>
  ```

Using the GUI to configure the credit card check

In the GUI, you configure the credit card security check in the pane for the profile associated with your application.

To add or modify the Credit Card security check by using the GUI

1. Navigate to **Web App Firewall** > **Profiles, highlight** the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a) If you just want to enable or disable Block, Log, Stats, and Learn actions for Credit Card, you can select or clear check boxes in the table, click **OK**, and then click **Save** and **Close** to close the **Security Check** pane.
   b) If you want to configure additional options for this security check, double click Credit Card, or select the row and click **Action Settings** to display additional options as follows:
      - Out—Mask any credit card number detected in a response by replacing each digit, except the digits in the final group, with the letter “X.
      - Maximum credit cards allowed per page—Specify the number of credit cards that can be forwarded to the client without triggering a block action.
      - Protected Credit Cards. Select or clear a check box to enable or disable protection for each type of credit card.
      - You can also edit the Block, Log, Stats and Learn actions in the Credit Card Settings pane.

        After making any of the above changes, click OK to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click OK to save all the changes you have made in the Security Checks section and then click Save and Close to close the Security Check pane.
3. In the **Advanced Settings** pane, click **Profile settings**. To enable or disable secure logging of credit card Numbers, select or clear the **Secure Credit Card Logging** check box. (By default, it is selected).

   Click **OK** to save the changes.

- To configure a Credit Card relaxation rule by using the GUI

  1. Navigate to **Web App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.

2. In the **Advanced Settings** pane, click **Relaxation Rules**. The Relaxation Rules table has a Credit Card entry. You can double click, or select this row and click **Edit** to access the **Credit Card Relaxation Rules** dialogue. You can perform Add, Edit, Delete, Enable, or Disable operations for relaxation rules.

Using the learn feature with the credit card check

When the learn action is enabled, the Web App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, if you want to exempt a specific string of digits from the Credit Card security check, you can by deploy the learned rule as a relaxation rule.

- To view or use learned data by using the command line interface

  ```
  show appfw learningdata <profilename> creditCardNumber
  ```

  ```
  rm appfw learningdata <profilename> -creditcardNumber <credit card
  number> "<url>"
  ```

  ```
  export appfw learningdata <profilename> creditCardNumber
  ```

- To view or use learned data by using the GUI

  1. Navigate to **Web App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
  2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the Credit Card entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

  You also have the option to show a summarized view of the learned relaxations by selecting the Credit Card entry in the Learned Rules table and clicking Visualizer to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the log feature with the credit card check

When the log action is enabled, the Credit Card security check violations are logged in the audit log as APPFW_SAFECOMMERCE or APPFW_SAFECOMMERCE_XFORM violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

The default setting for doSecureCreditCardLogging is ON. If you change it to OFF, both credit card number and type are included in the log message.

Depending on the settings configured for the Credit Card checks, the application-firewall generated log messages might include the following information:

- Response was blocked or not blocked.
- Credit card numbers were transformed (X'd out). A separate log message is generated for each transformed credit card number, so multiple log messages might be generated during processing of a single response.
- Response contained the maximum number of potential credit card numbers.
- Credit card numbers and their corresponding types.
- To access the log messages by using the command line

  Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Credit Card violations:

  - Shell

```
tail -f /var/log/ns.log                    grep SAFECOMMERCE
```

  -

- To access the log messages by using the GUI

  1. The Citrix GUI includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have a couple of options for accessing the Syslog Viewer: Navigate to the **target profile** > **Security Checks**. Highlight the Credit Card row and click Logs. When you access the logs directly from the Credit Card security check of the profile, it filters out the log messages and displays only the logs pertaining to these security check violations.
  2. You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.

     The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To access Credit Card security check violation log messages, filter by selecting APPFW in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the APPFW_SAFECOMMERCE and APPFW_SAFECOMMERCE_XFORM check boxes and click the

Apply button, only log messages pertaining to the Credit Card security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as Module and EventType, appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Example of a Native format log message when the response is not blocked

```
May 29 01:26:31 <local0.info> 10.217.31.98 05/29/2015:01:26:31 GMT ns
    0-PPE-0 :
default APPFW APPFW_SAFECOMMERCE 2181 0 :  10.217.253.62 1098-PPE0
4erNfkaHy0IeGP+nv2S9Rsdu77I0000 pr_ffc http://aaron.stratum8.net/FFC/
    CreditCardMind.html
Maximum number of potential credit card numbers seen <not blocked>
<!--NeedCopy-->
```

Example of a CEF format log message when the response is transformed

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE_XFORM|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Transformed (xout) potential credit card numbers seen in server
    response
cn1=66 cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002
cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

Example of a CEF format log message when the response is blocked. The credit card number and type can be seen in the log, because the doSecureCreditCardLogging parameter is disabled.

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Credit Card number 4505050504030302 of type Visa is seen in
    response cn1=68
cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```


Statistics for the credit card violations

When the stats action is enabled, the corresponding counter for the Credit Card check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled and the Max Allowed credit card setting is 0, the request for a page that contains 20 credit card numbers increments the stats counter by one when the page is blocked as soon as the first credit card number is detected. However, if block is disabled and transform is enabled, processing the same request increments the statistics counter for logs by 20, because each credit card transformation generates a separate log message.

- To display Credit Card statistics by using command line

  At the command prompt, type:

  ```
  sh appfw stats
  ```

  To display stats for a specific profile, use the following command:

  ```
  stat appfw profile <profile name>
  ```

  To display Credit Card statistics by using GUI

  1. Navigate to **System** > **Security** > **Web App Firewall**.
  2. In the right pane, access the **Statistics** Link.
  3. Use the scroll bar to view the statistics about Credit Card violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the Credit Card security check:

- The Web App Firewall enables you to protect credit card information and detect any attempts to access this sensitive data.
- To use the Credit Card protection check, you must specify at least one type of credit card and an action. The check is then applied to HTML, XML, and Web 2.0 profiles.

You can pipe the output of sh appfw profile command and grep for CreditCard to see all the Credit Card specific configuration. For example, sh appfw profile my_profile	grep CreditCard displays the configured settings of various parameters as well as the relaxation rules pertaining to the Credit Card check for the Web App Firewall profile named my_profile.

-

- You can exclude specific numbers from Credit Card inspection without bypassing the security check inspection for the rest of the credit card numbers.
- Relaxation is available for all Web App Firewall protected credit card patterns. In the GUI, you can use the visualizer to specify Add, Edit, Delete, Enable, or Disable operations on relaxation rules.
- The Web App Firewall learning engine can monitor the outgoing traffic to recommend rules based on observed violations. Visualizer support is also available for managing the learned credit card rules in the GUI. You can edit and deploy the learned rules, or skip them after careful inspection.
- The setting for number of allowed credit cards applies to each response. It does not pertain to the cumulative total of credit card numbers observed during the entire user session.
- The number of X'd out digits depends on the length of the credit card numbers. Ten digits are X'd out for credit cards that have 13 through 15 digits. Twelve digits are X'd out for credit cards that have 16 digits. If your application does not require sending the entire credit card number in the response, Citrix recommends that you enable this action to mask the digits in the credit card numbers.
- The X-out operation transforms all the credit cards and works independently of the configured settings for the maximum number of allowed credit cards. For example, if there are 4 credit cards in the response and the creditCardMaxAllowed parameter is set to 10, all 4 credit cards are X'd-out, but they are not blocked. If the credit card numbers are spread out in the document, a partial response with X'd-out numbers might be sent to the client before the response is blocked.
- Do not disable the doSecureCreditCardLogging parameter before due consideration. When this parameter is turned off, the credit card numbers are displayed and are accessible in the log messages. These numbers are not masked in the logs, even if the X-out action is enabled. If you are sending logs to a remote syslog server, and the logs are compromised, the credit card numbers can be exposed.
- When the response page is blocked because of a Credit Card violation, the Web App Firewall does not redirect to the error page.

Safe object check

September 14, 2021

The Safe Object check provides user-configurable protection for sensitive business information, such as customer numbers, order numbers, and country-specific or region-specific telephone numbers or

postal codes. A user-defined regular expression or custom plug-in tells the Web App Firewall the format of this information and defines the rules to be used to protect it. If a string in a user request matches a safe object definition, the Web App Firewall either blocks the response, masks the protected information, or removes the protected information from the response before sending it to the user, depending on how you configured that particular safe object rule.

The Safe Object check prevents attackers from exploiting a security flaw in your web server software or on your website to obtain sensitive private information, such as company credit card numbers or social security numbers. If your websites do not have access to these types of information, you do not need to configure this check. If you have a shopping cart or other application that can access such information, or your websites have access to database servers that contain such information, you must configure protection for each type of sensitive private information that you handle and store.

> **Note:**
>
> A website that does not access an SQL database usually does not have access to sensitive private information.

The Safe Object Check dialog box is unlike that for any other check. Each safe object expression that you create is the equivalent of a separate security check, similar to the Credit Card check, for that type of information. If you use the wizard or the GUI, you add a new expression by clicking Add and configuring the expression in the Add Safe Object dialog box. You modify an existing expression by selecting it, then clicking Open, and then configuring the expression in the Modify Safe Object dialog box.

In the Safe Object dialog box for each safe object expression, you can configure the following:

- **Safe Object Name.** A name for your new safe object. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 255 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.
- **Actions.** Enable or disable the Block, Log, and Statistics actions, and the following actions:
  - **X-Out.** Mask any information that matches the safe object expression with the letter “X”.
  - **Remove.** Remove any information that matches the safe object expression.
- **Regular Expression.** Enter a PCRE-compatible regular expression that defines the safe object. You can create the regular expression in one of three ways: by typing the regular expression directly into the text box, by using the **Regex Tokens** menu to enter regular expression elements and symbols directly into the text box, or by opening the Regular Expressions Editor and using it to construct the expression. The regular expression must consist of ASCII characters only. Do not cut and paste characters that are not part of the basic 128-character ASCII set. If you want to include non-ASCII characters, you must manually type those characters in PCRE hexadecimal character encoding format.
  Note: Do not use start anchors (^) at the beginning of Safe Object expressions, or end anchors

($) at the end of Safe Object expressions. These PCRE entities are not supported in Safe Object expressions, and if used, will cause your expression not to match what it was intended to match.

- **Maximum Match Length.** Enter a positive integer that represents the maximum length of the string that you want to match. For example, if you want to match U.S. social security numbers, enter the number eleven (11) in this field. That allows your regular expression to match a string with nine numerals and two hyphens. If you want to match California driver's license numbers, enter the number eight (8).

  > **Caution:**
  >
  > If you do not enter a maximum match length in this field, the Web App Firewall uses a default value of one (1) when filtering for strings that match your safe object expressions. As a result, most safe object expressions fail to match their target strings.

You cannot use the command-line interface to configure the Safe Object check. You must configure it by using either the Web App Firewall wizard or the GUI.

Following are examples of Safe Object check regular expressions:

- Look for strings that appear to be U.S. social security numbers, which consist of three numerals (the first of which must not be zero), followed by a hyphen, followed by two more numerals, followed by a second hyphen, and ending with a string of four more numerals:

```
1  [1-9][0-9]{
2  3,3 }
3  -[0-9]{
4  2,2 }
5  -[0-9]{
6  4,4 }
7
8  <!--NeedCopy-->
```

- Look for strings that appear to be California driver's license IDs, which start with a letter and are followed by a string of exactly seven numerals:

```
1  [A-Za-z][0-9]{
2  7,7 }
3
4  <!--NeedCopy-->
```

- Look for strings that appear to be customer IDs which, consist of a string of five hexadecimal characters (all the numerals and the letters A through F), followed by a hyphen, followed by a three-letter code, followed by a second hyphen, and ending with a string of ten numerals:

```
1  [0-9A-Fa-f]{
```



```
5,5 }
-[A-Za-z]{
3,3 }
-[0-9]{
10,10 }

<!--NeedCopy-->
```

> **Caution:**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write to ensure that they define exactly the type of string you want to add as a safe object definition, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you did not want or expect, such as blocking access to web content that you did not intend to block.

Advanced form protection checks

September 14, 2021

The advanced Form Protection checks examine web form data to prevent attackers from compromising your system by modifying the web forms on your websites or sending unexpected types and quantities of data to your website in a form.

> **Note**
>
> SQL, cross-site scripting, FFC, and FieldFormat protection checks are applied if the **Exclude Upload Files From Security Checks** is unset.

A file upload is also a form element that has control-name **name** field that is submitted as part of form submission.

Refer to this page, for more information: Forms

Field formats check

September 14, 2021

The Field Formats check verifies the data that users send to your websites in web forms. It examines both the length and type of data to ensure that it is appropriate for the form field in which it appears. If the Web App Firewall detects inappropriate web form data in a user request, it blocks the request.

By preventing an attacker from sending inappropriate web form data to your website, the Field Formats check prevents certain types of attacks on your website and database servers. For example, if a particular field expects the user to enter a phone number, the Field Formats check examines the user-submitted input to ensure that the data matches the format of a phone number. If a particular field expects a first name, the Field Formats check ensures that the data in that field is of a type and length appropriate for a first name. It does the same thing for each form field that you configure it to protect.

This check applies to HTML requests only. It does not apply to XML requests. You can configure Field Format Checks in HTML profiles or Web 2.0 profiles to inspect HTML payload for protecting your applications. The Web App Firewall also supports Field Format Check protection for Google Web Toolkit (GWT) applications.

The Field Formats check requires that you enable one or more actions. The Web App Firewall examines the submitted inputs and applies the specified actions.

> **Note**
>
> Field format rules are tightening rules. Adding them to relaxation list from learned data acts as a blocking rule.
>
> To relax field format rules, please remove particular "fieldname" from the fieldformat relaxations list.

You have the option to set the default field formats to specify Field Type and the minimum and maximum length of data expected in each form field on each web form that you want to protect. You can deploy relaxation rules to configure a Field Format for an individual field of a specific form. Multiple rules can be added to specify the field name, the action URL, and Field Formats. Specify Field Formats to accept different types of inputs in different form fields. The learning feature can provide recommendations for the relaxation rules.

Field Format Actions—You can enable Block, Log, Stats, and Learn actions. At least one of these actions must be enabled to engage the Field Format Check protection.

- **Block**. If you enable block, the block action is triggered if the input does not conform to the specified Field Format. If a rule was configured for the target field, the input is checked against the specified rule. Otherwise, it is checked against the default field format specification. Any mismatch in the Field Type or min/max length specification results in blocking the request.
- **Log**. If you enable the log feature, the Field Format check generates log messages indicating the actions that it takes. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate malicious attempts to launch an attack.
- **Stats**. If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack, or you might have to revisit the configuration to see if the specified field format is too restrictive.

- **Learn**. If you are not sure which Field Types or minimum and maximum length values might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The Web App Firewall learning engine monitors the traffic and provides field format recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.
  Note: The Web App Firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Default Field Format—In addition to configuring the actions, you can configure the default Field Format to specify the type of data expected in all the form fields for your application. A Field Type can be selected as the Field Format type. Minimum length and Maximum length parameters can be used to specify the length of the allowed inputs. As an alternative to Field Types, you can use Character Maps to specify what's allowed in a field (except in cluster deployments).

- **Field Type**—Field Types are named expression to which you assign assigned priority values. Field Type expressions specify the allowed inputs and are matched against the submitted data to determine whether the received values are consistent with the allowed values. The Field Types are checked in the order of their priority numbers. A lower number indicates a higher priority. The Web App Firewall gives you the option to add your own Field Types and assign them the priorities you want. The priority value can range from 0 through 64000. The following built-in Field Types are provided to help simplify the configuration process:

```
> sh appfw fieldtype
1)      Name:  integer           Regex:  "^[+-]?[0-9]+$"
                Priority:  30            Comment:  Integer
                Builtin:  IMMUTABLE
2)      Name:  alpha             Regex:  "^[a-zA-Z]+$"
                Priority:  40            Comment:  "Alpha
                   characters"
             Builtin:  IMMUTABLE
3)      Name:  alphanum          Regex:  "^[a-zA-Z0-9]+$"
                Priority:  50            Comment:  "Alpha-numeric
                    characters"
                Builtin:  IMMUTABLE
4)      Name:  nohtml            Regex:  "^[^&<>]*$"
                Priority:  60            Comment:  "Not HTML"
                Builtin:  IMMUTABLE
5)      Name:  any               Regex:  "^.*$"
             Priority:  70            Comment:  Anything
                Builtin:  IMMUTABLE
```



```
17        Done
18    >
19    <!--NeedCopy-->
```

> **Note:** The built-in Field Types are IMMUTABLE. They cannot be modified or removed. Any Field Types that you add are MODIFIABLE. You can edit them or remove them.

Configuring a Field Type as a default Field Format might be useful when you have a PCRE expression that can identify the valid inputs in all or most of the form fields for your application and exclude the invalid inputs. For example, if all the inputs in your application forms are expected to contain only numbers and letters, you might want to use the built-in Field Type alphanum as the default Field Type. Any non-alphanumeric character such as a backslash () or semicolon ; in the input will trigger a violation. You can also add your own customized Field Types and use them to configure default Field Formats. For example, if you want to make the lowercase “x”, “y”, and “z” the only allowed alpha characters, you can configure a customized Field Type with regular expression “^[x-z]+$”. You can assign it a higher priority (lower priority number) then the built-in Field Types and use it as the default Field Type.

- **Minimum Length** — The default minimum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 0 by default, which allows the user to leave the field blank. Any higher setting forces users to fill in the field.

  Caution: If the minimum length value is 0 but the Field Type is integer, alpha, or alphanum, a request is blocked if any input field is left empty, despite the minimum length setting. That is because the regEx for these Field Types contains a + character, which means one or more characters. Distinguishing an integer from an alpha character requires at least one character.

- **Maximum Length**—The default maximum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 65535 by default.

  > **Note:** Characters vs bytes. The minimum and the maximum lengths for the field formats represent the number of bytes, not the number of characters. Languages that have greater than one-byte character representation can cause the limit to be exceeded with fewer characters than the number configured for the maximum value. For example, with double-byte character representation, the maximum value of 9 allows no more than 4 characters.
  > **Tip:** The GUI allows you to cut and paste UTF-8 characters directly into the GUI without having to convert them to hex.

- **Character Maps**: In addition to recommending the Field Types, the Web App Firewall learning engine offers you an additional option, Use Character Maps, to deploy the Format Check rules. A Character Map is a set of all the characters allowed in a particular form field. You can fine tune the Field format specification to allow or disallow specific characters by using Character Maps. A separate Character Map is generated for each form field. The alpha and numeric characters

are treated differently in Character Maps. If any alpha character is seen in the input, all alpha characters [A-Za-z] will be allowed by the recommended PCRE expression in the Character Map. Similarly, if any digit is included, all digits [0-9] will be allowed. Non-printable characters are specified by using the x construct. Only single Byte characters with values between 0-255 are considered for Character Map recommendations.

A Character Map can be more specific than the corresponding Field Type recommendation. In some situations, Character Maps might be a better option, because they give you tighter control over the set of characters allowed as inputs. The deployed character maps are displayed as strings that start with prefix "CM" followed by digits. The priority for the Character Maps starts at 10000. As with user-added Field Types, you can add, edit or remove a Character Map. Character Maps that are currently used in deployed rules cannot be modified or removed.

Note: Character Maps are not supported in cluster deployments.

> **Note**
>
> When you add a field formats rule with any built-in Field Type and use character map instead of Field Type and save it, the changes do not get saved and rule still shows with Field Type.
>
> When the character map matches one of the built-in type, the field type is reused instead of creating a new character map.

Using the command line to configure the field format check

In the command line interface, you can use the add appfw fieldtype command to add a new Field Type. You can use either the set appfw profile command or the add appfw profile command to configure the Field Format check and specify which actions to perform. You can use the unset appfw profile command to revert the configured settings back to their defaults. To specify a Field Format rule, use the bind appfw command to bind a Field Type to a form Field and the action URL, along with the minimum and maximum length specifications.

To add, remove or view a Field Type by using the command line:

Use the add command to add a Field Type. You must specify the Name, Regular expression and Priority when adding a new Field Type. You also have the option to add a Comment. You can use the show command to display the configured Field Types. You can also delete a Field Type by using the remove command, which requires only the Name of the Field Type.

```
add [appfw] fieldType <name> <regex> <priority> [-comment <string>]
```

where:

<regex> is a regular expression

<priority> is a positive_integer

Example:


```
add fieldtype "Cust_Zipcode" "^[0-9]{
 5 }
 [-][0-9]{
 4 }
 $" 4

-  show [appfw] fieldType [<name>]

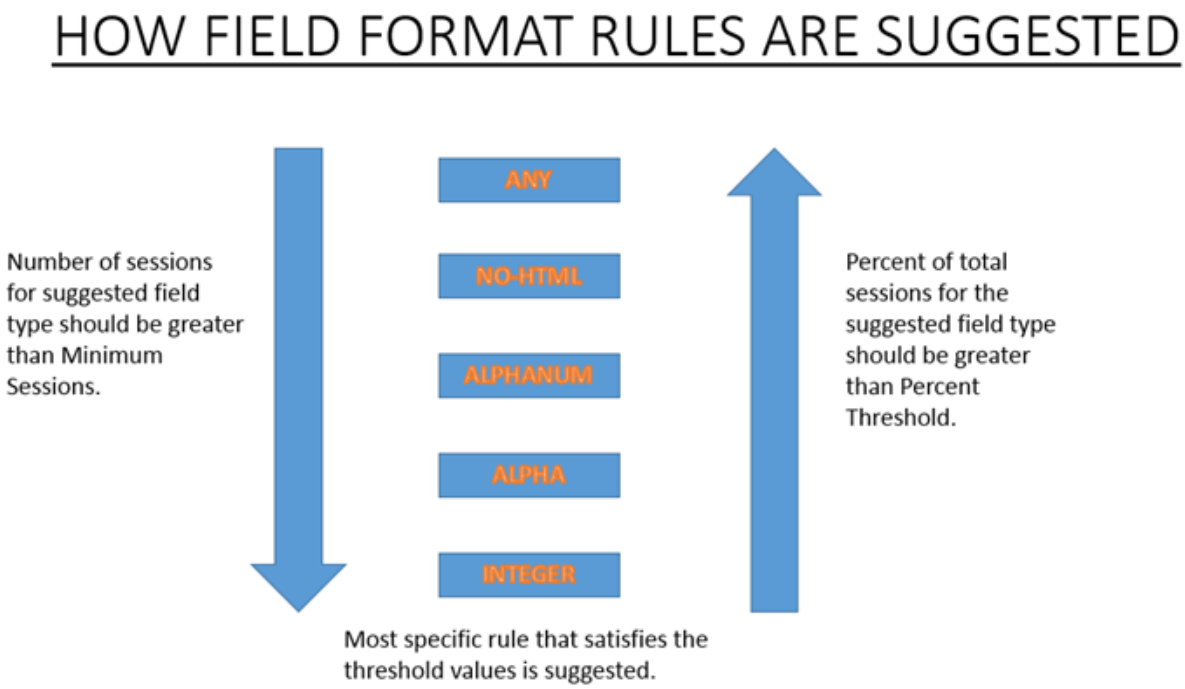
    Example: sh fieldtype

    sh appfw fieldtype

    sh appfw fieldtype cust_zipcode

-  `rm [appfw] fieldType <name>`

    Example: rm fieldtype cusT_ziPcode

    `rm appfw fieldtype cusT_ziPcode`
<!--NeedCopy-->
```

> **Note:** As shown above, use of “appfw” in the command is optional. For example, “Add FieldType” or “Add appfw fieldType” are both valid options. The names of the Field Types are case insensitive due to normalization. As shown in the above examples, Cust_Zipcode, cust_zipcode, and cUsT_ziPcode refer to the same Field Type.

To configure a Field Format check by using the command line

Use either the set appfw profile command or the add appfw profile command, as follows:

- `set appfw profile <name> -fieldFormatAction (([block] [learn] [log] [stats])| [none])`
- `set appfw profile <name>-defaultFieldFormatType <string>`
- `set appfw profile <name> -defaultFieldFormatMinLength <integer>`
- `set appfw profile <name> -defaultFieldFormatMaxLength <integer>`

To configure a Field Format relaxation rule by using the command line

```
bind appfw profile <name> (-fieldFormat <string> <formActionURL>  <
    fieldType>
[-fieldFormatMinLength <positive_integer>]  [-fieldFormatMaxLength <
    positive_integer>]
[-isRegex ( REGEX | NOTREGEX )])
<!--NeedCopy-->
```


Example:

```
bind appfw profile pr_ffc -fieldFormat "login_name" ".*/login.php"
    integer -fieldformatMinLength 3 -FieldformatMaxlength 6
<!--NeedCopy-->
```

Using the GUI to configure the field formats security check

In the GUI, you can manage the Field Types. You can also configure the Field Formats security check in the pane for the profile associated with your application.

To add, modify or remove a Field Type using the GUI

1. Navigate to Application Firewall node. In the Settings, click **Manage Field Types** to display the Configure Application Firewall Field Type dialogue box.
2. Click **Add** to add a new Field Type. Follow the instructions in this pane and click Create. You can also edit or delete any user-added Field Type if it is currently not being used by a deployed rule.

To add or modify the Field Formats security check by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a) If you just want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for Field Formats, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.
   b) If you want to configure additional options for this security check, double click Field Formats, or select the row and click Action Settings, to display the following options for **Default Field Format**:
      - **Field Type**—Select the Field Type that you want to configure as the default Field Type. You can select the built-in and user-defined Field Types. The deployed Character Maps are also included in the list and can be selected.
      - **Minimum Length**—Specify the minimum number of characters that must be in each field. Possible values: 0-65535.
      - **Maximum Length**— Specify the maximum number of characters that must be in each field. Possible values: 1-65535.
        You can also edit the **Block**, **Log**, **Stats** and **Learn** actions in the Field Formats Settings pane.

   After making any of the above changes, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all

the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure a Field Formats relaxation rule by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**. The Relaxation Rules table has a Field Formats entry. You can double click, or select this row and click the Edit button, to access the Field Formats Relaxation Rules dialogue. You can perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

For a consolidated view of all the relaxation rules, you can highlight the Field Formats row and click Visualizer. The visualizer for deployed relaxations offers you the option to Add a new rule or Edit an existing one. You can also Enable or Disable a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

Using the learn feature with the Field Formats Check

When the learn action is enabled, the Web App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as a Field Format relaxation rule.

Field formats learning enhancement—An Web App Firewall learning enhancement was introduced in release 11.0. In the previous releases, once the learned field format recommendation is deployed, the Web App Firewall learning engine stops monitoring the valid requests for the purpose of recommending new rules on the basis of the new data points. This limits the configured security protection, because the learning database does not include any representations of the new data seen in the valid requests processed by the security check.

Violations are no longer coupled with learning. The learning engine learns and makes recommendations for the field formats regardless of the violations. In addition to checking the blocked requests to determine whether the current field format is too restrictive and needs to be relaxed, the learning engine also monitors the allowed requests to determine whether the current field format is too permissive, and allows elevating the security by deploying a more restrictive rule.

Following is a summary of the Field Formats learning behavior:

No Field format is bound—The behavior remains unchanged in this scenario. All learn data is sent to the aslearn engine. The learning engine suggests a field format rule based on the data set.

Field format is bound: In the previous releases, observed data is sent to the aslearn engine only in the case of a violation. The learning engine suggests a field format rule based on the data set. In the 11.0 release, all data is sent to aslearn engine even if no violation is triggered. The learning engine suggests a field format rule based on the entire data set of all received inputs.

Use Case for learning enhancement:

If the initial field format learned rules are based on a small sample of data, a few non typical values might result in a recommendation that is too lenient for the target field. The ongoing learning allows the Web App Firewall to observe data points from every request to collect a representative sample for the learned recommendations. This is helpful in further tightening the security to deploy the optimal input format with an adequate range value.



The field format learning makes use of the priority of the Field Types as well as the configured settings of the following learning thresholds:

- **FieldFormatMinThreshold**—Minimum number of times a specific form field must be observed before a learned relaxation is generated. Default: 1.
- **FieldFormatPercentThreshold**—Percentage of times a form field matched a particular Field Type, before a learned relaxation is generated. Default: 0.

The field format rule recommendations are based on the following criteria:

- **Field Type recommendations**—The Field Type recommendations are determined by the assigned priorities of the existing Field Types and the specified Field Format thresholds. The priorities determine the order in which the Field Types are matched against the inputs. A lower number specifies a higher priority. For example, Field Type integer has the higher priority (30) and is therefore evaluated before Field Type alphanum (50). The thresholds determine the number of inputs evaluated to collect a representative sample for the data point. Assigning the right priority to the configured Field Types, and configuring an appropriate **learningsetting** value for the **fieldFormatPercentThreshold** and **fieldFormatMinThreshold** parameters, is essential

for getting the correct Field Format recommendation. The Field Type with the highest priority, based on the configured thresholds, is matched first against the inputs. If there is a match, this Field Type is suggested without considering the other Field Types. For example, three default Field Types, integer, alphanum, and any will match if all the inputs contain only numbers. However, integer will be recommended since it has the highest priority.

- **Minimum and Maximum length recommendations**—Calculations for the minimum and maximum lengths for the Field Format are done independently of the determination for the Field Type. The field format length calculations are based on the average length of all the observed inputs. Half of this calculated average is suggested as the min value, and twice the value of this average is suggested as the max value. The range for the Minimum Length is 0-65535 and the range for the Maximum Length is 1-65535. The configured value for the minimum length cannot exceed the maximum length.
- **Handling of space character**—The Field Format check counts every space character when checking for the Field Formats length. Leading or trailing spaces are not stripped, and multiple consecutive spaces in the middle of the input string are no longer consolidated to a single space during input processing.

Example to illustrate the Field Format recommendations:

```
Total requests: 100
Number of Req with Field Type:
Int : 22            (22 int values)  -  22%
Alpha : 44                  (44 alpha values)  -  44%
Alphanum: 14            (14 + 44 + 22 = 80 alphanum values) = 80%
noHTML: 10              (80 + 10 = 90 noHTML values) = 90%
any : 10                     (90 + 10 = 100 any values) = 100%

% threshold                              Suggested Field Type
0-22                int
23-44               alpha
45-80               alphanum
81-90               noHTML
91-100              any
<!--NeedCopy-->
```

To view or use learned data by using the command line interface

```
show appfw learningdata <profilename> FieldFormat
rm appfw learningdata <profilename> -fieldFormat <string>  <
   formActionURL>
export appfw learningdata <profilename> FieldFormat
<!--NeedCopy-->
```

To view or use learned data by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the Field Formats entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

   You also have the option to show a summarized view of the learned relaxations by selecting the Field Formats entry in the Learned Rules table and clicking Visualizer to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the log feature with the field formats check

When the log action is enabled, the Field Formats security check violations are logged in the audit log as APPFW_FIELDFORMAT violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Field Formats violations:

- `Shell`
- `tail -f /var/log/ns.log | grep APPFW_FIELDFORMAT`

To access the log messages by using the GUI

The Citrix GUI includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the Field Formats row and click **Logs**. When you access the logs directly from the **Field Formats security** check of the profile, it filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **Citrix ADC** > **System** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the **Syslog Viewer**, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.

- Navigate to **Application Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

  The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To access Field Formats security check violation log messages, filter by selecting APPFW in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the **APPFW_FIELDFORMAT** check box and click the **Apply** button, only log messages pertaining to the Field Formats security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as Module and EventType, appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Example of a Native format log message when the request is not blocked

```
Jun 10 22:32:26 <local0.info> 10.217.31.98 06/10/2015:22:32:26 GMT ns
    0-PPE-0 :
default APPFW APPFW_FIELDFORMAT 97 0 :  10.217.253.62 562-PPE0
x1MV+YnNGzQFM3Bsy2wti4bhXio0001 pr_ffc http://aaron.stratum8.net/FFC/
    login_post.php
Field format check failed for field passwd="65568888sz-*_" <not blocked
    >
Example of a CEF format log message when the request is blocked
Jun 11 00:03:51 <local0.info> 10.217.31.98
CEF:0|Citrix|Citrix ADC|NS11.0|APPFW|APPFW_FIELDFORMAT|6|src
    =10.217.253.62 spt=27076
method=POST requet=http://aaron.stratum8.net/FFC/maxlen_post.php msg=
    Field format check
failed for field text_area="" cn1=108 cn2=644 cs1=pr_ffc cs2=PPE0
cs3=GaUROfl1Nx1jJTvja5twH5BBqI0000 cs4=ALERT cs5=2015 act=blocked
<!--NeedCopy-->
```

Statistics for the field formats violations

When the stats action is enabled, the corresponding counter for the Field Formats check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 Field Format violations increments the stats counter by one, because the page is blocked as soon as the first Field Formats violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by 3, because each

Field Formats violation generates a separate log message.

To display Field Formats statistics by using command line

At the command prompt, type:

```
sh appfw stats
```

To display stats for a specific profile, use the following command:

```
stat appfw profile <profile name>
```

To display Field Formats statistics by using GUI

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the Statistics Link.
3. Use the scroll bar to view the statistics about Field Formats violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Deployment Tip

- Enable Field format actions log, learn and stats.
- After running a representative sample of the traffic to your application, review the learned recommendations.
- If a Field Type is recommended by most of the learned rules, configure that Field Type as the Default Field Type. For minimum and maximum lengths, use the widest range suggested by these rules.
- Deploy rules for other fields for which different Field Types or different minimum/maximum lengths are better suited.
- Enable blocking and disable learning.
- Monitor stats and logs. If a significant number of violations are still being triggered, you might want to review the log messages to confirm that the violations represent malicious requests that must have been blocked. If valid requests are being flagged as violations, you can either edit the configured Field Format rule to further relax it or enable learning again to get recommendations based on the new data points.

> **Note:** You can fine tune your configuration by getting new learning recommendations.

Highlights

Note the following points about the Field Format security check:

- **Protection**—By configuring optimal field format rules, you can protect against many attacks. For example, if you specify that a field can only have integers, hackers will not be able to launch SQL Injection or cross-site scripting attacks by using this field, because the inputs required to launch such attacks will not meet the configured field format requirement.

- **Performance**—You can limit the minimum and the maximum allowed length for the inputs in the field format rules. This can prevent a malicious user from entering excessively large input strings in an attempt to add processing overhead to the server, or worse, cause the server to dump core because of stack overflow. By limiting the input size, you can shorten the time required for processing legitimate requests.
- **Configuring Field Formats**—You must enable one of the actions (block, log, stats, learn) to engage the field Format protection. You can also specify the Field format rules to identify the allowed inputs in your form fields.
- **Selecting Character Maps vs. Field Types**—Both Character Maps and Field Types use regular expressions. However, a Character Map provides a more specific expression by narrowing down the list of allowed characters. For example, for an input such as janedoe@citrix.com, the learning engine might recommend the Field Type nohtml but the Character Map [.@-Za-z] might be more specific, because it narrows down the allowed set of non-alpha characters. The Character Map option allows, in addition to alpha characters, only two non-alpha characters: period (.) and at (@).
- **Ongoing Learning**—The Web App Firewall monitors and takes into account all the incoming data (violations as well as allowed inputs) to build a learning table for recommending rules. The rules are revised and updated as new incoming data arrives. New field format rules are suggested for a field even if it already has a bound field format rule. If the configured Field Formats are too restrictive and are blocking the valid requests, you can deploy a more relaxed Field Format. Similarly, if the current Field Formats are too generic, you can further refine and tighten the security by deploying a more restrictive Field Format.
- **Overwriting Rules**—If a rule has already been deployed for a field/URL combination, the GUI allows the user to update the field format. A dialog box asks for confirmation to replace the existing rule. If you are using the command line interface, you have to explicitly unbind the previous binding and then bind the new rule.
- **Multiple match**—If multiple field formats match a given field name and its action URL, the Web App Firewall arbitrarily selects one of them to apply.
- **Buffer boundary**—If a field value extends across multiple streaming buffers, and the format for these two parts of the field value is different, a field format corresponding to “any” is sent to the learn database.
- **Field Format vs. Field Consistency Check**—Both the Field Format check and the Field Consistency check are form-based protection checks. The Field Formats check provides a different type of protection than does the Form Field Consistency check. The Form Field Consistency check verifies that the structure of the web forms returned by users is intact, that data format restrictions configured in the HTML are respected, and that data in hidden fields has not been modified. It can do this without any specific knowledge about your web forms other than what it derives from the web form itself. The Field Formats check verifies that the data in each form field matches the specific formatting restrictions that you configured manually, or that the learn-

ing feature generated and you approved. In other words, the Form Field Consistency check enforces general web form security, while the Field Formats check enforces the specific rules for the allowed inputs for your web forms.

Form field consistency check

September 14, 2021

The Form Field Consistency check examines the web forms returned by users of your website, and verifies that web forms were not modified inappropriately by the client. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The Form Field Consistency check prevents clients from making unauthorized changes to the structure of the web forms on your website when they fill out and submit a form. It also ensures that the data a user submits meets the HTML restrictions for length and type, and that data in hidden fields is not modified. This prevents an attacker from tampering with a web form and using the modified form to gain unauthorized access to website, redirect the output of a contact form that uses an insecure script and thereby send unsolicited bulk email, or exploit a vulnerability in your web server software to gain control of the web server or the underlying operating system. Web forms are a weak link on many websites and attract a wide range of attacks.

The Form Field Consistency check verifies all of the following:

- If a field is sent to the user, the check ensures that it is returned by the user.
- The check enforces HTML field lengths and types.

  > **Note:**
  >
  > - The Form Field Consistency check enforces HTML restrictions on data type and length but does not otherwise validate the data in web forms. You can use the Field Formats check to set up rules that validate data returned in specific form fields on your web forms.
  > - The Form Field Consistency protection inserts a hidden field “as_fid” in the response forms which are sent to the client. The same hidden field will be stripped by ADC when the client submits the form. If there is any client-side javascript doing checksum computation on the form fields and verifying the same checksum at the backend may cause application breakage. In this scenario, It is recommended to relax the application firewall form field consistency hidden field “as_fid” from client-side javascript checksum computation.

- If your web server does not send a field to the user, the check does not allow the user to add that field and return data in it.

- If a field is a read-only or hidden field, the check verifies that the data has not changed.
- If a field is a list box or radio button field, the check verifies that the data in the response corresponds to one of the values in that field.

If a web form returned by a user violates one or more of the Form Field consistency checks, and you have not configured the Web App Firewall to allow that web form to violate the Form Field Consistency checks, the request is blocked.

If you use the wizard or the GUI, in the Modify Form Field Consistency Check dialog box, on the General tab you can enable or disable the Block, Log, Learn, and Statistics actions.

You also configure Sessionless Field Consistency in the General tab. If Sessionless Field Consistency is enabled, the Web App Firewall checks only the web form structure, dispensing with those parts of the Form Field Consistency check that depend upon maintaining session information. This can speed the Form Field Consistency check with little security penalty for websites that use many forms. To use Sessionless Field Consistency on all web forms, select On. To use it only for forms submitted with the HTTP POST method, select postOnly

If you use the command-line interface, you can enter the following command to configure the Form Field Consistency Check:

- `set appfw profile <name> -fieldConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`

To specify relaxations for the Form Field Consistency check, you must use the GUI. On the Checks tab of the Modify Form Field Consistency Check dialog box, click Add to open the Add Form Field Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Form Field Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation, as described in Manual Configuration By Using the GUI.

Following are examples of Form Field Consistency check relaxations:

Form Field Names:

- Choose form fields with the name UserType:

```
^UserType$
<!--NeedCopy-->
```

- Choose form fields with names that begin with UserType_ and are followed by a string that begins with a letter or number and consists of from one to twenty-one letters, numbers, or the apostrophe or hyphen symbol:

```
^UserType_[0-9A-Za-z][0-9A-Za-z'-]{
0,20 }
$
<!--NeedCopy-->
```


- Choose form fields with names that begin with Turkish-UserType_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

```
^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$
<!--NeedCopy-->
```

> **Note:**
>
> See PCRE Character Encoding Format for a complete description of supported special characters and how to encode them properly.

- Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

```
^[0-9A-Za-z]*Num[0-9A-Za-z]*$
<!--NeedCopy-->
```

Form Field Action URLs:

- Choose URLs beginning with `http://www.example.com/search.pl?` and containing any string after the query except for a new query:

```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- Choose URLs that begin with `http://www.example-español.com` and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- Choose all URLs that contain the string /search.cgi?:

```
^[^?<>]*/search[.]cgi\?[^?<>]*$
<!--NeedCopy-->
```


> **Caution:**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

CSRF form tagging check

September 14, 2021

The Cross Site Request Forgery (CSRF) Form Tagging check tags each web form sent by a protected website to users with a unique and unpredictable FormID, and then examines the web forms returned by users to ensure that the supplied FormID is correct. This check protects against cross-site request forgery attacks. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The CSRF Form Tagging check prevents attackers from using their own web forms to send high volume form responses with data to your protected websites. This check requires relatively little CPU processing capacity compared to certain other security checks that analyze web forms in depth. It is therefore able to handle high volume attacks without seriously degrading the performance of the protected website or the Web App Firewall itself.

Before you enable the CSRF Form Tagging check, you must be aware of the following:

- You need to enable form tagging. The CSRF check depends on form tagging and does not work without it.
- You must disable the Citrix ADC Integrated Caching feature for all web pages containing forms that are protected by that profile. The Integrated Caching feature and CSRF form tagging are not compatible.
- You must consider enabling Referer checking. Referer checking is part of the Start URL check, but it prevents cross-site request forgeries, not Start URL violations. Referer checking also puts less load on the CPU than does the CSRF Form Tagging check. If a request violates Referer checking, it is immediately blocked, so the CSRF Form Tagging check is not invoked.
- The CSRF Form Tagging check does not work with web forms that use different domains in the form-origin URL and form-action URL. For example, CSRF Form Tagging cannot protect a web form with a form-origin URL of `http://www.example.com` and a form action URL of `http://www.example.org/form`.pl, because example.com and example.org are different domains.

If you use the wizard or the GUI, in the Modify CSRF Form Tagging Check dialog box, on the General tab you can enable or disable the Block, Log, Learn and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the CSRF Form Tagging Check:

- `set appfw profile <name> -CSRFtagAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

To specify relaxations for the CSRF Form Tagging check, you must use the GUI. On the Checks tab of the Modify CSRF Form Tagging Check dialog box, click Add to open the Add CSRF Form Tagging Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify CSRF Form Tagging Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

An alert is generated when you set the Citrix Web App Firewall session limit to a value of 0 or lower, because such a setting affects advanced protection check functionality that requires a properly functioning Web App Firewall session.

Following are examples of CSRF Form Tagging check relaxations:

Note: The following expressions are URL expressions that can be used in both the Form Origin URL and Form Action URL roles.

- Choose URLs beginning with `http://www.example.com/search.pl?` and containing any string after the query, except for a new query:

```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- Choose URLs that begin with `http://www.example-español.com` and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- Choose all URLs that contain the string /search.cgi?:

```
^[^?<>]*/search[.]cgi\?[^?<>]*$
<!--NeedCopy-->
```


Important

Regular expressions are powerful. If you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the check would otherwise have blocked.

Tip

When enableValidate referrer header is enabled under the Start URL Action, ensure that the Referrer Header URL is added to StartURL as well.

Note

When Citrix ADC reaches the appfw_session_limit and CSRF checks are enabled, the web application freezes.

To prevent web application freeze, decrease the session timeout and increase the session limit by using the following commands:

From CLI: > set appfw settings –sessiontimeout 300
From shell: root@ns# nsapimgr_wr.sh -s appfw_session_limit=200000

Logging and generating SNMP alarms when appfw_session_limit is reached assists you in troubleshooting and debugging issues.

Managing CSRF form tagging check relaxations

September 14, 2021

You configure an exception (or relaxation) to the CSRF Form Tagging security check in the Add Cross-Site Request Forgery Tagging Check Relaxation dialog box or the Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box.

To configure a CSRF form tagging check relaxation by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the **Profiles** pane, select the profile you want to configure, and then click **Open**.
3. In the **Configure Web App Firewall Profile** dialog box, click the **Security Checks** tab. The **Security Checks** tab contains the list of Web App Firewall security checks.
4. To add or modify a CSRF relaxation, do one of the following:

- To add a new relaxation, click Add.
- To modify an existing relaxation, select the relaxation that you want to modify, and then click **Open**.

The **Add Cross-Site Request Forgery Tagging Check Relaxation** or **Modify Cross-Site Request Forgery Tagging Check Relaxation** dialog box is displayed. Except for the title, these dialog boxes are identical.

5. Fill in the dialog box as described below.
   - **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
   - **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the form.
   - **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the form is delivered.
   - **Comments**—In the text area, type a comment. Optional.

   > **Note:**
   >
   > For any element that requires a regular expression, you can type the regular expression, use the **Regex Tokens** menu to insert regular expression elements and symbols directly into the text box, or click **Regex Editor** to open the **Add Regular Expression** dialog box, and use it to construct the expression.

6. Click **OK**. The **Add Cross-Site Request Forgery Tagging Check Relaxation** or **Modify Cross-Site Request Forgery Tagging Check Relaxation** dialog box closes and you return to the **Modify Cross-Site Request Forgery Tagging Check** dialog box.
7. To remove a relaxation or rule, select it, and then click **Remove**.
8. To enable a relaxation or rule, select it, and then click **Enable**.
9. To disable a relaxation or rule, select it, and then click **Disable**.
10. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click **Visualizer**, and use the display tools.
11. To review and configure learned rules for the CSRF check, click **Learning** and perform the steps in To configure and use the Learning feature.
12. Click **OK**.

URL protection checks

September 14, 2021

The URL Protection checks examine request URLs to prevent attackers from aggressively attempting to access multiple URLs (forceful browsing) or using a URL to trigger a known security vulnerability in web server software or website scripts.

Start URL check

September 14, 2021

The Start URL check examines the URLs in incoming requests and blocks the connection attempt if the URL does not meet the specified criteria. To meet the criteria, the URL must match an entry in the Start URL list, unless the Enforce URL Closure parameter is enabled. If you enable this parameter, a user who clicks a link on your website is connected to the target of that link.

The primary purpose of the Start URL check is to prevent repeated attempts to access random URLs on a website, (forceful browsing) through bookmarks, external links, or jumping to pages by manually typing in the URLs to skip the pages required to reach that part of the website. Forceful browsing can be used to trigger a buffer overflow, find content that users were not intended to access directly, or find a back door into secure areas of your Web server. The Web App Firewall enforces a website's given traversal or logic path by allowing access to only the URL's that are configured as start URLs.

If you use the wizard or the GUI, in the Modify Start URL Check dialog box, on the General tab you can enable or disable Block, Log, Statistics, Learn actions, and the following parameters:

- **Enforce URL Closure.** Allow users to access any web page on your website by clicking a hyperlink on any other page on your website. Users can navigate to any page on your website that can be reached from the home page or any designated start page by clicking hyperlinks.
  Note: The URL closure feature allows any query string to be appended to and sent with the action URL of a web form submitted by using the HTTP GET method. If your protected websites use forms to access an SQL database, make sure that you have the SQL injection check enabled and properly configured.
- **Sessionless URL Closure.** From the client's point of view, this type of URL closure functions in exactly the same way as standard, session-aware URL Closure, but uses a token embedded in the URL instead of a cookie to track the user's activity, which consumes considerably fewer resources. When sessionless URL closure is enabled, the Web App Firewall appends a "as_url_id" tag to all the URL's that are in URL closure.
  Note: When enabling sessionless (Sessionless URL Closure), you must also enable regular URL

closure (
Enforce URL Closure) or sessionless URL closure does not work.

- **Validate Referer Header.** Verify that the Referer header in a request that contains web form data from your protected website instead of another website. This action verifies that your website, not an outside attacker, is the source of the web form. Doing so protects against cross-site request forgeries (CSRF) without requiring form tagging, which is more CPU-intensive than header checks. The Web App Firewall can handle the HTTP Referer header in one of the following four ways, depending on which option you select in the drop-down list:
  - **Off**—Do not validate the Referer header.
  - **If-Present**—Validate the Referer header if a Referer header exists. If an invalid Referer header is found, the request generates a referer-header violation. If no Referer header exists, the request does not generate a referer-header violation. This option enables the Web App Firewall to perform Referer header validation on requests that contain a Referer header, but not block requests from users whose browsers do not set the Referer header or who use web proxies or filters that remove that header.
  - **Always Except Start URLs**—Always validate the Referer header. If there is no Referer header and the requested URL is not exempted by the startURL relaxation rule, the request generates a referer-header violation. If the Referer header is present but it is invalid, the request generates a referer-header violation.
  - **Always Except First Request**—Always validate the referer header. If there is no referer header, only the URL that is accessed first is allowed. All other URL's are blocked without a valid referer header. If the Referer header is present but it is invalid, the request generates a referer-header violation.

One Start URL setting, **Exempt Closure URLs from Security Checks**, is not configured in the Modify Start URL Check dialog box, but is configured in the Settings tab of the Profile. If enabled, this setting directs the Web App Firewall not to run further form based checks (such as Cross-Site Scripting and SQL Injection inspection) on URLs that meet the URL Closure criteria.

> **Note**
>
> Although the referer header check and Start URL security check share the same action settings, it is possible to violate the referer header check without violating the Start URL check. The difference is visible in the logs, which log referer header check violations separately from Start URL check violations.

The Referer header settings (OFF, if-Present, AlwaysExceptStartURLs, and AlwaysExceptFirstRequest) are arranged in order of least restrictive to most restrictive and work as follows:

OFF:

- Referer Header Not checked.

If-Present:

- Request has no referer header -> Request is allowed.
- Request has referer header and the referer URL is in URL closure -> Request is allowed.
- Request has referer header and the referer URL is **not** in URL closure -> Request is blocked.

AlwaysExceptStartURLs:

- Request has no referer header and the request URL is a start URL -> Request is allowed.
- Request has no referer header and the request URL is not a start URL ->Request is blocked.
- Request has referer header and the referer URL is in URL closure -> Request is allowed.
- Request has referer header and the referer URL is **not** in URL closure -> Request is blocked.

AlwaysExceptFirstRequest:

- Request has no referer header and is the first request URL of the session -> Request is allowed.
- Request has no referer header and is **not** the first request URL of the session -> Request is blocked.
- Request has referer header and is either the first request URL of the session or is in URL closure -> Request is allowed.
- Request has referer header and is neither the first request URL of the session nor is in URL closure -> Request is blocked.

If you use the command-line interface, you can enter the following commands to configure the Start URL Check:

- `set appfw profile <name> -startURLAction [block] [learn] [log] [stats] [none]`
- `set appfw profile <name> -startURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -sessionlessURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([ON] | [OFF)`
- `set appfw profile <name> -RefererHeaderCheck ([OFF] | [if-present] | [AlwaysExceptStartURLs] | [AlwaysExceptFirstRequest])`

To specify relaxations for the Start URL check, you must use the GUI. On the Checks tab of the Modify Start URL Check dialog box, click Add to open the Add Start URL Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Start URL Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Start URL check relaxations:

- Allow users to access the home page at www.example.com:

```
1  ^http://www[.]example[.]com$
2  <!--NeedCopy-->
```


- Allow users to access all static HTML (.htm and .html), server-parsed HTML (.htp and .shtml), PHP (.php), and Microsoft ASP (.asp) format web pages at www.example.com:

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- Allow users to access web pages with pathnames or file names that contain non-ASCII characters at www.example-español.com:

```
^http://www[.]example-espaxC3xB1ol[.]com/(([0-9A-Za-z]|x[0-9A-Fa-
   f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*/)*
([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f
   ][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

Note: In the above expression, each character class has been grouped with the string x[0-9A-Fa-f][0-9A-Fa-f], which matches all properly-constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash () is an escaped backslash, which tells the Web App Firewall to interpret it as a literal backslash. If you included only one backslash, the Web App Firewall would instead interpret the following left square bracket ([) as a literal character instead of the opening of a character class, which would break the expression.

- Allow users to access all GIF (.png), JPEG (.jpg and .jpeg), and PNG (.png) format graphics at www.example.com:

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$
<!--NeedCopy-->
```

- Allow users to access CGI (.cgi) and PERL (.pl) scripts, but only in the CGI-BIN directory:

```
^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_
   .-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- Allow users to access Microsoft Office and other document files in the docsarchive directory:

```
^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_
   -.]*[.](doc|xls|pdf|ppt)$
<!--NeedCopy-->
```


> **Note**
>
> By default, all Web App Firewall URLs are considered to be regular expressions.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions that you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (
.*) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the Start URL check would otherwise have blocked.

> **Tip**
>
> You can add the *-and-* to the allowed list of SQL keywords for URL naming scheme. For example, example https://FQDN/bread-and-butter.

Deny URL check

September 14, 2021

The Deny URL check examines and blocks connections to URLs that are commonly accessed by hackers and malicious code. This check contains a list of URLs that are common targets of hackers or malicious code and that rarely if ever appear in legitimate requests. You can also add URLs or URL patterns to the list. The Deny URL check prevents attacks against various security weaknesses known to exist in web server software or on many websites.

The Deny URL check takes priority over the Start URL check, and thus denies malicious connection attempts even when a Start URL relaxation would normally allow a request to proceed.

In the Modify Deny URL Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the Deny URL Check:

- `set appfw profile <name> -denyURLAction [**block**] [**log**] [**stats**] [**none**]`

To create and configure your own deny URLs, you must use the GUI. On the Checks tab of the Modify Deny URL Check dialog box, click Add to open the Add Deny URL dialog box, or select an existing user-defined deny URL and click Open to open the Modify Deny URLdialog box. Either dialog box provides the same options for creating and configuring a deny URL.

Following are examples of Deny URL expressions:

- Do not allow users to access the image server at images.example.com directly:

```
^http://images[.]example[.]com$
<!--NeedCopy-->
```

- Do not allow users to access CGI (.cgi) or PERL (.pl) scripts directly:

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- Here is the same deny URL, modified to support non-ASCII characters:

```
^http://www[.]example[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f
   ])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|x[0-9A-Fa
   -f][0-9A-Fa-f])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
<!--NeedCopy-->
```

> **Caution:**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL or pattern that you want to block, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block.

XML protection checks

September 14, 2021

The XML Protection checks examine requests for XML-based attacks of all types.

> **Caution:**
>
> The XML security checks apply only to content that is sent with an HTTP content-type header of text/xml. If the content-type header is missing, or is set to a different value, all XML security checks are bypassed. If you plan to protect XML or Web 2.0 web applications, the webmasters of each web server that hosts those applications must ensure that the proper HTTP content-type header is sent.

XML format check

September 14, 2021

The XML Format check examines the XML format of incoming requests and blocks those requests that are not well formed or that do not meet the criteria in the XML specification for properly-formed XML documents. Some of those criteria are:

- An XML document must contain only properly-encoded Unicode characters that match the Unicode specification.
- No special XML syntax characters—such as < , > and &—can be included in the document except when used in XML markup.
- All begin, end, and empty-element tags must be correctly nested, with none missing or overlapping.
- XML element tags are case-sensitive. All beginning and end tags must match exactly.
- A single root element must contain all the other elements in the XML document.

A document that does not meet the criteria for well-formed XML does not meet the definition of an XML document. Strictly speaking, it is not XML. However, not all XML applications and web services enforce the XML well-formed standard, and not all handle poorly-formed or invalid XML correctly. Inappropriate handling of a poorly-formed XML document can cause security breaches. The purpose of the XML Format check is to prevent a malicious user from using a poorly-formed XML request to breach security on your XML application or web service.

If you use the wizard or the GUI, in the Modify XML Format Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Format Check:

- `set appfw profile <name> -xmlFormatAction [**block**] [**log**] [**stats**] [**none**]`

You cannot configure exceptions to the XML Format check. You can only enable or disable it.

XML Denial-of-Service check

September 14, 2021

The XML Denial of Service (XML DoS or XDoS) check examines incoming XML requests to determine whether they match the characteristics of a denial-of-service (DoS) attack. If there is a match, blocks those requests. The purpose of the XML DoS check is to prevent an attacker from using XML requests to launch a denial-of-service attack on your web server or website.

If you use the wizard or the GUI, in the Modify XML Denial-of-Service Check dialog box, on the **General** tab you can enable or disable the Block, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Denial-of-Service check:

- `set appfw profile <name> -xmlDoSAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

To configure individual XML Denial-of-Service rules, you must use the GUI. On the **Checks** tab of the **Modify XML Denial-of-Service Check** dialog box, select a rule and click **Open** to open the **Modify XML Denial-of-Service** dialog box for that rule. The individual dialog boxes differ for the different rules but are simple. Some only allow you to enable or disable the rule; others allow you to modify a number by typing a new value in a text box.

> **Note:**
>
> The expected behavior of Learning engine for denial-of-service attack is based on the configured action. If the action is set as “Block”, the engine learns the configured bind value +1 and the XML parsing stops when there is a violation. If the configured action is not set as “Block”, the engine learns the actual incoming violation length value.

The individual XML Denial-of-Service rules are:

- Maximum Element Depth. Restrict the maximum number of nested levels in each individual element to 256. If this rule is enabled, and the Web App Firewall detects an XML request with an element that has more than the maximum number of allowed levels, it blocks the request. You can modify the maximum number of levels to any value from one (1) to 65,535.
- Maximum Element Name Length. Restrict the maximum length of each element name to 128 characters. This includes the name within the expanded namespace, which includes the XML path and element name in the following format:

```
{
http://prefix.example.com/path/ }
target_page.xml
<!--NeedCopy-->
```

The user can modify the maximum name length to any value between one (1) character and 65,535.

- Maximum # Elements. Restrict the maximum number of any one type of element per XML document to 65,535. You can modify the maximum number of elements to any value between one (1) and 65,535.
- Maximum # Element Children. Restrict the maximum number of children (including other elements, character information, and comments) each individual element is allowed to have to

65,535. You can modify the maximum number of element children to any value between one (1) and 65,535.

- Maximum # Attributes. Restrict the maximum number of attributes each individual element is allowed to have to 256. You can modify the maximum number of attributes to any value between one (1) and 256.
- Maximum Attribute Name Length. Restrict the maximum length of each attribute name to 128 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.
- Maximum Attribute Value Length. restrict the maximum length of each attribute value to 2048 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.
- Maximum Character Data Length. Restrict the maximum character data length for each element to 65,535. You can modify the length to any value between one (1) and 65,535.
- Maximum File Size. Restrict the size of each file to 20 MB. You can modify the maximum file size to any value.
- Minimum File Size. Require that each file is least 9 bytes in length. You can modify the minimum file size to any positive integer representing various bytes.
- Maximum # Entity Expansions. Limit the number of entity expansions allowed to the specified number. Default: 1024.
- Maximum Entity Expansion Depth. Restrict the maximum number of nested entity expansions to no more than the specified number. Default: 32.
- Maximum # Namespaces. Limit the number of namespace declarations in an XML document to no more than the specified number. Default: 16.
- Maximum Namespace URI Length. Limit the URL length of each namespace declaration to no more than the specified number of characters. Default: 256.
- Block Processing Instructions. Block any special processing instructions included in the request. This rule has no user-modifiable values.
- Block DTD. Block any document type definitions (DTD) included with the request. This rule has no user-modifiable values.
- Block External Entities. Block all references to external entities in the request. This rule has no user-modifiable values.
- SOAP Array Check. Enable or disable the following SOAP array checks:
  - **Maximum SOAP Array Size.** The maximum total size of all SOAP arrays in an XML request before the connection is blocked. You can modify this value. Default: 20000000.

- **Maximum SOAP Array Rank.** The maximum rank or dimensions of any single SOAP array in an XML request before the connection is blocked. You can modify this value. Default: 16.

XML cross-site scripting check

September 14, 2021

The XML Cross-Site Scripting check examines the user requests for possible cross-site scripting attacks in the XML payload. If it finds a possible cross-site scripting attack, it blocks the request.

To prevent misuse of the scripts on your protected web services to breach security on your web services, the XML Cross-Site Scripting check blocks scripts that violate the same origin rule, which states that scripts must not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

The Web App Firewall offers various action options for implementing XML Cross-Site Scripting protection. You have the option to configure **Block**, **Log**, and **Stats** actions.

The Web App Firewall XML cross-site scripting check is performed on the payload of the incoming requests and attack strings are identified even if they are spread over multiple lines. The check looks for cross-site scripting attack strings in the **element** and the **attribute** values. You can apply relaxations to bypass security check inspection under specified conditions. The logs and statistics can help you identify needed relaxations.

The CDATA section of the XML payload might be an attractive area of focus for the hackers because the scripts are not executable outside the CDATA section. A CDATA section is used for content that is to be treated entirely as character data. HTML mark up tag delimiters **<**, **>**, and **/>** will not cause the parser to interpret the code as HTML elements. The following example shows a CDATA Section with cross-site scripting attack string:

```
1     <![CDATA[rn
2     <script language="Javascript" type="text/javascript">alert ("Got
          you")</script>rn
3     ]]>
4 <!--NeedCopy-->
```


Action Options

An action is applied when the XML Cross-Site Scripting check detects an cross-site scripting attack in the request. The following options are available for optimizing XML Cross-Site Scripting protection for your application:

- **Block**—Block action is triggered if the cross-site scripting tags are detected in the request.
- **Log**—Generate log messages indicating the actions taken by the XML Cross-Site Scripting check. If block is disabled, a separate log message is generated for each location (ELEMENT, ATTRIBUTE) in which the cross-site scripting violation is detected. However, only one message is generated when the request is blocked. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
- **Stats**—Gather statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

Relaxation Rules

If your application requires you to bypass the Cross-Site Scripting check for a specific ELEMENT or ATTRIBUTE in the XML payload, you can configure a relaxation rule. The XML Cross-Site Scripting check relaxation rules have the following parameters:

- **Name**—You can use literal strings or regular expressions to configure the name of the ELEMENT or the Attribute. The following expression exempts all ELEMENTS beginning with the string name_ followed by a string of uppercase or lowercase letters, or numbers, that is at least two and no more than fifteen characters long:

```
^name_[0-9A-Za-z]{ 2,15 } $
```

> **Note**
>
> The names are case sensitive. Duplicate entries are not allowed, but you can use capitalization of the names and differences in location to create similar entries. For example, each of the following relaxation rules is unique:
>
> 1. `XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED`
> 2. `XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ELEMENT State: ENABLED`
> 3. `XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ELEMENT State: ENABLED`


```
4. XMLcross-site scripting: abc IsRegex: NOTREGEX
   Location: ATTRIBUTE State: ENABLED
```

- **Location**—You can specify the Location of the Cross-site Scripting Check exception in your XML payload. The option ELEMENT is selected by default. You can change it to ATTRIBUTE.
- **Comment**—This is an optional field. You can use up to a 255 character string to describe the purpose of this relaxation Rule.

> **Warning**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML Cross-Site Scripting check would otherwise have blocked.

Using the Command Line to Configure the XML Cross-Site Scripting check

To configure XML Cross-Site Scripting check actions and other parameters by using the command line

If you use the command-line interface, you can enter the following commands to configure the XML Cross-Site Scripting Check:

```
> set appfw profile <name> -XMLcross-site scriptingAction (([block] [log] [
stats])| [none])
```

To configure a XML Cross-Site Scripting check relaxation rule by using the command line

You can add relaxation rules to bypass inspection of cross-site scripting script attack inspection in a specific location. Use the bind or unbind command to add or delete the relaxation rule binding, as follows:

```
> bind appfw profile <name> -XMLcross-site scripting <string> [isRegex (
REGEX | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )] -comment <string> [-
state ( ENABLED | DISABLED )]
```

```
> unbind appfw profile <name> -XMLcross-site scripting <String>
```

Example:

```
> bind appfw profile test_pr -XMLcross-site scripting ABC
```

After executing the above command, the following relaxation rule is configured. The rule is enabled, the name is treated as a literal (NOTREGEX), and ELEMENT is selected as the default location:

```
1)      XMLcross-site scripting:  ABC            IsRegex:  NOTREGEX
2
```



```
3         Location:  ELEMENT       State:  ENABLED
4
5 `> unbind appfw profile test_pr -XMLcross-site scripting abc`
6
7 ERROR: No such XMLcross-site scripting check
8
9 `> unbind appfw profile test_pr -XMLcross-site scripting ABC`
10
11  Done
12 <!--NeedCopy-->
```

Using the GUI to configure the XML Cross-Site scripting check

In the GUI, you can configure the XML Cross-Site scripting check in the pane for the profile associated with your application.

To configure or modify the XML Cross-Site Scripting check by using the GUI

1. Navigate to **Web App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a) If you just want to enable or disable **Block**, **Log**, and **Stats** actions for the **XML Cross-Site Scripting check**, you can select or clear check boxes in the table, click **OK**, and then click Save and Close to close the Security Check pane.

b) You can double click **XML Cross-Site Scripting**, or select the row and click **Action Settings**, to display the action options. After changing any of the action settings, click **OK** to save the changes and return to the Security Checks table.

You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure a XML Cross-Site Scripting relaxation rule by using the GUI

1. Navigate to **Web App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **XML Cross-Site Scripting** entry, or select it and click **Edit**.
4. In the **XML Cross-Site Scripting Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

To manage XML Cross-Site Scripting relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **XML Cross-Site Scripting** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the Cross-Site Scripting patterns by using the GUI

You can use the GUI to view or customize the default list of cross-site scripting allowed attributes or allowed tags. You can also view or customize the default list of cross-site scripting denied Patterns.

The default lists are specified in **Web App Firewall** > **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default cross-site scripting Allowed and Denied list specified in the Default Signatures object will be used by the profile for the Cross-Site Scripting security check processing. The Tags, Attributes, and Patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the Allowed or Denied lists in the new user-defined signature object and use this signature object in the profile that is processing the traffic for which you want to use these customized allowed and denied lists.

For more information about signatures, see http://support.citrix.com/proddocs/topic/ns-security-10-map/appfw-signatures-con.html.

To view default cross-site scripting patterns:

1. Navigate to **Web App Firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/cross-site scripting Patterns**.

The **Manage SQL/cross-site scripting Paths** table shows following three rows pertaining to cross-site scripting :

```
            xss/allowed/attribute

            xss/allowed/tag

            xss/denied/pattern
<!--NeedCopy-->
```

Select a row and click **Manage Elements** to display the corresponding cross-site scripting Elements (Tag, Attribute, Pattern) used by the Web App Firewall **Cross-Site Scripting** check.

To customize cross-site scripting Elements: You can edit the user-defined signature object to customize the allowed Tag, allowed Attributes and denied Patterns. You can add new entries or remove the existing ones.

1. **Navigate to Web App Firewall** > **Signatures**, highlight the target user-defined signature, and click **Edit**. Click **Manage SQL/cross-site scripting Patterns** to display the **Manage SQL/cross-site scripting paths** table.

2. Select the target cross-site scripting row.

a) Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding cross-site scripting element.

b) Click **Remove** to remove the selected row.

> **Warning**
>
> Be very careful when you remove or modify any default cross-site scripting element, or delete the cross-site scripting path to remove the entire row. The signatures, HTML Cross-Site Scripting security check, and XML Cross-Site Scripting security check rely on these Elements for detecting attacks to protect your applications. Customizing the cross-site scripting Elements can make your application vulnerable to Cross-Site Scripting attacks if the required pattern is removed during editing.

Using the log feature with the XML cross-site scripting check

When the log action is enabled, the XML Cross-Site Scripting security check violations are logged in the audit log as **APPFW_XML_cross-site scripting** violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the XML Cross-Site Scripting violations:

```
> **Shell**

> **tail -f /var/log/ns.log | grep APPFW_XML_cross-site scripting**
<!--NeedCopy-->
```

Example of a XML Cross-Site Scripting security check violation log message in Native log format showing <blocked> action

```
Oct  7 01:44:34 <local0.warn> 10.217.31.98 10/07/2015:01:44:34 GMT ns
    0-PPE-1 : default APPFW APPFW_XML_cross-site scripting 1154 0 :
    10.217.253.69 3466-PPE1 - owa_profile http://10.217.31.101/FFC/login
    .html Cross-site script check failed for field script="Bad tag:
    script" <**blocked**>
<!--NeedCopy-->
```

Example of a XML Cross-Site Scripting security check violation log message in CEF log format showing <not blocked> action

```
Oct  7 01:46:52 <local0.warn> 10.217.31.98 CEF:0|Citrix|Citrix ADC|NS11
    .0|APPFW|APPFW_XML_cross-site scripting|4|src=10.217.30.17
```



```
      geolocation=Unknown spt=33141 method=GET request=http://
      10.217.31.101/FFC/login.html msg=Cross-site script check failed for
      field script="Bad tag: script" cn1=1607 cn2=3538 cs1=owa_profile cs2
      =PPE0 cs4=ERROR cs5=2015 act=**not blocked**
<!--NeedCopy-->
```

To access the log messages by using the GUI

The Citrix GUI includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Web App Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **XML Cross-Site Scripting** row and click **Logs**. When you access the logs directly from the XML Cross-Site Scripting check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **Citrix ADC** > **System** > **Auditing**. In the Audit Messages section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Web App Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **XML Cross-Site Scripting** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XML_cross-site scripting** check box and click the **Apply** button, only log messages pertaining to the XML Cross-Site Scripting security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** and so forth. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Statistics for the XML cross-site scripting violations

When the stats action is enabled, the counter for the XML Cross-Site Scripting check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three XML Cross-Site Scripting violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing

the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display XML Cross-Site Scripting check statistics by using the command line

At the command prompt, type:

```
> **sh appfw stats**
```

To display stats for a specific profile, use the following command:

```
> **stat appfw profile** <profile name>
```

To display XML Cross-Site Scripting statistics by using the GUI

1. Navigate to **System** > **Security** > **Web App Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about XML Cross-Site Scripting violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

XML SQL injection check

September 14, 2021

The XML SQL injection check examines the user requests for possible XML SQL Injection attacks. If it finds injected SQL in XML payloads, it blocks the requests.

A XML SQL attack can inject source code into a web application such that it can be interpreted and run as a valid SQL query to perform a database operation with malicious intent. For example, XML SQL attacks can be launched to gain unauthorized access to the contents of a database or to manipulate the stored data. XML SQL Injection attacks are not only common, but can also be very harmful and costly.

Compartmentalizing the privileges of the database users can assist in protecting the database to some extent. All database users must be given only the required privileges to complete their intended tasks, so that they cannot run SQL queries to perform other tasks. For example, a read-only user must not be allowed to write or manipulate data tables. The Web App Firewall XML SQL Injection check inspects all XML requests to provide special defenses against injection of unauthorized SQL code that might break security. If the Web App Firewall detects unauthorized SQL code in any XML request of any user, it can block the request.

The Citrix Web App Firewall inspects the presence of SQL keywords and special characters to identify the XML SQL Injection attack. A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch XML SQL attacks. The Web App Firewall considers three characters, single straight quote (‘), backslash (), and semicolon (;) as special

characters for SQL security check processing. You can add new patterns, and you can edit the default set to customize the XML SQL check inspection.

The Web App Firewall offers various action options for implementing XML SQL Injection protection. You can **Block** the request, **Log** a message in the ns.log file with details regarding the observed violations, and collect **Stats** to keep track of the number of observed attacks.

In addition to actions, there are several parameters that can be configured for XML SQL injection processing. You can check for **SQL wildcard characters**. You can change the XML SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the XML payload. The XML **SQL Comments Handling** parameter gives you an option to specify the type of comments that need to be inspected or exempted during XML SQL Injection detection.

You can deploy relaxations to avoid false positives. The Web App Firewall XML SQL check is performed on the payload of the incoming requests, and attack strings are identified even if they are spread over multiple lines. The check looks for SQL Injection strings in the **element** and the **attribute** values. You can apply relaxations to bypass security check inspection under specified conditions. The logs and statistics can help you identify needed relaxations.

Action options

An action is applied when the XML SQL Injection check detects an SQL Injection attack string in the request. The following actions are available for configuring an optimized XML SQL Injection protection for your application:

Block—If you enable block, the block action is triggered only if the input matches the XML SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the XML SQL injection type, a request is not blocked if it does not contain any key words, even if SQL special characters are detected in the payload. Such a request is blocked if the XML SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.

Log—If you enable the log feature, the XML SQL Injection check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each location (**ELEMENT**, **ATTRIBUTE**) in which the XML SQL violation was detected. However, only one message is generated when the request is blocked. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

Stats—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

XML SQL parameters

In addition to the block, log and stats actions, you can configure the following parameters for XML SQL Injection check:

Check for XML SQL wildcard Characters—Wild card characters can be used to broaden the selections of a structured query language (SQL-SELECT) statement. These wild card operators can be used in conjunction with **LIKE** and **NOT LIKE** operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

```
SELECT * from customer WHERE name like "%D%"
```

The following example combines the operators to find any salary values that have 0 as the second and third character.

```
SELECT * from customer WHERE salary like '_00%
```

Different DBMS vendors have extended the wildcard characters by adding extra operators. The Citrix Web App Firewall can protect against attacks that are launched by injecting these wildcard characters. The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening square bracket ([), and closing square bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures**:

```
1 -   <wildchar type=" LITERAL" >%</wildchar>
2 -   <wildchar type=" LITERAL" >_</wildchar>
3 -   <wildchar type=" LITERAL" >^</wildchar>
4 -   <wildchar type=" LITERAL" >[</wildchar>
5 -   <wildchar type=" LITERAL" >]</wildchar>
6 <!--NeedCopy-->
```

Wildcard characters in an attack can be PCRE, like [^A-F]. The Web App Firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

> **Note**
>
> The XML SQL **wildcard character** check is different from the XML SQL **special character** check. This option must be used with caution to avoid false positives.

Check Request Containing SQL Injection Type—The Web App Firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the

application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

- **SQL Special Character and Keyword**—Both an SQL keyword and an SQL special character must be present in the inspected location to trigger SQL violation. This least restrictive setting is also the default setting.
- **SQL Special Character**—At least one of the special characters must be present in the processed payload string to trigger SQL violation.
- **SQL keyword**—At least one of the specified SQL keywords must be present in the processed payload string to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.
- **SQL Special Character or Keyword**—Either the keyword or the special character string must be present in the payload to trigger the security check violation.

> **Tip**
>
> If you select the SQL Special Character option, the Web App Firewall skips strings that do not contain any special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this option can significantly reduce the load on the Web App Firewall and speed up processing without placing your protected websites at risk.

SQL comments handling—By default, the Web App Firewall parses and checks all comments in XML data for injected SQL commands. Many SQL servers ignore anything in a comment, even if preceded by an SQL special character. For faster processing, if your XML SQL server ignores comments, you can configure the Web App Firewall to skip comments when examining requests for injected SQL. The XML SQL comments handling options are:

- **ANSI**—Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.
- **Nested**—Skip nested SQL comments, which are normally used by Microsoft SQL Server.
- **ANSI/Nested**—Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.
- **Check all Comments**—Check the entire request for injected SQL, without skipping anything. This is the default setting.

> **Tip**
>
> In most cases, you must not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they might indicate an attempt to breach security on that server.

Relaxation rules

If your application requires you to bypass the XML SQL Injection inspection for a specific ELEMENT or ATTRIBUTE in the XML payload, you can configure a relaxation rule. The XML SQL Injection inspection relaxation rules have the following parameters:

- **Name:** You can use literal strings or regular expressions to configure the name of the **ELEMENT** or the **ATTRIBUTE**. The following expression exempts all **ELEMENTS** beginning with the string **PurchaseOrder_** followed by a string of numbers that is at least two and no more than ten characters long:

  Comment: “Exempt XML SQL Check for Purchase Order Elements”

```
    XMLSQLInjection:  "PurchaseOrder_[0-9A-Za-z]{
 2,10 }
 "

    IsRegex:  REGEX          Location:  ELEMENT

    State:  ENABLED
<!--NeedCopy-->
```

Note: The names are case sensitive. Duplicate entries are not allowed, but you can use capitalization of the names and differences in location to create similar entries. For example, each of the following relaxation rules is unique:

```
1)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

2)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

3)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

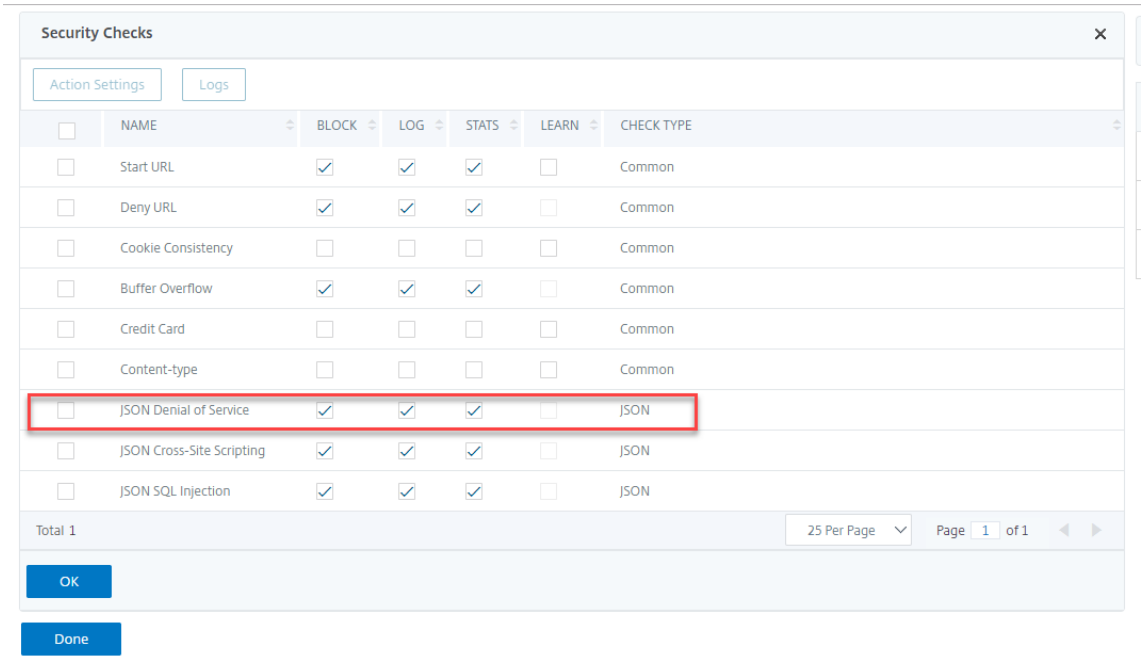
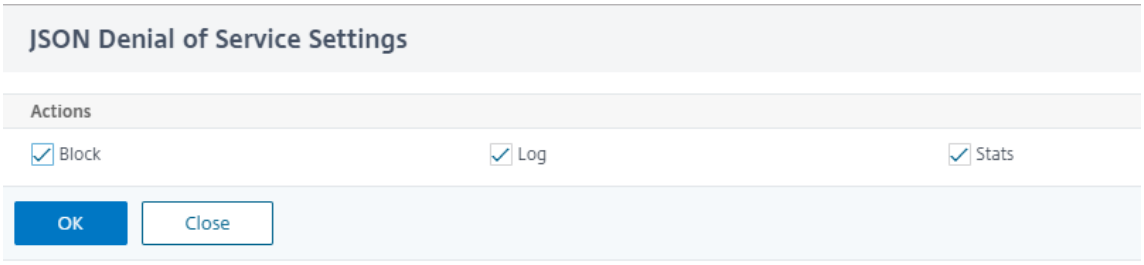
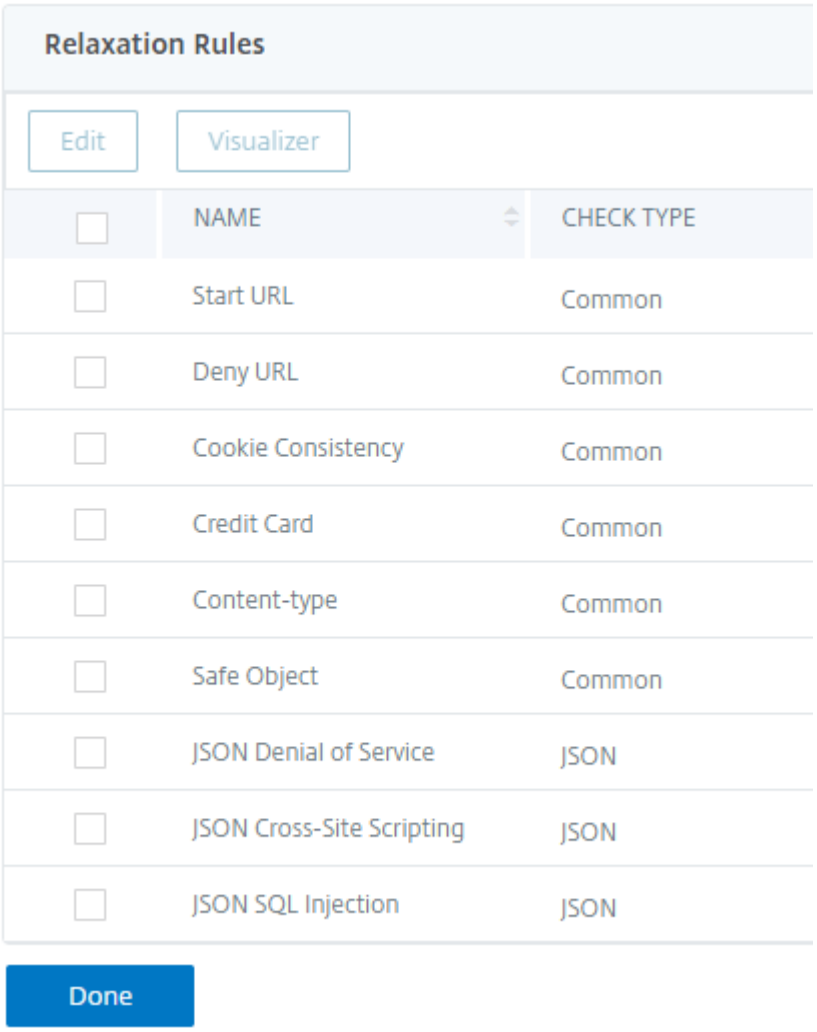
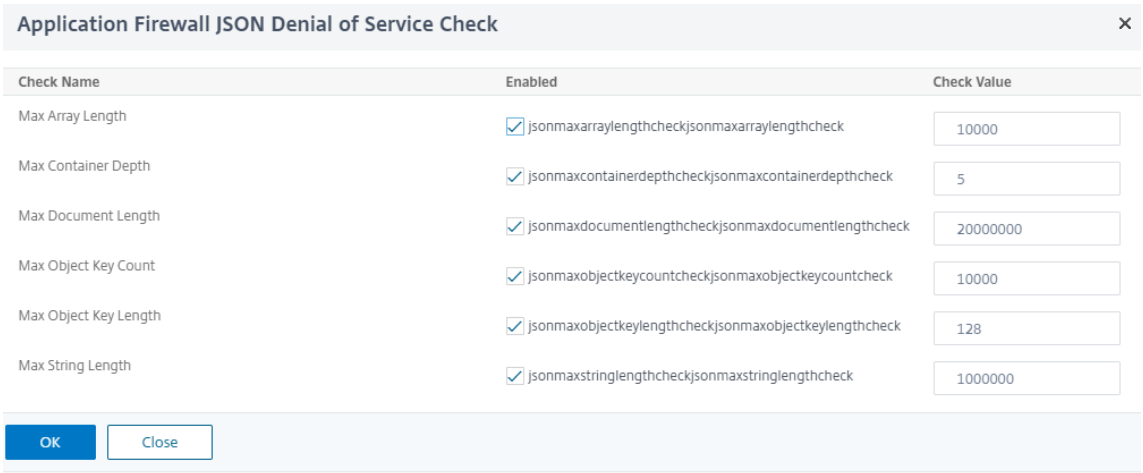
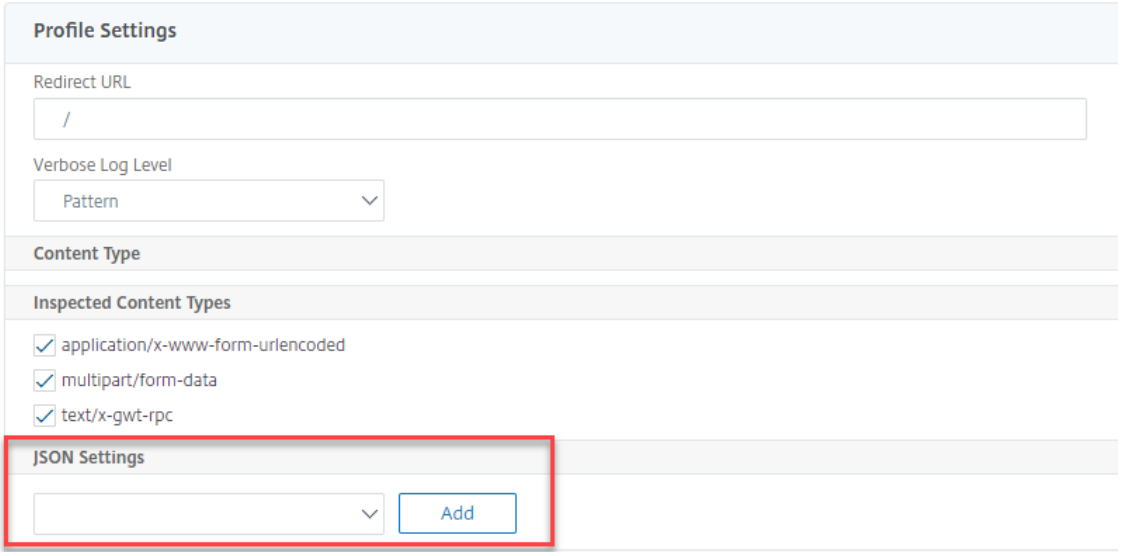
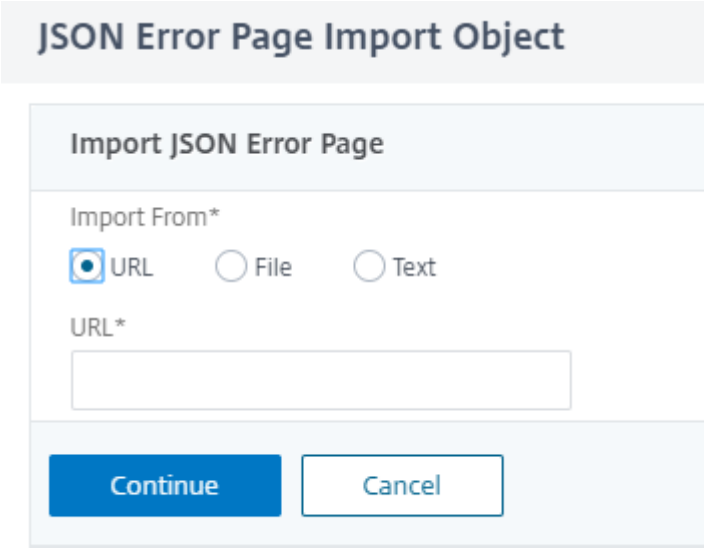
        Location:  ATTRIBUTE     State:  ENABLED

4)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED
<!--NeedCopy-->
```

- **Location**: You can specify the Location of the XML SQL Inspection exception in your XML payload. The option **ELEMENT** is selected by default. You can change it to **ATTRIBUTE**.

- **Comment**: This is an optional field. You can use up to a 255 character string to describe the purpose of this relaxation Rule.

> **Warning**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML SQL Injection inspection would otherwise have blocked.

Using the command line to configure the XML SQL Injection Check

To configure XML SQL Injection actions and other parameters by using the command line:

In the command line interface, you can use either the **set appfw profile** command or the **add appfw profile** command to configure the XML SQL Injection protections. You can enable the block, log, and stats action(s). Select the type of SQL attack pattern (key words, wildcard characters, special strings) you want to detect in the payloads. Use the **unset appfw profile** command to revert the configured settings back to their defaults. Each of the following commands sets only one parameter, but you can include multiple parameters in a single command:

- `set appfw profile <name> **-XMLSQLInjectionAction** (([block] [log] [stats])| [none])`
- `set appfw profile <name> -XMLSQLInjectionCheckSQLWildChars (ON |OFF)`
- `set appfw profile <name> -XMLSQLInjectionType ([SQLKeyword] | [SQLSplChar] | [SQLSplCharANDKeyword] | [SQLSplCharORKeyword])`
- `set appfw profile <name> -XMLSQLInjectionParseComments ([checkall] | [ansi|nested] | [ansinested])`

To configure a SQL Injection relaxation rule by using the command line

Use the bind or unbind command to add or delete relaxation rules, as follows:

```
1 -  bind appfw profile <name> -XMLSQLInjection <string> [isRegex (REGEX
     | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )]  - comment <string>
     [-state ( ENABLED | DISABLED )]
2 -  unbind appfw profile <name> -XMLSQLInjection <String>
3 <!--NeedCopy-->
```

Example:

```
1 > bind appfw profile test_profile -XMLSQLInjection "PurchaseOrder_[0-9A
     -Za-z]{
```



```
2   2,15 }
3   " -isregex REGEX -location ATTRIBUTE
4
5  > unbind appfw profile test_profile  - XMLSQLInjection "PurchaseOrder_
       [0-9A-Za-z]{
6   2,15 }
7   " -location ATTRIBUTE
8  <!--NeedCopy-->
```

Using the GUI to configure the XMLSQL injection security check

In the GUI, you can configure the XML SQL Injection security check in the pane for the profile associated with your application.

To configure or modify the XML SQL Injection check by using the GUI

1. Navigate to **Web App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a. If you just want to enable or disable Block, Log, and Stats actions for XML SQL Injection, you can select or clear check boxes in the table, click OK, and then click Save and Close to close the Security Check pane.

b. If you want to configure additional options for this security check, double click XML SQL Injection, or select the row and click **Action Settings**, to display the following options:

Check for SQL Wildcard Characters—Consider SQL Wildcard characters in the payload to be attack patterns.

Check Request Containing—Type of SQL injection (SQLKeyword, SQLSplChar, SQLSplCharANDKeyword, or SQLSplCharORKeyword) to check.

SQL Comments Handling—Type of comments (Check All Comments, ANSI, Nested, or ANSI/Nested) to check.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save** and **Close** to close the Security Check pane.

To configure a XML SQL Injection relaxation rule by using the GUI

1. Navigate to **Web App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.

3. In the Relaxation Rules table, double-click the **XML SQL Injection** entry, or select it and click **Edit**.
4. In the **XML SQL Injection Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

To manage XML SQL Injection relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **XML SQL Injection** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the SQL Injection patterns by using the GUI:

You can use the GUI to view or customize the SQL patterns.

The default SQL patterns are specified in **Web App Firewall** > **Signatures** > ***Default Signatures**. If you do not bind any signature object to your profile, the default SQL patterns specified in the Default Signatures object will be used by the profile for XML SQL Injection security check processing. The rules and patterns in the Default Signatures object are read-only. You cannot edit or modify them. If you want to modify or change these patterns, create a user-defined signature object by making a copy of the Default Signatures object and changing the SQL patterns. Use the user-defined signature object in the profile that processes the traffic for which you want to use these customized SQL patterns.

For more information, see Signatures.

To view default SQL patterns:

a. Navigate to **Web App Firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/cross-site scripting Patterns**.

The Manage SQL/cross-site scripting Paths table shows following four rows pertaining to SQL Injection:

```
Injection (not_alphanum, SQL)/ Keyword

Injection (not_alphanum, SQL)/ specialstring

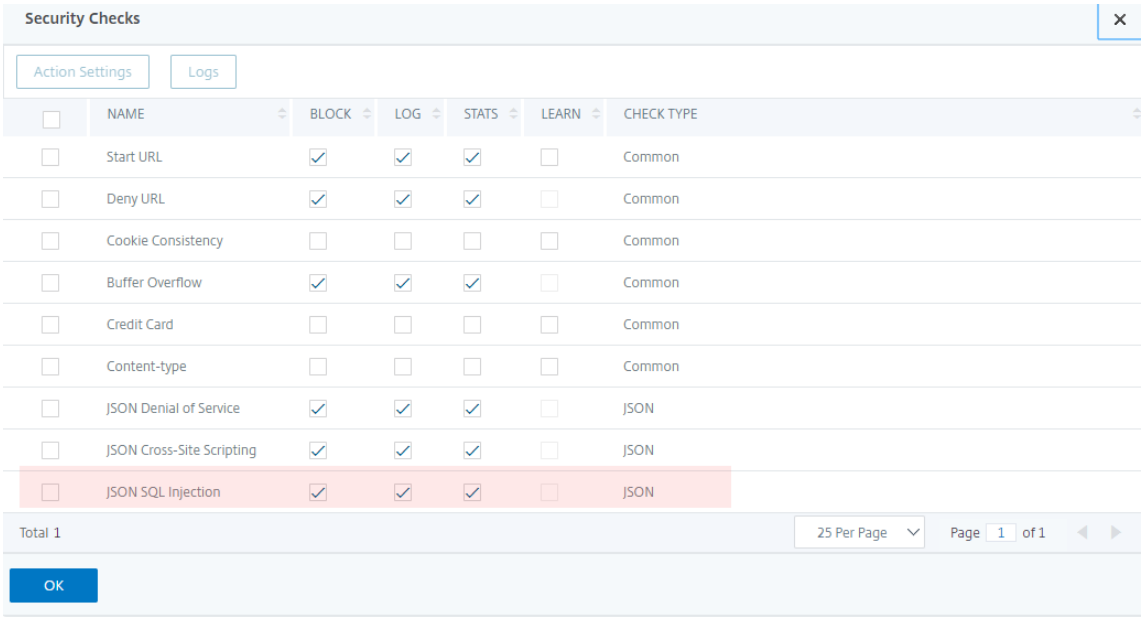
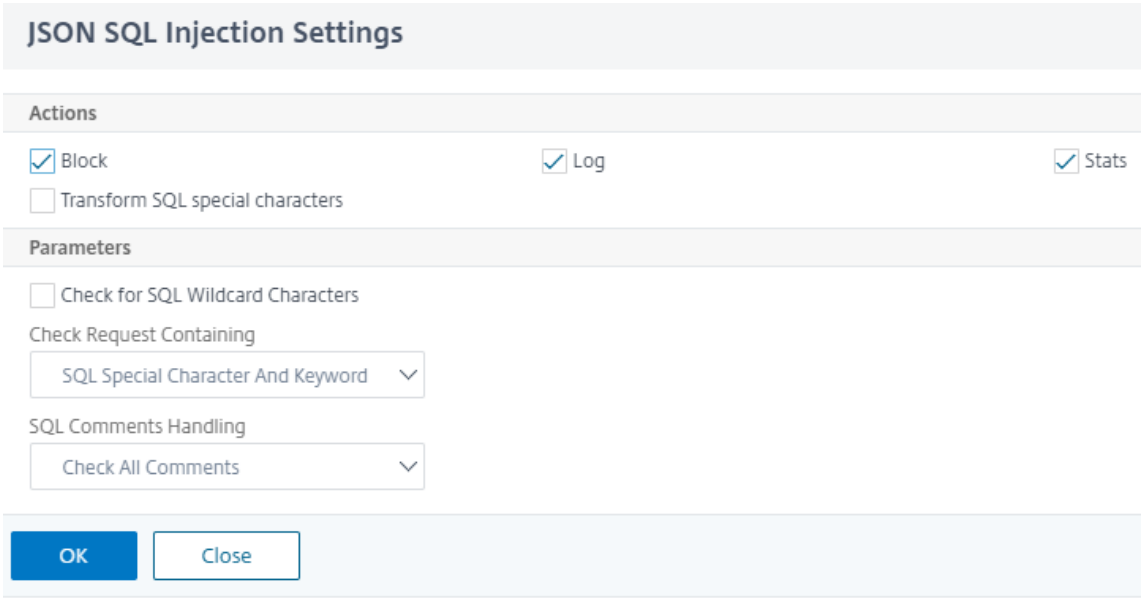
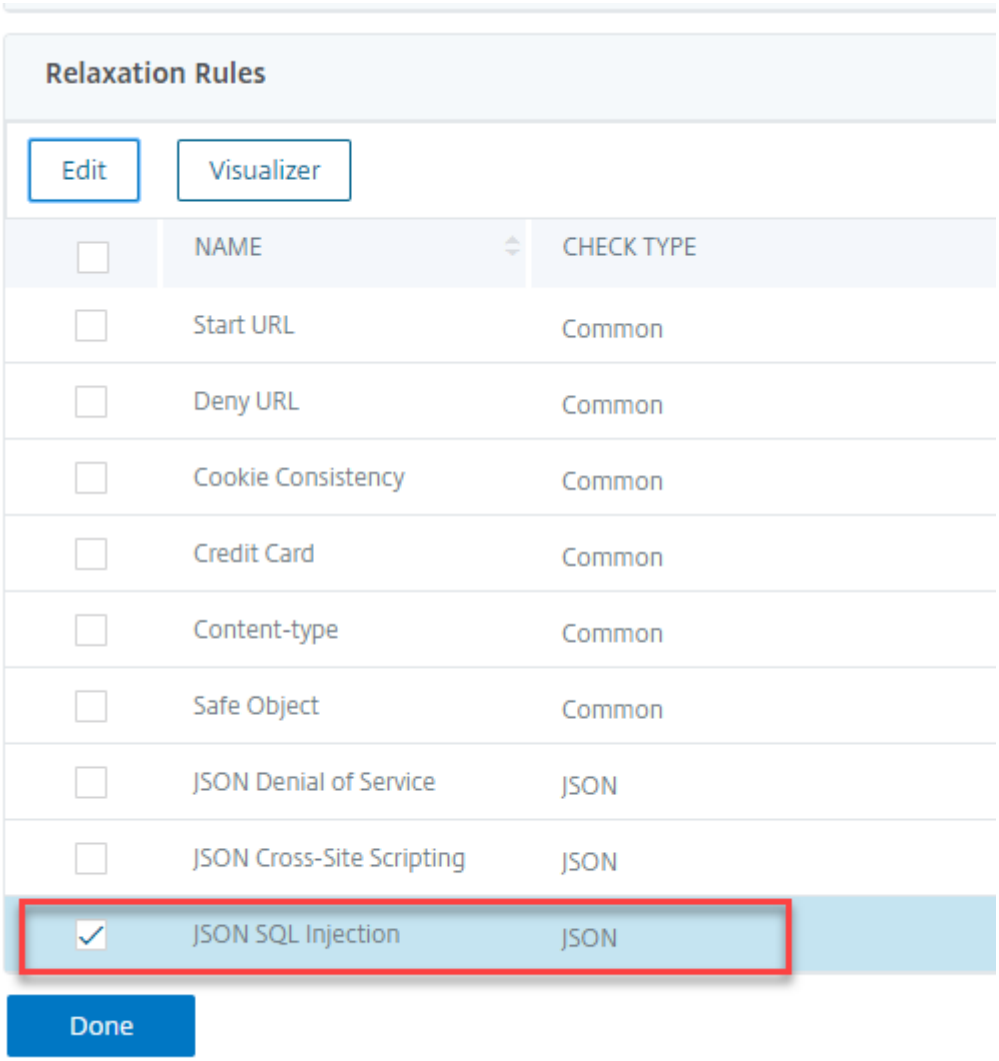
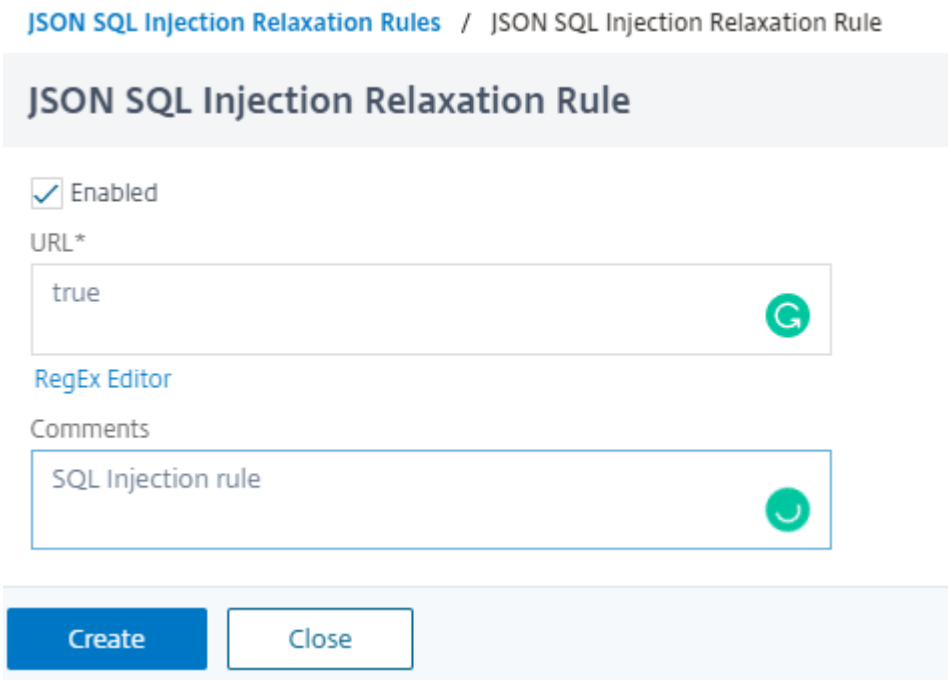
Injection (not_alphanum, SQL)/ transformrules/transform

Injection (not_alphanum, SQL)/ wildchar
<!--NeedCopy-->
```

b. Select a row and click **Manage Elements** to display the corresponding SQL patterns (keywords, special strings, transformation rules or the wildcard characters) used by the Web App Firewall SQL injection check.

To customize SQL Patterns: You can edit a user-defined signature object to customize the SQL key words, special strings, and wildcard characters. You can add new entries or remove the existing ones. You can modify the transformation rules for the SQL special strings.

a. Navigate to **Web App Firewall** > **Signatures**, highlight the target user-defined signature, and click **Edit**. Click **Manage SQL/cross-site scripting Patterns** to display the **Manage SQL/cross-site scripting paths** table.

b. Select the target SQL row.

i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding SQL element.

ii. Click **Remove** to remove the selected row.

> **Warning**
>
> You must be very careful when removing or modifying any default SQL element, or deleting the SQL path to remove the entire row. The signature rules as well as the XML SQL Injection security check rely on these elements for detecting SQL Injection attacks to protect your applications. Customizing the SQL patterns can make your application vulnerable to XML SQL attacks if the required pattern is removed during editing.

Using the log feature with the XML SQL injection check

When the log action is enabled, the **XML SQL Injection** security check violations are logged in the audit log as **APPFW_XML_SQL** violations. The Web App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line:

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the XML Cross-Site Scripting violations:

```
1 > Shell
2
3 > tail -f /var/log/ns.log | grep APPFW_XML_SQL
4 <!--NeedCopy-->
```

To access the log messages by using the GUI

The Citrix GUI includes a useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to **Web App Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **XML SQL Injection** row and click **Logs**. When you access the logs directly from the XML SQL Injection check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.

- You can also access the Syslog Viewer by navigating to **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Web App Firewall** > **Policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the **Syslog Viewer**, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **XML SQL Injection** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XML_SQL** check box and click the **Apply** button, only log messages pertaining to the **XML SQL Injection** security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, and **Client IP** appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Statistics for the XML SQL injection violations

When the stats action is enabled, the counter for the **XML SQL Injection** check is incremented when the Web App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three **XML SQL Injection** violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display XML SQL Injection check statistics by using the command line

At the command prompt, type:

```
> sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> stat appfw profile <profile name>
```

To display XML SQL Injection statistics by using the GUI

1. Navigate to **System** > **Security** > **Web App Firewall**.
2. In the right pane, access the **Statistics** Link.

3. Use the scroll bar to view the statistics about **XML SQL Injection** violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

XML attachment check

September 14, 2021

The XML Attachment check examines incoming requests for malicious attachments, and it blocks those requests that contain attachments that might breach applications security. The purpose of the XML Attachment check is to prevent an attacker from using an XML attachment to breach security on your server.

If you use the wizard or the GUI, in the Modify XML Attachment Check dialog box, on the General tab you can enable or disable the Block, Learn, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Attachment Check:

- `set appfw profile <name> -xmlAttachmentAction [block] [learn] [log] [stats] [none]`

You must configure the other XML Attachment check settings in the GUI. In the `Modify XML Attachment` Check dialog box, on the Checks tab, you can configure the following settings:

- **Maximum Attachment Size.** Allow attachments that are no larger than the maximum attachment size you specify. To enable this option, first select the Enabled check box, and then type the maximum attachment size in bytes in the `Size` text box.
- **Attachment Content Type.** Allow attachments of the specified content type. To enable this option, first select the Enabled check box, and then enter a regular expression that matches the Content-Type attribute of the attachments that you want to allow.
  - You can type the URL expression directly in the text window. If you do so, you can use the `Regex Tokens` menu to enter a number of useful regular expressions at the cursor instead of typing them manually.
  - You can click Regex Editor to open the `Add Regular Expression` dialog box and use it to construct the URL expression.

Web services interoperability check

September 14, 2021

The Web Services Interoperability (WS-I) check examines both requests and responses for adherence to the WS-I standard, and blocks those requests and responses that do not adhere to this standard. The purpose of the WS-I check is to block requests that might not interact with other XML appropriately. An attacker can use inconsistencies in interoperability to launch an attack on your XML application.

If you use the wizard or the GUI, in the Modify Web Services Interoperability Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions.

If you use the command-line interface, you can enter the following command to configure the Web Services Interoperability check:

- `set appfw profile <name> -xmlWSIAction [block] ][log] [learn] [stats] [ none]`

To configure individual Web Services Interoperability rules, you must use the GUI. On the Checks tab of the Modify Web Services Interoperability Check dialog box, select a rule and click Enable or Disable to enable or disable the rule. You can also click Open to open the Web Services Interoperability Detail message box for that rule. The message box displays read-only information about the rule. You cannot modify or make other configuration changes to any of these rules.

The WS-I check uses the rules listed in WS-I Basic Profile 1.0. WS-I delivers best practices for developing interoperable Web Services solutions. WS-I checks are performed only on SOAP Messages.

Description of each WSI standard rule is provided below:

Rule	Description
BP1201	Message body should be a soap:envelope with namespace.
R1000	When an ENVELOPE is a Fault, the soap:Fault element MUST NOT have element children other than faultcode, faultstring, faultactor and detail.
R1001	When an ENVELOPE is a Fault, the element children of the soap:Fault element MUST be unqualified.
R1003	A RECEIVER MUST accept fault messages that have any number of qualified or unqualified attributes, including zero, appearing on the detail element. The namespace of qualified attributes can be anything other than the namespace of the qualified document element Envelope.

Rule	Description
R1004	When an ENVELOPE contains a faultcode element, the content of that element must be either one of the fault codes defined in SOAP 1.1 (supplying additional information if necessary in the detail element), or a Qname whose namespace is controlled by the fault's specifying authority (in that order of preference).
R1005	An ENVELOPE MUST NOT contain soap:encodingStyle attribute on any of the elements whose namespace is the same as the namespace of the qualified document element Envelope.
R1006	An ENVELOPE MUST NOT contain soap:encodingStyle attributes on any element that is a child of soap:Body.
R1007	An ENVELOPE described in an rpc-literal binding MUST NOT contain soap:encodingStyle attribute on any element that is a grandchild of soap:Body.
R1011	An ENVELOPE MUST NOT have any element children of soap:Envelope following the soap:Body element.
R1012	A MESSAGE MUST be serialized as either UTF-8 or UTF-16.
R1013	An ENVELOPE containing a soap:mustUnderstand attribute MUST only use the lexical forms 0 and 1.
R1014	The children of the soap:Body element in an ENVELOPE MUST be namespace qualified.
R1015	A RECEIVER MUST generate a fault if they encounter an envelope whose document element is not soap:Envelope.

Rule	Description
R1031	When an ENVELOPE contains a faultcode element the content of that element must NOT use of the SOAP 1.1 dot notation to refine the meaning of the fault.
R1032	The soap:Envelope, soap:Header, and soap:Body elements in an ENVELOPE MUST NOT have attributes in the same namespace as that of the qualified document element Envelope
R1033	An ENVELOPE SHOULD NOT contain the namespace declaration: `xmlns:xml=http://www.w3.org/XML/1998/namespace.`
R1109	The value of the SOAPAction HTTP header field in a HTTP request MESSAGE MUST be a quoted string.
R1111	An INSTANCE SHOULD use a 200 OK HTTP status code on a response message that contains an envelope that is not a fault.
R1126	An INSTANCE MUST return a 500 Internal Server Error HTTP status code if the response envelope is a Fault.
R1132	A HTTP request MESSAGE MUST use the HTTP POST method.
R1140	A MESSAGE SHOULD be sent using HTTP/1.1.
R1141	A MESSAGE MUST be sent using either HTTP/1.1 or HTTP/1.0.
R2113	An ENVELOPE MUST NOT include the soapenc:arrayType attribute.
R2211	An ENVELOPE described with an rpc-literal binding MUST NOT have the xsi:nil attribute with a value of 1 or true on the part accessors.
R2714	For one-way operations, an INSTANCE MUST NOT return a HTTP response that contains an envelope. Specifically, the HTTP response entity-body must be empty.

Rule	Description
R2729	An ENVELOPE described with an rpc-literal binding that is a response MUST have a wrapper element whose name is the corresponding wsdl:operation name suffixed with the stringResponse.
R2735	An ENVELOPE described with an rpc-literal binding MUST place the part accessor elements for parameters and return value in no namespace.
R2738	An ENVELOPE MUST include all soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding that describes it.
R2740	A wsdl:binding in a DESCRIPTION SHOULD contain a soapbind:fault describing each known fault.
R2744	A HTTP request MESSAGE MUST contain a SOAPAction HTTP header field with a quoted value equal to the value of the soapAction attribute of soapbind:operation, if present in the corresponding WSDL description.

XML message validation check

September 14, 2021

The XML Message Validation check examines requests that contain XML messages to ensure that they are valid. If a request contains an invalid XML message, the Web App Firewall blocks the request. The purpose of the XML Validation check is to prevent an attacker from using specially constructed invalid XML messages to breach the security of your application.

If you use the wizard or the GUI, in the Modify XML Message Validation Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Message Validation Check:

- `set appfw profile <name> -xmlValidationAction [**block**] [**log**] [**stats**] [**none**]`

You must use the GUI to configure the other XML Validation check settings. In the Modify XML Message Validation Check dialog box, on the Checks tab, you can configure the following settings:

- **XML Message Validation.** Use one of the following options to validate the XML message:
  - **SOAP Envelope.** Validate only the SOAP envelope of XML messages.
  - **WSDL.** Validate XML messages by using an XML SOAP WSDL. If you choose WSDL validation, in the WSDL Object drop-down list you must choose a WSDL. If you want to validate against a WSDL that has not already been imported to the Web App Firewall, you can click the Import button to open the Manage WSDL Imports dialog box and import your WSDL. See WSDL for more information.
    - If you want to validate the entire URL, leave the Absolute radio button in the End Point Check button array selected. If you want to validate only the portion of the URL after the host, select the Relative radio button.
    - If you want the Web App Firewall to enforce the WSDL strictly, and not allow any additional XML headers not defined in the WSDL, you must clear the Allow additional headers not defined in the WSDL check box.
      Caution: If you uncheck the
      Allow Additional Headers not defined in the WSDL check box, and your WSDL does not define all XML headers that your protected XML application or Web 2.0 application expects or that a client sends, you may block legitimate access to your protected service.
  - **XML Schema.** Validate XML messages by using an XML schema. If you choose XML schema validation, in the XML Schema Object drop-down list you must choose an XML schema. If you want to validate against an XML schema that has not already been imported to the Web App Firewall, you can click the Import button to open the Manage XML Schema Imports dialog box and import your WSDL. See WSDL for more information.
- **Response Validation.** By default, the Web App Firewall does not attempt to validate responses. If you want to validate responses from your protected application or Web 2.0 site, select the Validate Response check box. When you do, the Reuse the XML Schema specified in request validation check box and the XML Schema Object drop-down list are activated.
  - Check the Reuse XML Schema check box to use the schema you specified for request validation to do response validation as well.
    Note: If you check this check box, the
    XML Schema Object drop-down list is grayed out.
  - If you want to use a different XML schema for response validation, use the XML Schema Object drop-down list to select or upload that XML schema.

XML SOAP fault filtering check

September 14, 2021

The XML SOAP fault filtering check examines responses from your protected web services and filters out XML SOAP faults. This prevents leaking of sensitive information to attackers.

If you use the wizard or the GUI, in the Modify XML SOAP Fault Filtering Check dialog box, on the **General** tab you can enable or disable the Block, Log, and Statistics actions, and the Remove action, which removes SOAP faults before forwarding the response to the user.

If you use the command-line interface, you can enter the following command to configure the XML SOAP Fault Filtering Check:

```
set appfw profile <name> -XMLSOAPFaultAction [block] [log] [stats] [none]
```

You cannot configure exceptions to the XML SOAP Fault Filtering check. You can only enable or disable it.

JSON Protection Checks

September 14, 2021

Citrix Web App Firewall protects your JSON applications from content-level DoS, SQL, or cross-site scripting attacks. When an JSON request has a DoS, SQL, or cross-site scripting attack, you must protection your application by configuring limits on JSON structures such as arrays and strings.

> **Note:**
>
> The JSON security checks apply only to content that is sent with a JSON content-type header. If the content-type header is missing, or is set to a different value, all JSON security checks are bypassed. If you want to protect your JSON applications, the webmasters of each web server that hosts those applications must ensure a proper JSON content-type header is sent.
>
> The learning feature is not support for JSON SQL, cross-site scripting, DOS content types.

JSON Denial-of-Service protection check

September 14, 2021

The JSON denial-of-service (DoS) check examines an incoming JSON request and validates if there is any data that matches the characteristics of a DoS attack. If the request had JSON violations, the

appliance blocks the request, logs the data, sends an SNMP alert, and also displays a JSON error page. The purpose of the JSON DoS check is to prevent an attacker from sending JSON request to launch DoS attacks on your JSON applications or website.

When a client sends a request to a Citrix ADC appliance, the JSON parser parses the request payload and if a violation is observed, the appliance enforces constraints on the JSON structure. The constraint enforces a size limit on the JSON request. As a result, if any JSON violation was observed, the appliance applies an action and responds with the JSON error page.

JSON DoS rules

When the appliance receives a JSON request, The JSON DOS protection enforces size limit on the following DoS parameters in the request payload.

1. maximum depth: Maximum nesting (depth) of JSON document. This check protects against documents that have excessive depth of hierarchy.
2. maximum document length: Maximum document length of JSON document.
3. maximum array length: Maximum array length in the any of JSON object. This check protects against arrays having large lengths.
4. maximum string length: Maximum string length in the JSON. This check protects against strings that have large length.
5. maximum object key count: Maximum key count in the any of JSON object. This check protects against objects that have large number of keys.
6. maximum object key length: Maximum key length in the any of JSON object. This check protects against objects that have large keys.

Following is a list of JSON DoS rules validated during JSON parsing.

1. JSONMaxContainerDepth. This check can be enabled by configuring the JSONMaxContainerDepth check and by default the option is OFF.
2. JSONMaxContainerDepth. This check can be enabled/disabled by configurable option JSONMaxContainerDepthCheck and default value can be changed by option JSONMaxContainerDepth. However, you can vary the maximum levels to a value ranging from 1 to 127. Default value: 5, Minimum value: 1, Maximum value: 127
3. JSONMaxDocumentLength. This check can be enabled by configuring the JSONMaxDocumentLength check and the default option is OFF.
4. JSONMaxDocumentLength. This check can be enabled by configuring the JSONMaxDocumentLength check and the default length is set to 20000000 bytes. Minimum value: 1, Maximum value: 2147483647
5. JSONMaxObjectKeyCount. The rule validates if the JSON maximum object key count check is turned on or off. Possible values: ON, OFF, Default value: OFF

6. JSONMaxObjectKeyCount. This check can be enabled by configuring the JSONMaxObjectKeyCount check. The check protects against objects that have large number of keys and the default value is set to 1000 bytes. Minimum value: 0, Maximum value: 2147483647
7. JSONMaxObjectKeyLength. This check can be enabled by configuring the JSONMaxObjectKeyLength check. The rule validates if the JSON maximum object key length check is turned on or off. By default it is turned OFF.
8. JSONMaxObjectKeyLength. The check protects against objects that have large key length. Default value: 128. Minimum value: 1, Maximum value: 2147483647
9. JSONMaxArrayLength. The rule validates if the JSON maximum array length check is ON or OFF. By default it is turned off.
10. JSONMaxArrayLength. The check protects against arrays that has large lengths. By default, the value is set to 10000. Minimum value: 1, Maximum value: 2147483647
11. JSONMaxStringLength. This check can be enabled by configuring the JSONMaxStringLength check. The check validates if the JSON maximum string length is ON or OFF. By default it is turned off.
12. JSONMaxStringLength. The check protects against strings that has large length. By default, it is set to 1000000. Minimum value: 1, Maximum value: 2147483647

Configure JSON DoS protection check

For configure JSON DoS protection, you must complete the following steps:

1. Add application firewall profile for JSON.
2. Set application firewall profile for JSON DoS settings.
3. Configure JSON DoS variables by binding the application firewall profile.

Add application firewall profile for JSON DoS protection

You must first create a profile that specifies how the application firewall must protect your JSON web content from JSON DoS attack.
At the command prompt, type:

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> *Note*:
>
> When you set the profile type as JSON, other checks such as HTML or XML will not applicable.

Example

```
add appfw profile profile1 -type JSON
```


Set application firewall profile for JSON DoS protection

You must configure the profile for one or more JSON DoS actions and JSON DoS error object to be set on the application firewall profile.
At the command prompt, type:

```
set appfw profile <name> -JSONDoSAction [block] | [log] | [stats] | [none]
```

Block - Block connections that violate this security check.
Log - Log violations of this security check.
Stats - Generate statistics for this security check.
None - Disable all actions for this security check.

> *Note*:
>
> To enable one or more actions, type "set appfw profile -JSONDoSAction" followed by the actions to be enabled.

Example

```
set appfw profile profile1 -JSONDoSAction block log stat
```

Configure DoS variables by binding application firewall profile

To provide JSON DoS protection, you must bind the application firewall profile with JSON DoS settings.
At the command prompt, type:

```
bind appfw profile <name> -JSONDoSURL <expression> [-JSONMaxContainerDepthCheck
 ( ON | OFF )[-JSONMaxContainerDepth <positive_integer>]] [-JSONMaxDocumentLengthChec
 ( ON | OFF )[-JSONMaxDocumentLength <positive_integer>]] [-JSONMaxObjectKeyCountChec
 ( ON | OFF )[-SONMaxObjectKeyCount <positive_integer>]] [-JSONMaxObjectKeyLengthChec
 ( ON | OFF )[-JSONMaxObjectKeyLength <positive_integer>]] [-JSONMaxArrayLengthCheck
 ( ON | OFF )[-JSONMaxArrayLength <positive_integer>]] [-JSONMaxStringLengthCheck
 ( ON | OFF )[-JSONMaxStringLength <positive_integer>]]
```

Example

```
bind appfw profile profile1 -JSONDoSURL  ".*"  -JSONMaxContainerDepthCheck ON
```

> *Note*:
>
> The JSON DoS checks will be applicable only if the profile type is selected as JSON. Also, the SQL,

> cross-site scripting, Field format and Form field signatures are applied on Query parameters in cases of JSON profile.

Import JSON errorpage

If an incoming request had a DoS attack and when you block the request, the appliance displays an error message. For doing this, you must import the JSON error page.
At the command prompt, type:

```
import appfw jsonerrorpage <src> <name> [-comment <string>] [-overwrite]
```

Where,

src. URL (protocol, host, path, and name) for the location at which to store the imported JSON error object.

> *Note*:
>
> The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. This is a mandatory argument. Maximum Length: 2047.

Name. Name to assign to the JSON error object on the Citrix ADC. This is a mandatory argument. Maximum Length: 31
Comment. Any comments to preserve information about the JSON error object. Maximum Length: 255
Overwrite. Overwrite any existing JSON error object of the same name.

Sample configuration

```
1 Add appfw prof profjson - type JSON
2 Bind appfw prof profjson - JSONDoSURL ".*" -
    JSONMaxDocumentLengthCheck ON -JSONMaxDocumentLength 30 -
    JSONMaxContainerDepthCheck ON -JSONMaxContainerDepth 3
    JSONMaxObjectKeyCountCheck ON -JSONMaxObjectKeyCount 4 -
    JSONMaxObjectKeyLengthCheck ON -JSONMaxObjectKeyLength 10 -
    JSONMaxArrayLengthCheck ON -JSONMaxArrayLength 5 -
    JSONMaxStringLengthCheck ON -JSONMaxStringLength 30
3 <!--NeedCopy-->
```

Sample Payloads, Log Messages and Counters:

JSONMaxDocumentLength Violation

JSONMaxDocumentLength: 30
Payload: {"a":"A","b":"B","c":"C","d":"D","e":"E"}

Log Message:

```
Document Length exceeds 20000000 May 29 20:23:32 <local0.info>
    10.217.31.243 05/29/2019:20:23:32 GMT 0-PPE-0 : default APPFW
    APPFW_JSON_DOS_MAX_DOCUMENT_LENGTH 136 0 : 10.217.32.134 114-PPE0 -
    profjson http://10.217.30.120/forms/login.html Document exceeds
    maximum document length (30). cn1=30467 cn2=115 cs1=profjson cs2=
    PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

Counters:

```
1 0 6 as_viol_json_dos
2 0 3 as_viol_json_dos_max_document_length
3 0 6 as_log_json_dos
4 0 3 as_log_json_dos_max_document_length
5 0 6 as_viol_json_dos_profile appfw__(profile1)
6 0 3 as_viol_json_dos_max_document_length_profile appfw__(profile1)
7 0 6 as_log_json_dos_profile appfw__(profile1)
8 0 3 as_log_json_dos_max_document_length_profile appfw__(profile1)
<!--NeedCopy-->
```

JSONMaxContainerDepth Violation

JSONMaxContainerDepth: 3
Payload: {“a”: {”b”: {“c”: {“d”: { ”e” : “f” }}}}}

Log Message:

```
May 29 19:33:59 <local0.info> 10.217.31.243 05/29/2019:19:33:59 GMT 0-
    PPE-1 : default APPFW APPFW_JSON_DOS_MAX_CONTAINER_DEPTH 4626 0 :
    10.217.31.247 22-PPE1 - profjson http://10.217.30.120/forms/login.
    html Document at offset (15) exceeds maximum container depth (3).
    cn1=30466 cn2=113 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=
    blocked
<!--NeedCopy-->
```

Counters:

```
36 20999 7 1 0 as_viol_json_dos
37 0 6 1 0 as_viol_json_dos_max_container_depth
38 0 7 1 0 as_log_json_dos
39 0 6 1 0 as_log_json_dos_max_container_depth
40 0 7 1 0 as_viol_json_dos_profile appfw__(profile1)
```



```
41 0 6 1 0 as_viol_json_dos_max_container_depth_profile appfw__(
    profile1)
42 0 7 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 6 1 0 as_log_json_dos_max_container_depth_profile appfw__(profile1
    )
<!--NeedCopy-->
```

JSONMaxObjectKeyCount Violation

JSONMaxObjectKeyCount: 4

Payload: {“a”: ”A”, “b”: “B”, “c”:“C”, “d” :“D”, ”e” : “E” }

Log Message:

```
May 30 19:42:41 <local0.info> 10.217.31.243 05/30/2019:19:42:41 GMT 0-
    PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_COUNT 457 0 :
    10.217.32.134 219-PPE1 - profjson http://10.217.30.120/forms/login.
    html Object at offset (41) that exceeds maximum key count (4). cn1
    =30468 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

Counters:

```
94 119105 15 1 0 as_viol_json_dos
95 0 4 1 0 as_viol_json_dos_max_object_key_count
96 0 15 1 0 as_log_json_dos
97 0 4 1 0 as_log_json_dos_max_object_key_count
98 0 15 1 0 as_viol_json_dos_profile appfw__(profile1)
99 0 4 1 0 as_viol_json_dos_max_object_key_count_profile appfw__(
    profile1)
100 0 15 1 0 as_log_json_dos_profile appfw__(profile1)
101 0 4 1 0 as_log_json_dos_max_object_key_count_profile appfw__(
    profile1)
<!--NeedCopy-->
```

JSONMaxObjectKeyLength Violation

JSONMaxObjectKeyLength: 10

Payload: {“a”: ”A”, “b1234567890”: “B”, “c”:“C”, “d” :“D”, ”e” : “E” }

Log Message:


```
May 31 20:26:10 <local0.info> 10.217.31.243 05/31/2019:20:26:10 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_LENGTH 102 0 : 10.217.32.134 89-PPE1 - profjson http://10.217.30.120/forms/login.html Object key(b1234567890) at offset (12) exceeds maximum key length (10). cn1=30469 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

Counters:

```
242172 6 1 0 as_viol_json_dos
0 1 1 0 as_viol_json_dos_max_object_key_length
10 0 5 1 0 as_log_json_dos
11 0 1 1 0 as_log_json_dos_max_object_key_length
12 0 6 1 0 as_viol_json_dos_profile appfw__(profile1)
13 0 1 1 0 as_viol_json_dos_max_object_key_length_profile appfw__(profile1)
14 0 5 1 0 as_log_json_dos_profile appfw__(profile1)
15 0 1 1 0 as_log_json_dos_max_object_key_length_profile appfw__(profile1)
<!--NeedCopy-->
```

JSONMaxArrayLength Violation

JSONMaxArrayLength: 5

Payload: {“a”: ”A”, “c”:[”d”,”e”,”f”,”g”,”h”,”i”],”e”:[“E”,”e”]}

Log Message:

```
May 29 20:58:39 <local0.info> 10.217.31.243 05/29/2019:20:58:39 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_ARRAY_LENGTH 4650 0 : 10.217.32.134 153-PPE1 -profjson http://10.217.30.120/forms/login.html Array at offset (37) that exceeds maximum array length (5). cn1=30469 cn2=120 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

Counters:

```
36 182293 10 1 0 as_viol_json_dos
37 0 1 1 0 as_viol_json_dos_max_array_length
38 0 10 1 0 as_log_json_dos 39 0 1 1 0 as_log_json_dos_max_array_length
40 0 10 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 1 1 0 as_viol_json_dos_max_array_length_profile appfw__(profile1)
42 0 10 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 1 1 0 as_log_json_dos_max_array_length_profile appfw__(profile1))
```



```
<!--NeedCopy-->
```

JSONMaxStringLength Violation

JSONMaxStringLength: 10

Payload: {“a”: ”A”, “c”:”CcCcCcCcCcCcCcCcCc”,”e”:[“E”,”e”]}

Log Message:

```
May 29 20:05:02 <local0.info> 10.217.31.243 05/29/2019:20:05:02 GMT 0-PPE-0 : default APPFW APPFW_JSON_DOS_MAX_STRING_LENGTH 134 0 : 10.217.32.134 80-PPE0 - profjson http://10.217.30.120/forms/login.html String(CcCcCcCcCcCcCc) at offset (27) that exceeds maximum string length (10). n1=30470 cn2=122 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

Counters:

```
44 91079 3 1 0 as_viol_json_dos
45 0 1 1 0 as_viol_json_dos_max_string_length
46 0 3 1 0 as_log_json_dos
47 0 1 1 0 as_log_json_dos_max_string_length
48 0 3 1 0 as_viol_json_dos_profile appfw__(profile1)
49 0 1 1 0 as_viol_json_dos_max_string_length_profile appfw__(profile1)
50 0 3 1 0 as_log_json_dos_profile appfw__(profile1)
51 0 1 1 0 as_log_json_dos_max_string_length_profile appfw__(profile1
<!--NeedCopy-->
```

Configure JSON DoS protection by using Citrix GUI

Follow the procedure below to set the JSON DoS protection settings.

1. On the navigation pane, navigate to **Security** > **Profiles**.
2. In the **Profiles** page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Security Checks under Advanced Settings**.
4. In the **Security Checks** section, go to **JSON Denial of Service** settings.
5. Click the executable icon near the checkbox.



6. Click **Action Settings** to access the **JSON Denial of Service Settings** page.
7. Select the JSON DoS action.
8. Click **OK**.



9. In the **Citrix Web App Firewall Profile** page, click **Relaxation Rules** under **Advanced Settings**.
10. In **Relaxation Rules** section, select **JSON Denial of Service** settings and click **Edit**.



11. In the **Application Firewall JSON Denial of Service Check** set the JSON DoS validation values.
12. Click **OK**.



13. In the **Citrix Web App Firewall Profile** page, click **Profile Settings** under **Advanced Settings**.

14. In the **Profile Settings** section, go to **JSON Error Settings** sub section to set **JSON DoS error** page.



15. In the **JSON Error Page Import Object** page, set the following parameters:

    a) Import from. Import the error page as text, file or URL.
    b) URL. URL to redirect the user to the error page.
       1 File. Select a file to be imported as JSON DoS error file.
    c) Text. Enter the JSON file contents.
    d) Click Continue.
    e) File. Enter the file name.
    f) File Content. Add the error file content.
    g) Click **OK**.



16. Click **OK**.
17. Click **Done**.

JSON SQL Injection protection check

September 14, 2021

An incoming JSON request can have SQL injection in the form of partial SQL query strings or unauthorized commands in the code. This leads to stealing of data from the JSON database of your web servers. On receiving such request, the appliance blocks such request to protection your data.

Consider a scenario, where a client sends a JSON SQL request to a Citrix ADC appliance, the JSON parser parses the request payload and if an SQL Injection is observed, the appliance enforces constraints on the JSON SQL content. The constraint enforces a size limit on the JSON SQL request. As a result, if any JSON SQL Injection is observed, the appliance applies an action and responds with the JSON SQL error page.

Configure JSON SQL Injection protection

For configure JSON SQL protection, you must complete the following steps:

1. Add application firewall profile as JSON.
2. Set application firewall profile for JSON SQL Injection settings
3. Configure JSON SQL action by binding the application firewall profile.

Add application firewall profile of type JSON

You must first create a profile that specifies how the application firewall must protect your JSON web content from JSON SQL Injection attack.
At the command prompt, type:

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> *Note*:
>
> When you set the profile type as JSON, other checks such as HTML or XML will not applicable.

Example

```
add appfw profile profile1 -type JSON
```


Configure JSON SQL Injection action

You must configure one or more JSON SQL Injection actions to protection your application from JSON SQL injection attacks.
At the command prompt, type:

```
set appfw profile <name> - JSONSQLInjectionAction [block] [log] [stats] [
none]
```

SQL Injection actions are:
Block - Block connections that violate this security check.
Log - Log violations of this security check.
Stats - Generate statistics for this security check.
None - Disable all actions for this security check.

Configure JSON SQL Injection type

To configure the JSON SQL Injection type on an application firewall profile, at the command prompt, type:

```
set appfw profile <name> - JSONSQLInjectionType <JSONSQLInjectionType>
```

Example

```
set appfw profile profile1 -JSONSQLInjectionType SQLKeyword
```

Where the available SQL Injection types are:
Available SQL injection types.
SQLSplChar. Checks for SQL Special Characters,
SQLKeyword. Checks for SQL Keywords.
SQLSplCharANDKeyword. Checks for both and blocks if found.
SQLSplCharORKeyword. . Blocks if SQL special character or spl keyword is found.
Possible values: SQLSplChar, SQLKeyword, SQLSplCharORKeyword, SQLSplCharANDKeyword.

> *Note*:
> To enable one or more actions, type “set appfw profile - JSONSQLInjectionAction” followed by the actions to be enabled.

Example

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

The following example shows a sample payload, its corresponding log message and statistics counters:


```
Payload:
=======
{

  "test": "data",
  "username": "waf",
  "password": "select * from t1;",
  "details": {

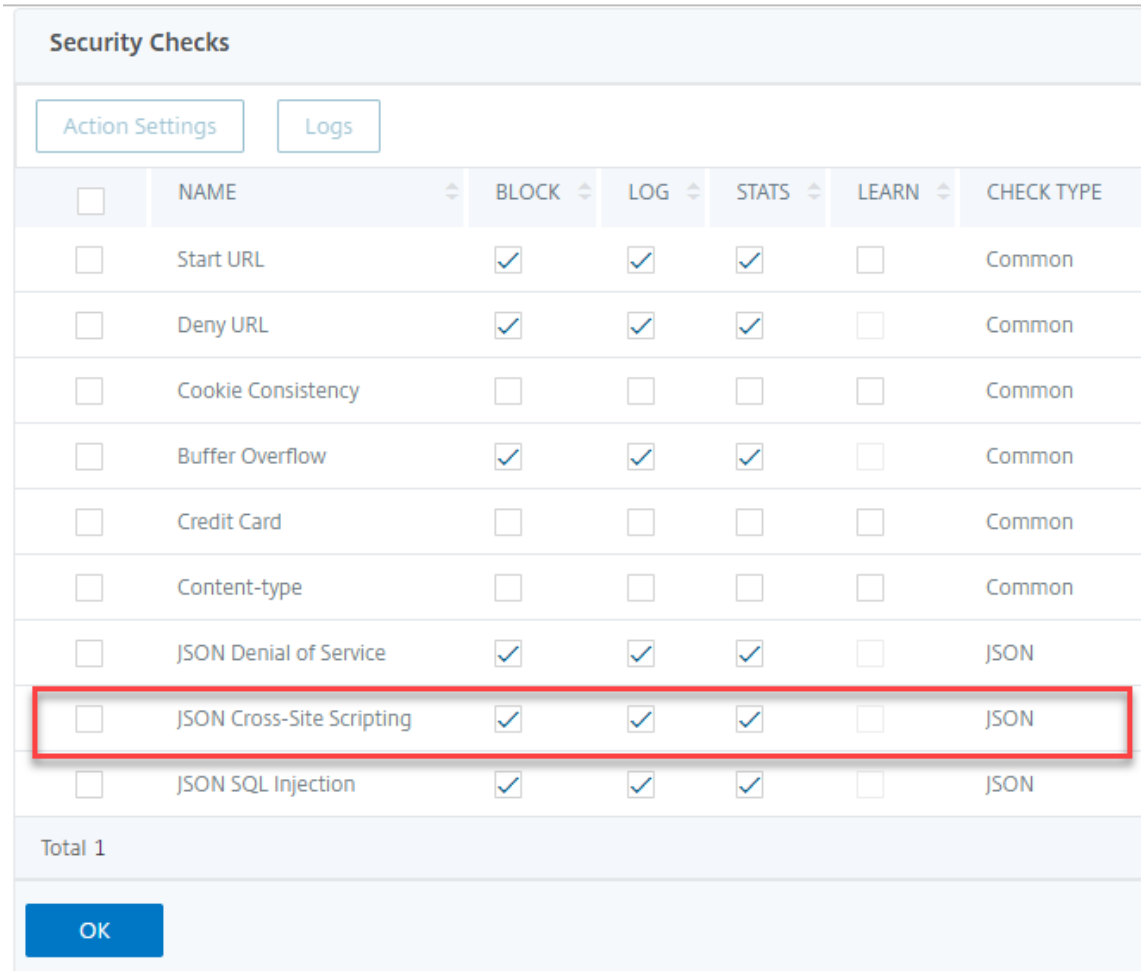
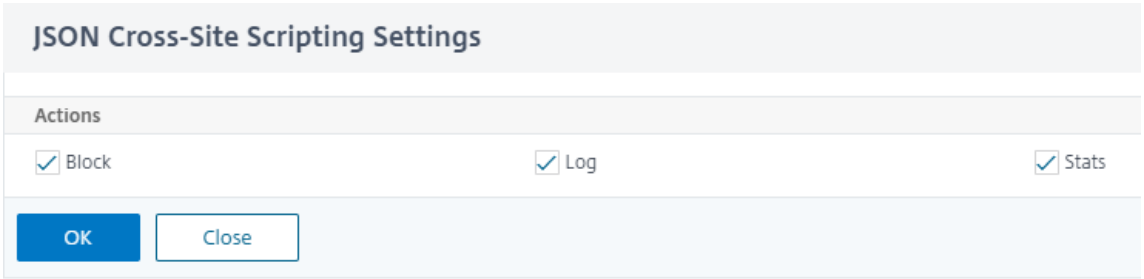
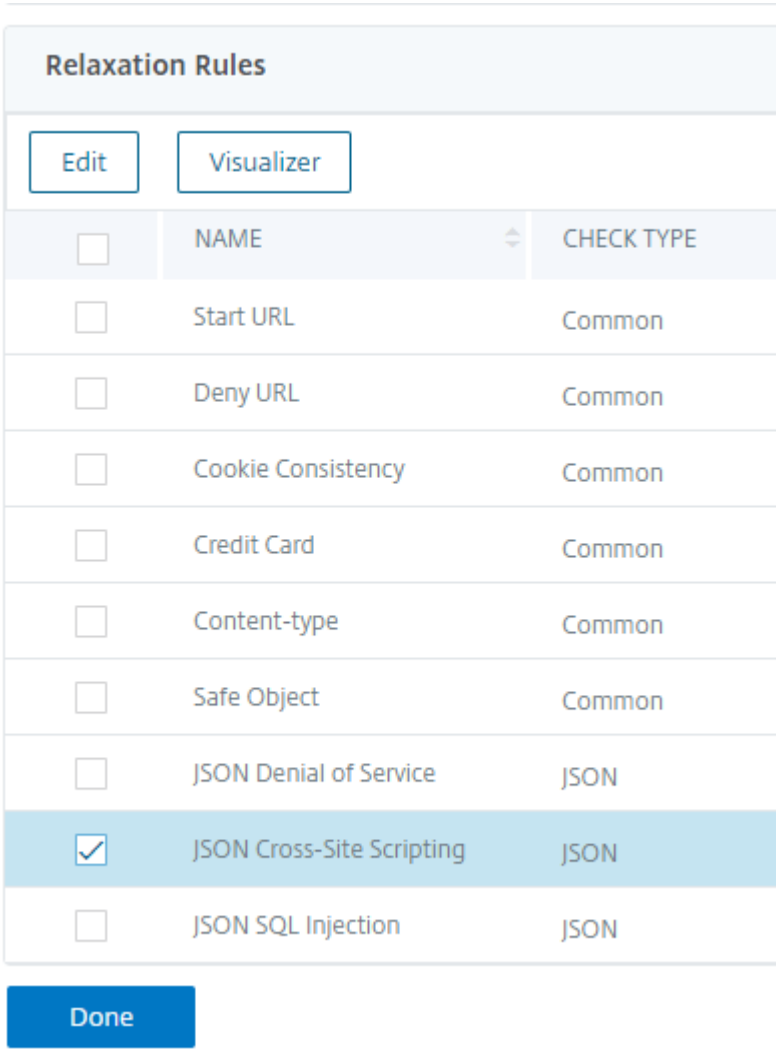
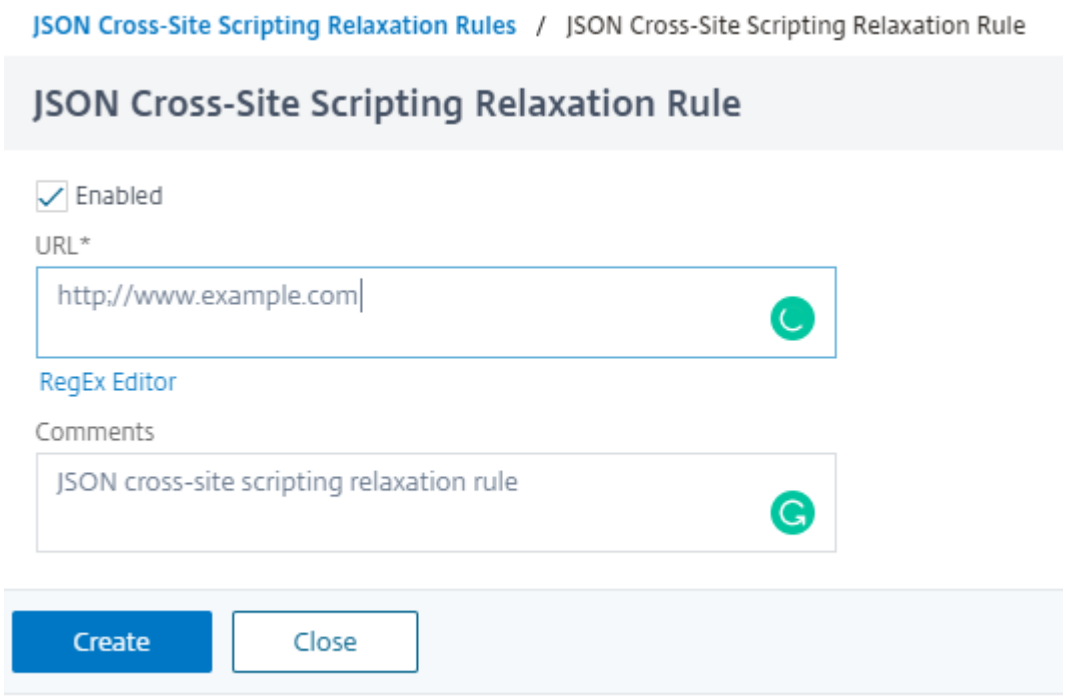
    "surname": "test",
    "age": "23"
   }

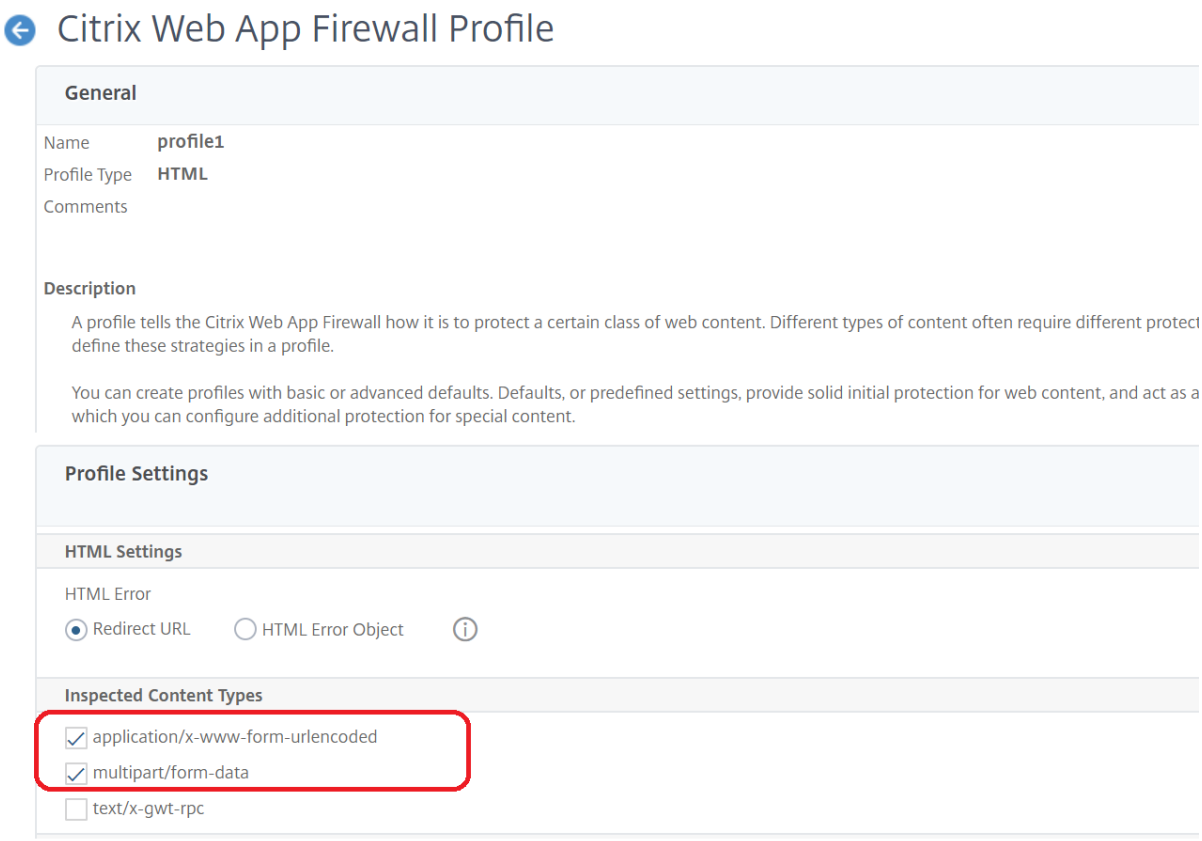
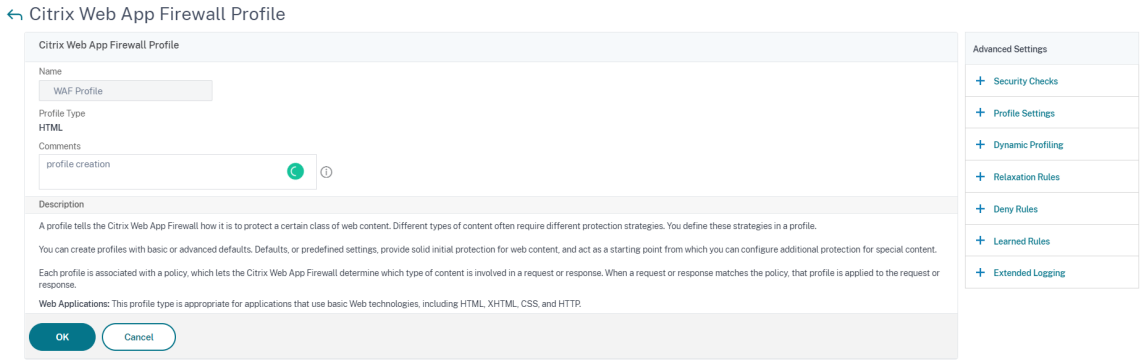
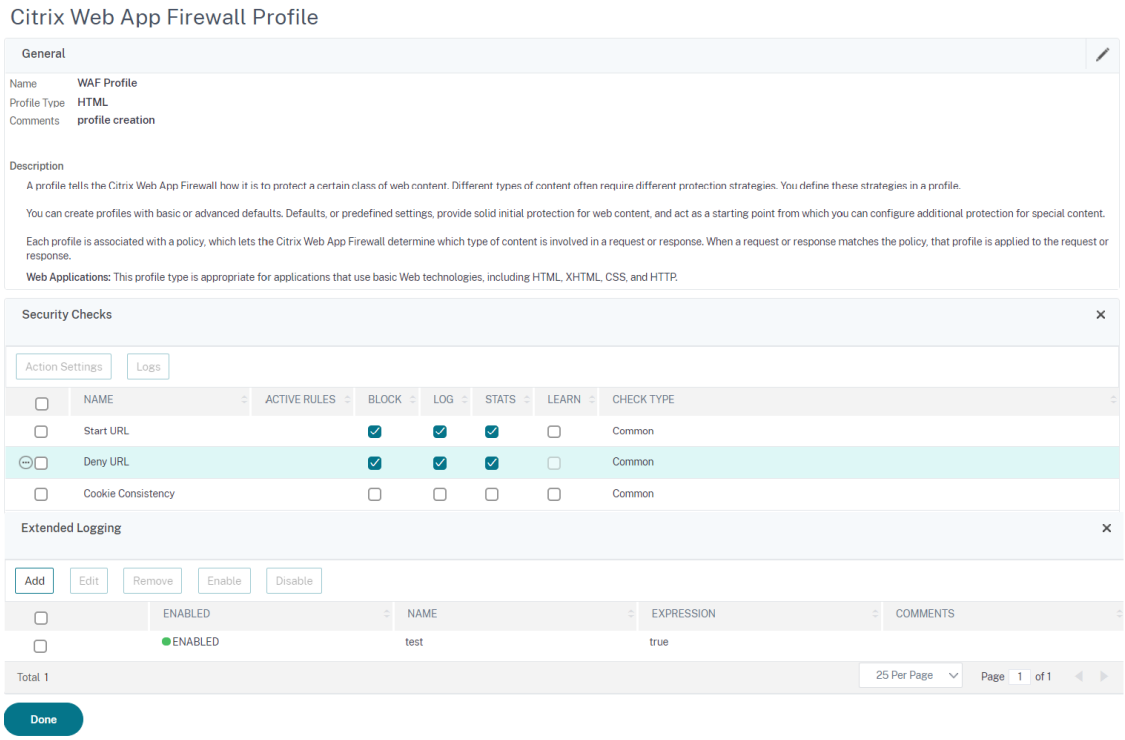
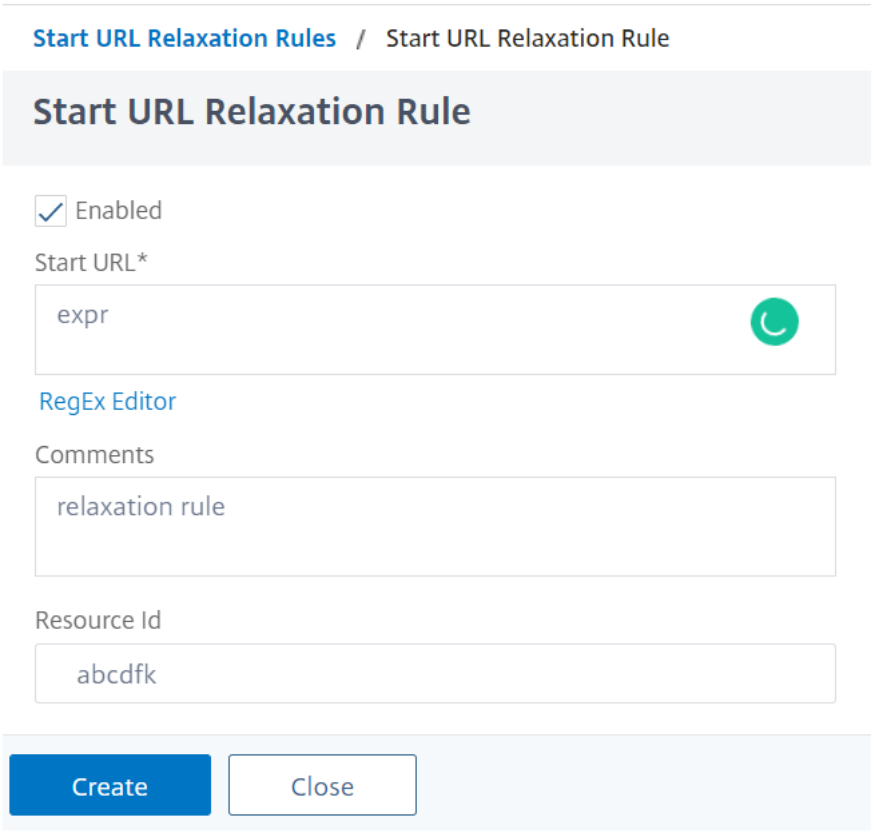
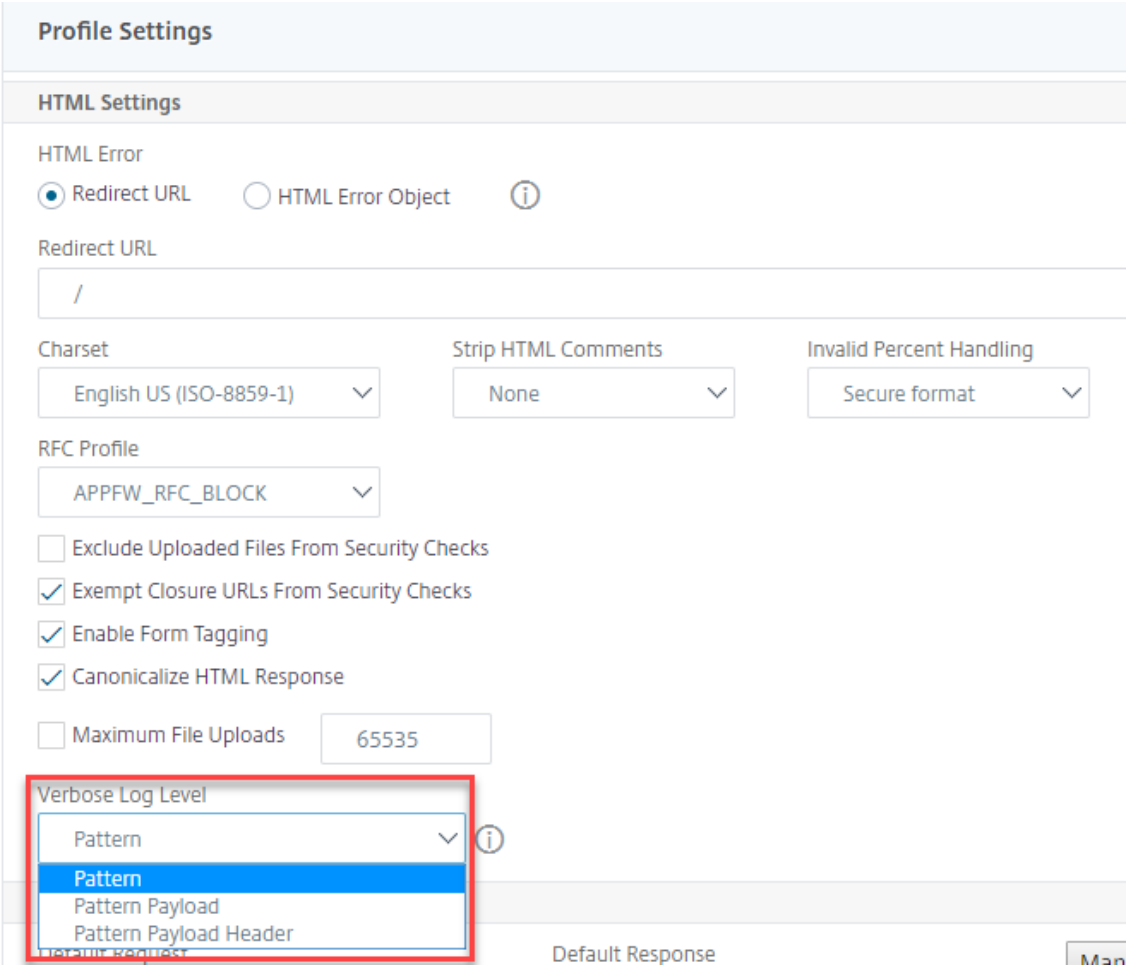
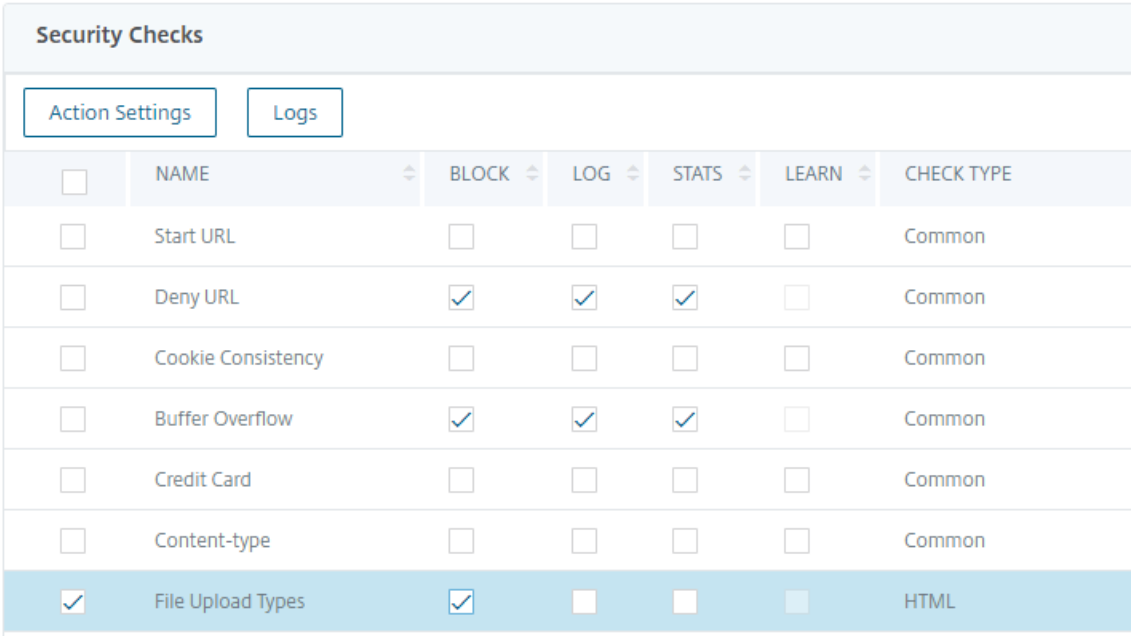
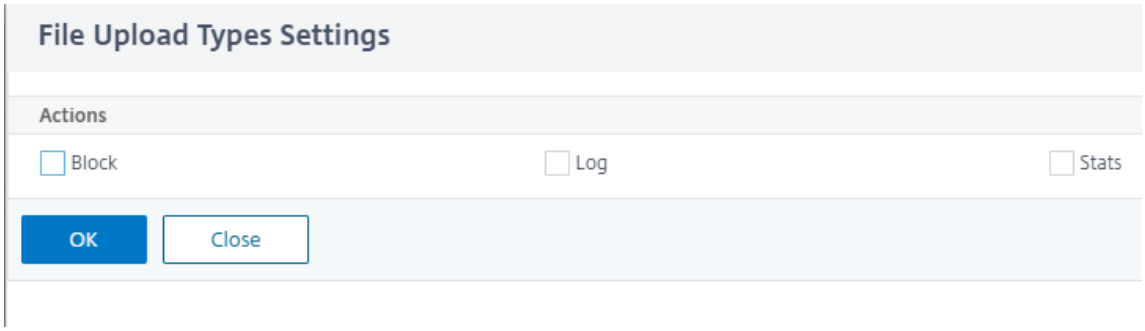
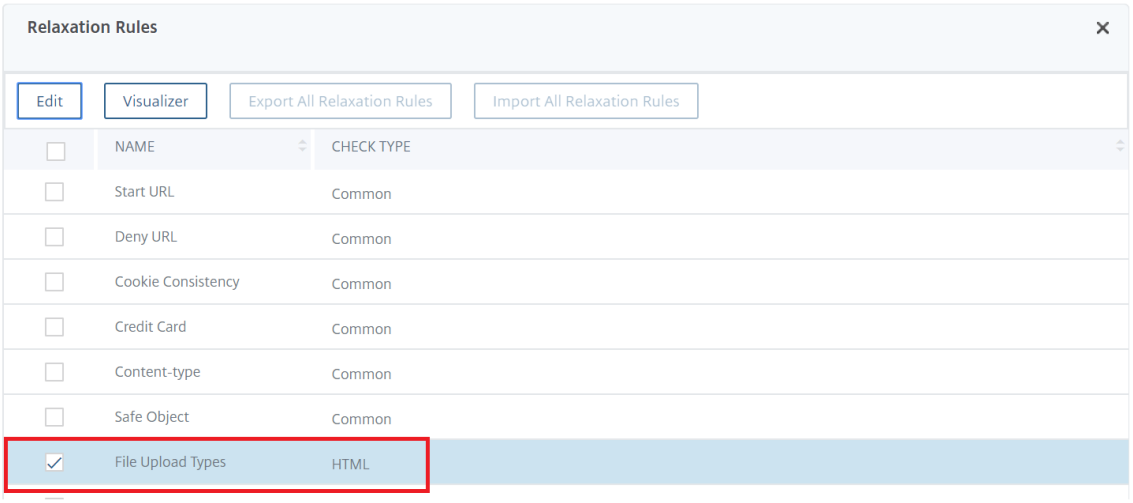
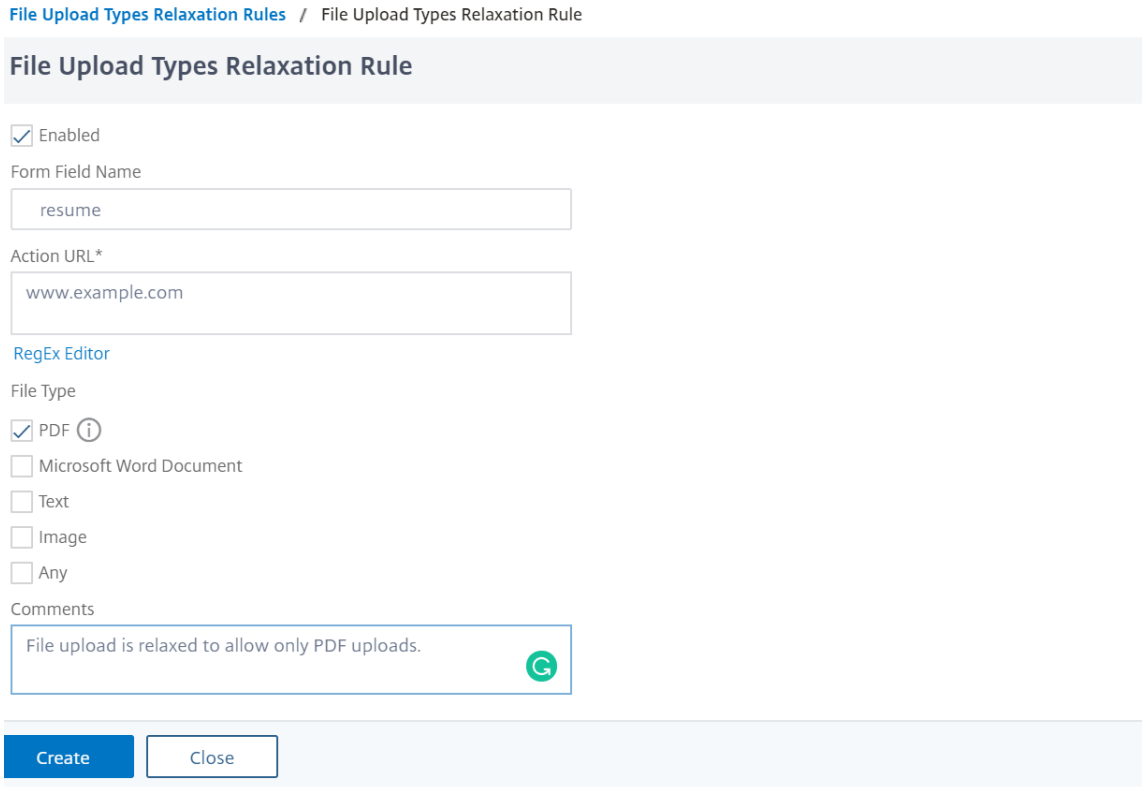
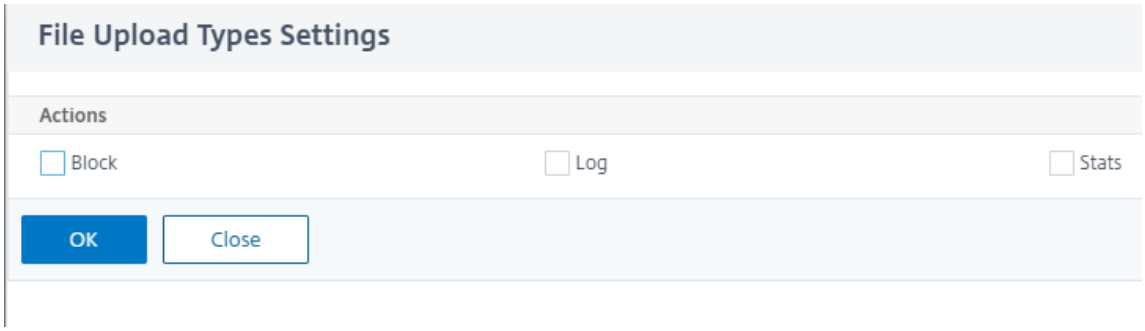
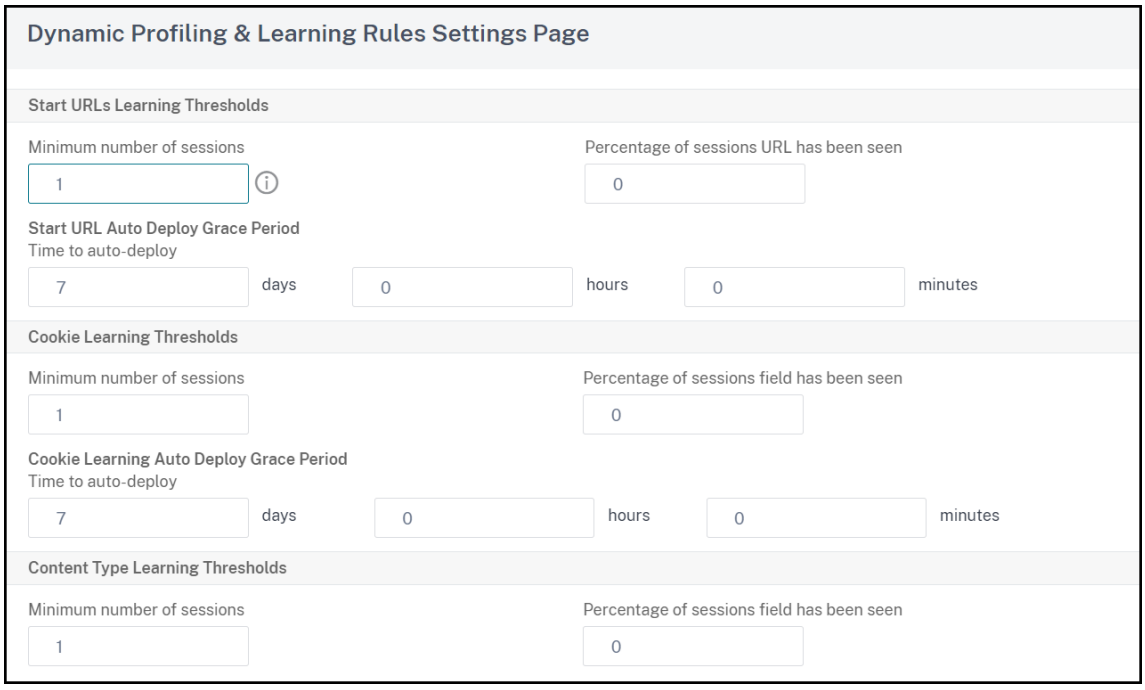
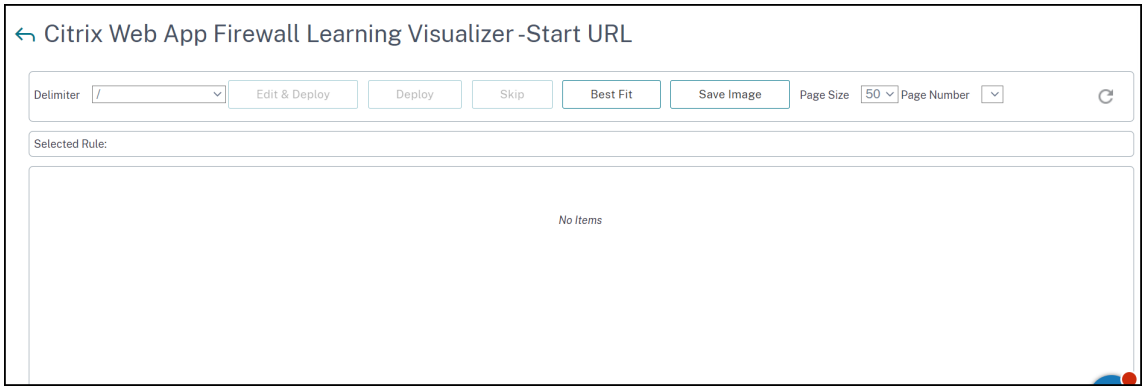
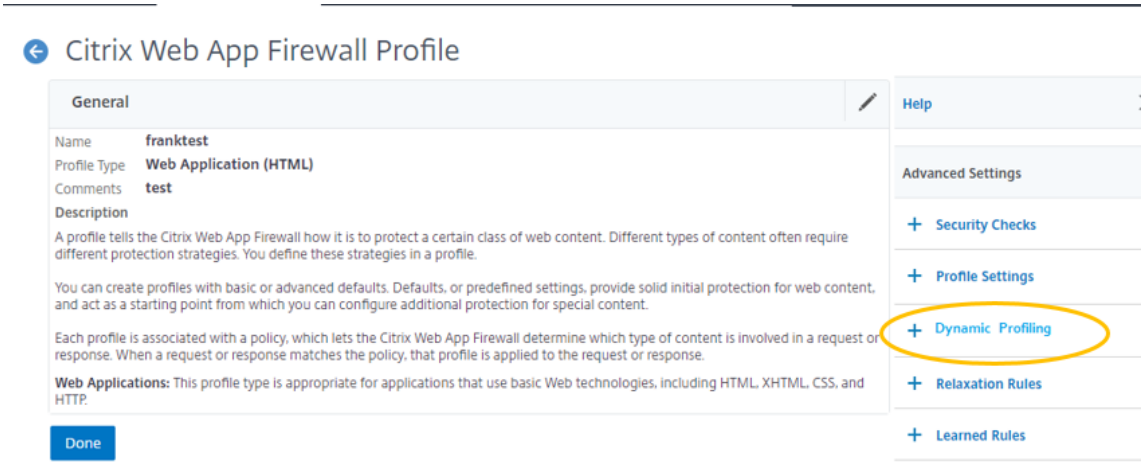
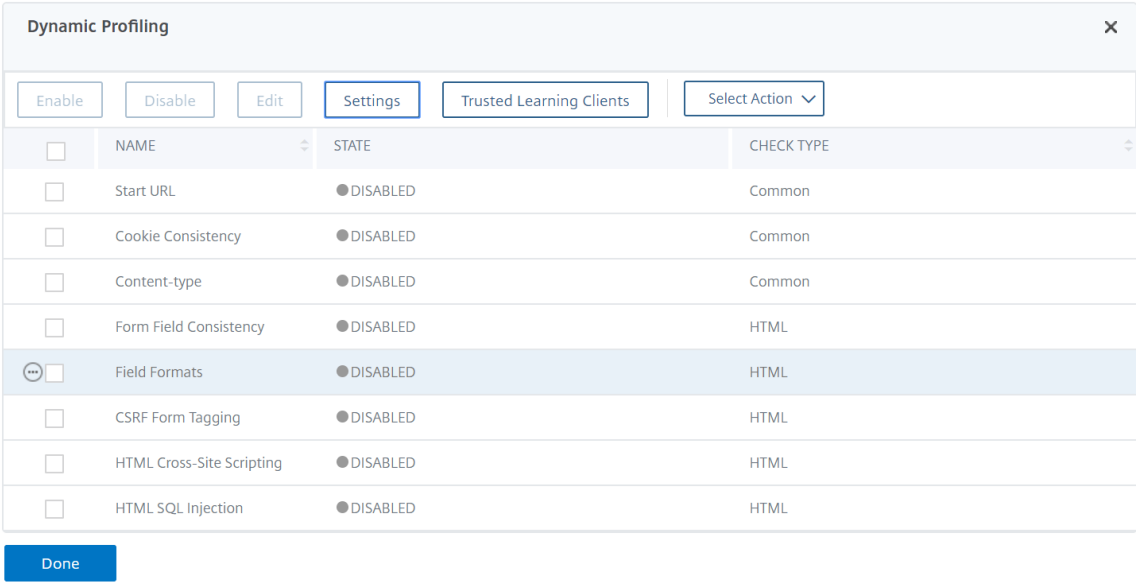
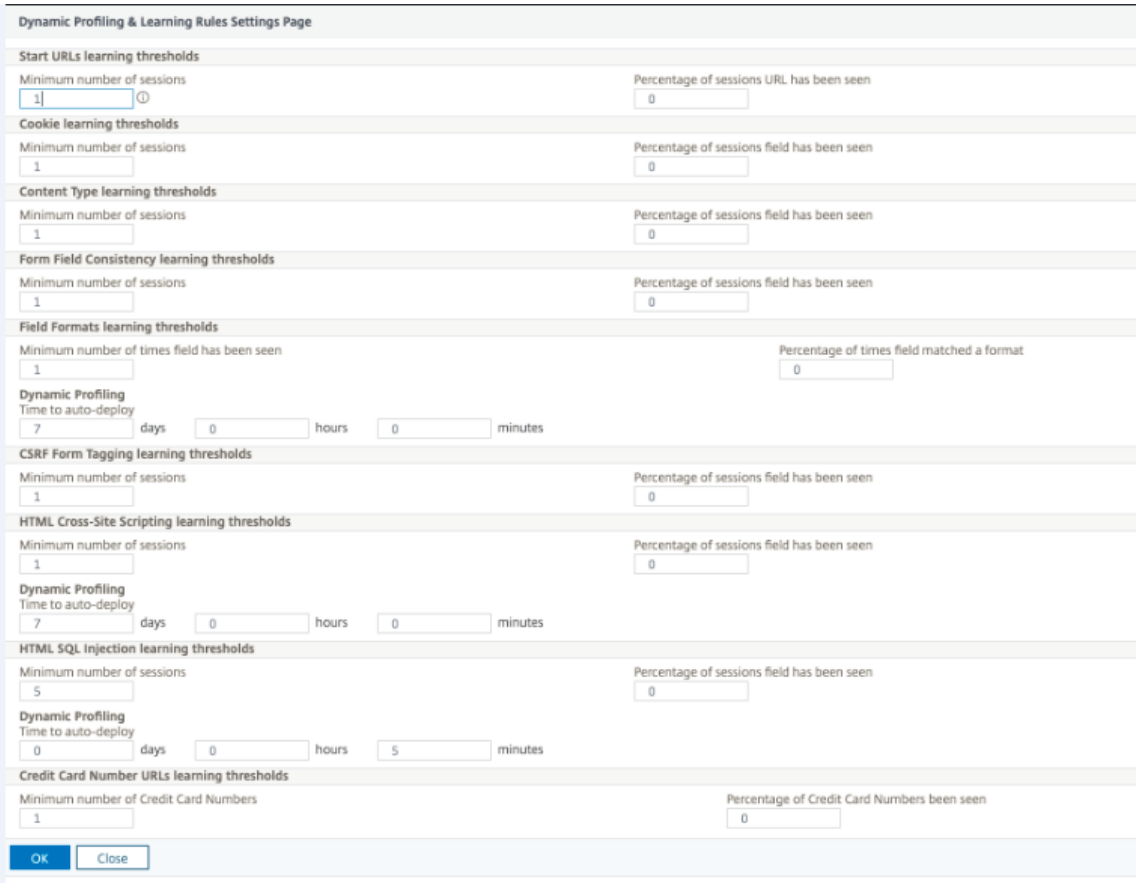
 }

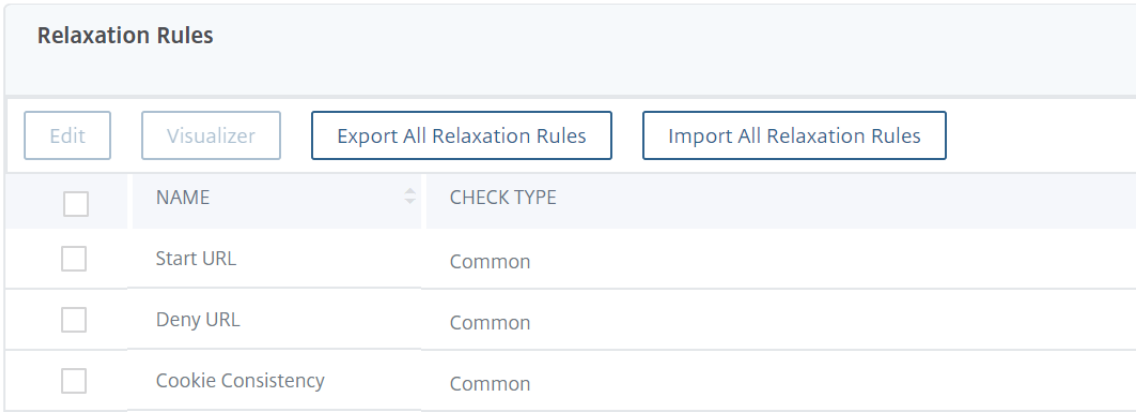
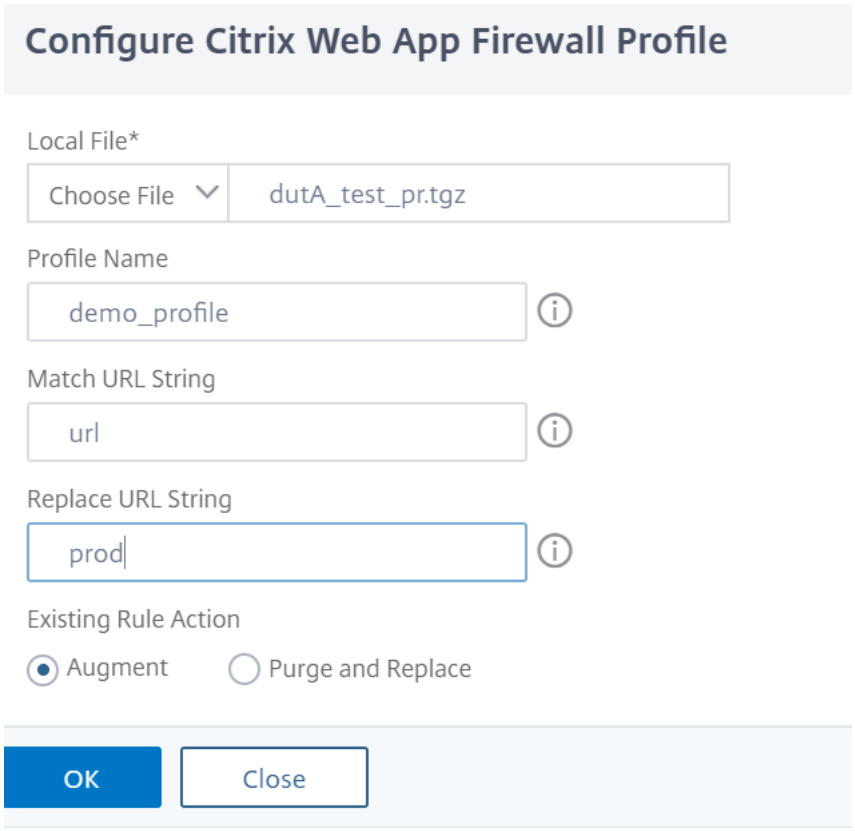
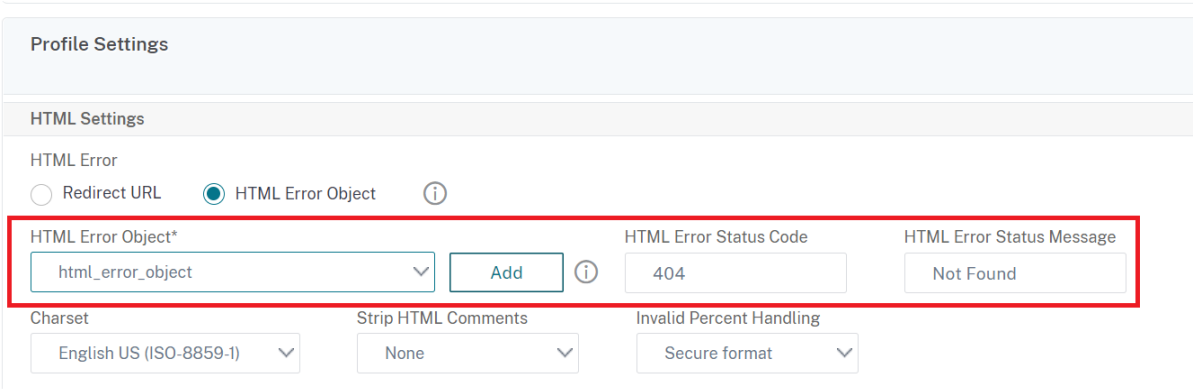
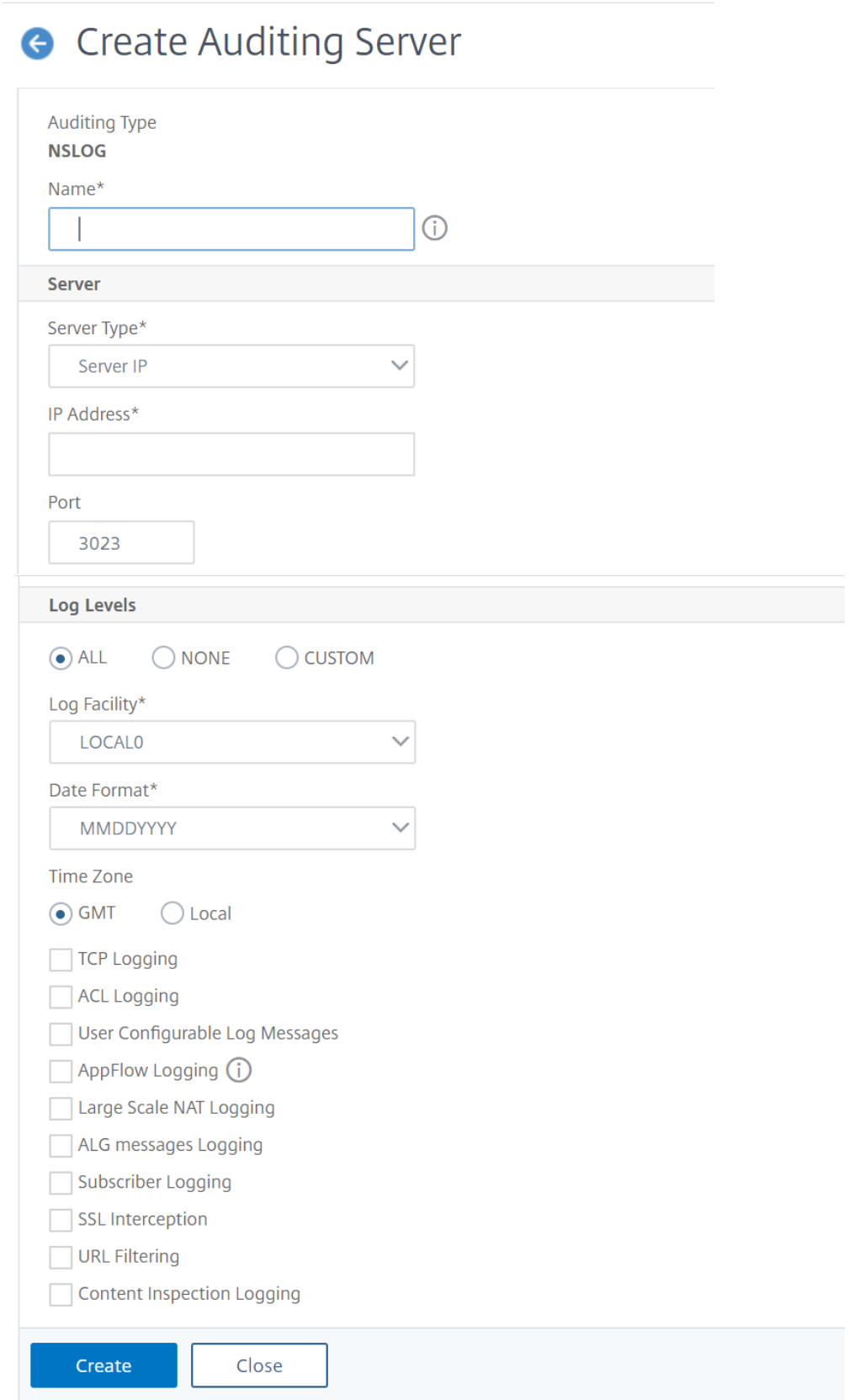
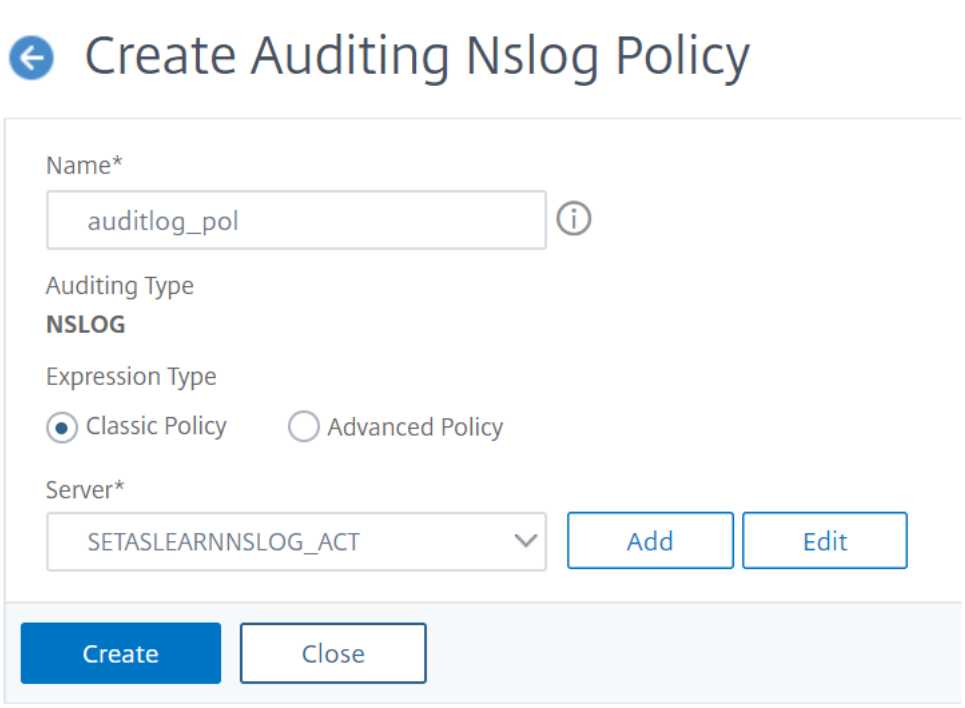

Log Message:
============
08/19/2019:08:49:46 GMT pegasus121 Informational 0-PPE-0 : default
   APPFW APPFW_JSON_SQL 6656 0 :  10.217.32.165 18402-PPE0 - profjson
   http://10.217.32.147/test.html SQL Keyword check failed for object
   value(with violation="select(;)") starting at offset(52) <blocked>
Counters:
========
   1  441083             1 as_viol_json_sql
   3       0             1 as_log_json_sql
   5       0             1 as_viol_json_sql_profile appfw__(profjson)
   7       0             1 as_log_json_sql_profile appfw__(profjson)
<!--NeedCopy-->
```

Configure JSON SQL Injection protection by using Citrix GUI

Follow the procedure below to set the JSON SQL Injection protection settings.

1. On the navigation pane, navigate to **Security** > **Profiles**.
2. In the **Profiles** page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Security Checks** under **Advanced Settings**.
4. In the **Security Checks** section, go to **JSON SQL Injection** settings.
5. Click the executable icon near the check box.



6. Click **Action Settings** to access the **JSON SQL Injection Settings** page.
7. Select the **JSON SQL Injection** actions.
8. Click **OK**.



9. In the **Citrix Web App Firewall Profile** page, click **Relaxation Rules** under **Advanced Settings**.
10. In **Relaxation Rules** section, select **JSON SQL Injection** settings and click **Edit**.



11. In the JSON SQL Injection Relaxation Rule page, enter the URL to which the request has to be sent. All requests sent to this URL will not be blocked.
12. Click **Create**.



JSON Cross-Site Scripting protection check

September 14, 2021

If an incoming JSON payload has a malicious cross-site scripting data, WAF blocks the request. The following procedures explain how you can configure this through CLI and GUI interfaces.

Configure JSON Cross-Site Scripting protection

For configure JSON cross-site scripting protection, you must complete the following steps:

1. Add application firewall profile as JSON.
2. Configure JSON cross-site scripting action to block cross-site scripting malicious payload

Add application firewall profile of type JSON

You must first create a profile that specifies how the application firewall must protect your JSON web content from JSON cross-site scripting attack.

At the command prompt, type:

```
add appfw profile <name> -type (HTML | XML | JSON)
```


> **Note**:
>
> When you set the profile type as JSON, other checks such as HTML or XML will not applicable.

Example

```
add appfw profile profile1 -type JSON
```

Sample output for JSON cross-site scripting violation

```
JSONcross-site scriptingAction: block log stats
Payload: {
 "username":"<a href=\"jAvAsCrIpT:alert(1)\">X</a>","password":"xyz" }


Log message: Aug 19 06:57:33 <local0.info> 10.106.102.21
    08/19/2019:06:57:33 GMT  0-PPE-0 : default APPFW APPFW_JSON_cross-
    site scripting 58 0 :  10.102.1.98 12-PPE0 - profjson http://
    10.106.102.24/ Cross-site script check failed for object value(with
    violation="Bad URL: jAvAsCrIpT:alert(1)") starting at offset(12). <
    blocked>

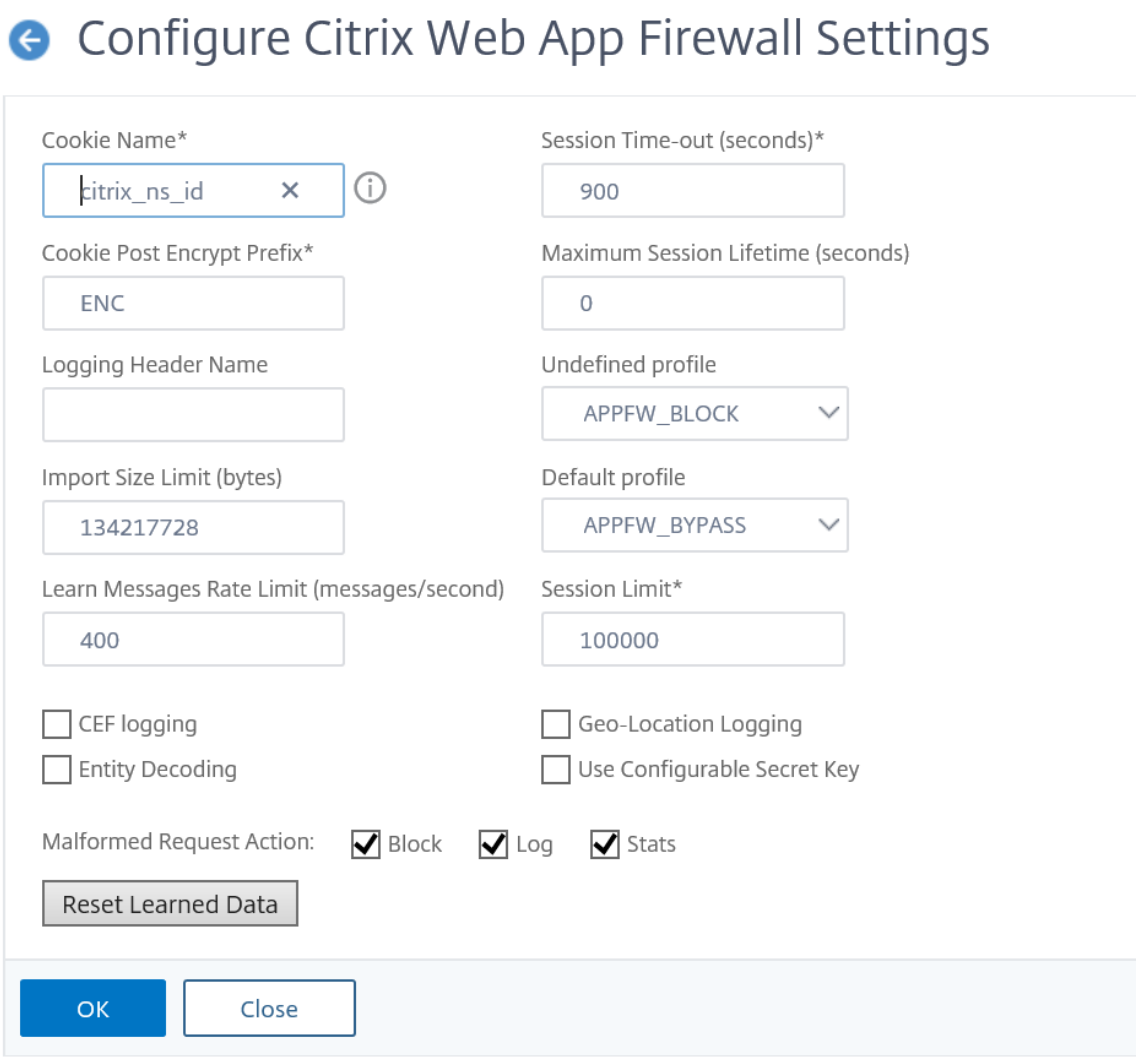
Counters
    1  357000                  1 as_viol_json_xss
    3  0                       1 as_log_json_xss
    5  0                       1 as_viol_json_xss_profile appfw__(
       profjson)
    7  0                       1 as_log_json_xss_profile appfw__(
       profjson)

<!--NeedCopy-->
```

Configure JSON Cross-Site Scripting action

You must configure one or more JSON cross-site scripting actions to protection your application from JSON Cross-Site Scripting attacks.
At the command prompt, type:

```
set appfw profile <name> - JSONcross-site scriptingAction [block] [log] [
stats] [none]
```

Example

```
set appfw profile profile1 -JSONcross-site scriptingAction block
```


The available Cross-Site Scripting actions are:
Block - Block connections that violate this security check.
Log - Log violations of this security check.
Stats - Generate statistics for this security check.
None - Disable all actions for this security check.

> **Note**:
> To enable one or more actions, type “set appfw profile - JSONcross-site scriptingAction “ followed by the actions to be enabled.

Example

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

Configure JSON Cross Site Scripting (cross-site scripting) protection by using Citrix GUI

Follow the procedure below to set the Cross Site Scripting (cross-site scripting) protection settings.

1. On the navigation pane, navigate to **Security** > **Profiles**.
2. In the **Profiles** page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Security Checks** under **Advanced Settings**.
4. In the **Security Checks** section, go to **JSON Cross-Site Scripting (cross-site scripting)** settings.
5. Click the executable icon near the checkbox.



6. Click **Action Settings** to access the **JSON Cross-Site Scripting Settings** page.
7. Select the JSON cross-site scripting actions.
8. Click **OK**.



9. In the **Citrix Web App Firewall Profile** page, click **Relaxation Rules** under **Advanced Settings**.
10. In **Relaxation Rules** section, select JSON Cross-Site Scripting settings and click **Edit**.



11. In the **JSON Cross-Site Scripting Relaxation Rule** page, click **Add** to add a JSON Cross-Site Scripting relaxation rule.
12. Enter the URL to which the request has to be sent. All requests sent to this URL will not be blocked.
13. Click **Create**.



Managing content types

September 14, 2021

Web servers add a Content-Type header with a MIME/type definition for each content type. Web servers serve many different types of content. For example, standard HTML is assigned the “text/html” MIME type. JPG images are assigned the “image/jpeg” or “image/jpg” content type. A normal web server can serve different types of content, all defined in the Content Type header by the assigned MIME/type.

Many Web App Firewall filtering rules are designed to filter a specific content type. The filtering rules apply to one type of content such as HTML and are often inappropriate when filtering a different type of content (such as images). As a result, the Web App Firewall attempts to determine the content type of requests and responses before it filters them. If a web server or browser does not add a Content-Type header to a request or response, the Web App Firewall applies a default content type and filters content accordingly.

The default content type is usually “application/octet-stream” with the most generic MIME/type definition. The MIME/type is appropriate for any content type a web server is likely to serve. But does not provide much information to the Web App Firewall to allow it to choose appropriate filtering. If a protected web server is configured to add accurate content type headers, you can then create a profile

for the web server and assign a default content type to it. This is done to improve both the speed and the accuracy of filtering.

You can also configure a list of allowed request content types for a specific profile. When this feature is configured, if the Web App Firewall filters a request that does not match one of the allowed content types, it blocks the request. After upgrade from release 10.5 to 11.0, unknown content-types which are not in the default allowed content-type list do not bind. You can add other content-types which you want to be allowed to the relaxed rules.

Requests must always be of either the "application/x-www-form-urlencoded", "multipart/form-data," or "text/x-gwt-rpc" types. The Web App Firewall blocks any request that has any other content type designated.

> **Note**
>
> You cannot include the "application/x-www-form-urlencoded" or "multipart/form-data" content types on the allowed response content types list.

To set the default request content type by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> -requestContentType <type>`
- `save ns config`

Example

The following example sets the "text/html" content type as the default for the specified profile:

```
1 set appfw profile profile1 -requestContentType "text/html"
2 save ns config
3 <!--NeedCopy-->
```

To remove the user-defined default request content type by using the command line interface

At the command prompt, type the following commands:

- `unset appfw profile <name> -requestContentType <type>`
- `save ns config`

Example

The following example unsets the default content type of “text/html” for the specified profile, allowing the type to revert to “application/octet-stream”:

```
unset appfw profile profile1 -requestContentType "text/html"
save ns config
<!--NeedCopy-->
```

> **Note**
>
> Always use last content-type header for processing and remove remaining content-type headers if any that ensures that the back-end server receives a request with only one content-type.
>
> To block requests that can be bypassed, add a Web App Firewall policy with rule as HTTP.REQ.HEADER (“content-type”).COUNT.GT(1)’ and profile as *appfw_block*.
>
> If a request is received without a Content-Type header or if the request has Content-Type header without any value, Web App Firewall applies the configured **RequestContentType** value and processes the request accordingly.

To set the default response content type by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> -responseContentType <type>`
- `save ns config`

Example

The following example sets the “text/html” content type as the default for the specified profile:

```
set appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

To remove the user-defined default response content type by using the command line interface

At the command prompt, type the following commands:

- `unset appfw profile <name> -responseContentType <type>`
- `save ns config`

Example

The following example unsets the default content type of “text/html” for the specified profile, allowing the type to revert to “application/octet-stream”:

```
unset appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

To add a content type to the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- `bind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

Example

The following example adds the “text/shtml” content type to the allowed content types list for the specified profile:

```
bind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```

To remove a content type from the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- `unbind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

Example

The following example removes the “text/shtml” content type from the allowed content types list for the specified profile:

```
unbind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```


Manage urlencoded and multipart-form content types

The Citrix ADC Web App Firewall now enables you to configure Urlencoded and Multipart-Form content types for forms. The content type configuration is similar to XML and JSON list. Based on the configuration, Web App Firewall classifies the requests and inspects for urlencoded or multipart-form content type.

To configure Web App Firewall profile with Urlencoded and Multipart-Form content types
At the command prompt, type:

```
bind appfw profile p2 -contentType <string>
```

Example:

```
bind appfw profile p2 -contentType UrlencodedFormContentType

bind appfw profile p2 -ContentType appfwmultipartform
```

To manage the default and allowed content types by using the GUI

1. Navigate to **Security** > **Web App Firewall** > **Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**. The **Configure Web App Firewall Profile** dialog box is displayed.
3. The **Configure Web App Firewall Profile** dialog box, click the **Settings** tab.
4. On the **Settings** tab, scroll down about halfway to the Content Type area.
5. In the Content Type area, configure the default request or response content type:
   - To configure the default request content type, type the MIME/type definition of the content type you want to use in the Default Request text box.
   - To configure the default response content type, type the MIME/type definition of the content type you want to use in the Default Response text box.
   - To create a new allowed content type, click **Add**. The **Add Allowed Content Type** dialog box is displayed.
   - To edit an existing allowed content type, select that content type, and then click **Open**. The **Modify Allowed Content Type** dialog box is displayed.
6. To manage the allowed content types, click Manage Allowed Content Types.
7. To add a new content type or modify an existing content type, click Add or Open, and in the **Add Allowed Content Type** or **Modify Allowed Content Type** dialog box, do the following steps.
   a) Select/clear the Enabled check box to include the content type in, or exclude it from, the list of allowed content types.
   b) In the Content Type text box, type a regular expression that describes the content type that you want to add, or change the existing content type regular expression.
   Content types are formatted exactly as MIME type descriptions are.

> **Note:**
> You can include any valid MIME type on the allowed contents type list. Since many types of document can contain active content and therefore can potentially contain malicious content, you must exercise caution when adding MIME types to this list.

   c) Provide a short description that explains the reason for adding this particular MIME type to the allowed contents type list.
   d) Click **Create** or **OK** to save your changes.
8. Click **Close** to close the Manage Allowed Content Types dialog box and return to the **Settings** tab.
9. Click **OK** to save your changes.

To manage Urlencoded and Multipart-form content types by using the Citrix ADC GUI

1. Navigate to **Security** > **Web App Firewall** > **Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**.
3. In the **Configure Web App Firewall Profile** page, select the **Profile Settings** in the **Advanced Settings** section.
4. Under **Inspected Content Type** section, set the following parameters:
   a) application/x-www-form-urlencoded. Select the checkbox to inspect Urlencoded content type.
   b) multipart/form-data. Select the check to inspect Multipart-form content type.
5. Click **OK**.



Profiles

September 14, 2021

A profile is a collection of security settings that are used to protect specific types of web content or specific parts of your website. In a profile, you determine how the Web App Firewall applies each of its filters (or checks) to requests to your websites, and responses from them. The Web App Firewall supports two types of profile: four built-in (default) profiles that do not require further configuration, and user-defined profiles that do require further configuration.

Built-in profiles

The four Web App Firewall built-in profiles provide simple protection for applications and websites that either do not require protection, or that must not be directly accessed by users at all. These profile types are:

- **APPFW_BYPASS.** Skips all Web App Firewall filtering and sends the unmodified traffic to the protected application or website, or to the client.

- **APPFW_RESET.** Resets the connection, requiring that the client re-establish his or her session by visiting a designated start page.
- **APPFW_DROP.** Drops all traffic to or from the protected application or website, and sends no response of any kind to the client.
- **APPFW_BLOCK.** Blocks traffic to or from the protected application or website.

You use the built-in profiles exactly as you do user-defined profiles, by configuring a policy that selects the traffic to which you want to apply the profile and then associating the profile with your policy. Since you do not have to configure a built-in policy, it provides a quick way to allow or block specified types of traffic or traffic that is sent to specific applications or websites.

User-defined profiles

User-defined profiles are profiles that are build and configured by users. Unlike the default profiles, you must configure a user-defined profile before it will be of use filtering traffic to and from your protected applications.

There are three types of user-defined profile:

- **HTML.** Protects HTML-based web pages.
- **XML.** Protects XML-based web services and websites.
- **Web 2.0.** Protects Web 2.0 content that combines HTML and XML content, such as ATOM feeds, blogs, and RSS feeds.

The Web App Firewall has a number of security checks, all of which can be enabled or disabled, and configured in a number of ways in each profile. Each profile also has a number of settings that control how it handles different types of content. Finally, rather than manually configuring all of the security checks, you can enable and configure the learning feature. This feature observes normal traffic to your protected websites for a period of time, and uses those observations to provide you with a tailored list of recommended exceptions (*relaxations*) to some security checks, and additional rules for other security checks.

During initial configuration, whether by using the Web App Firewall Wizard or manually, you normally create one general purpose profile to protect all content on your websites that is not covered by a more specific profile. After that, you can create as many specific profiles as you want to protect more specialized content.

The Profiles pane consists of a table that contains the following elements:

Name. Displays all the Web App Firewall profiles configured in the appliance.

Bound signature. Displays the signatures object that is bound to the profile in the previous column, if any.

Policies. Displays the Web App Firewall policy that invokes the profile in the leftmost column of that row, if any.

Comments. Displays the comment associated with the profile in the leftmost column of that row, if any.

Profile Type. Displays the type of profile. Types are Built-In, HTML, XML, and Web 2.0.

Above the table is a row of buttons and a drop-down list that allow you to create, configure, delete, and view information about your profiles:

- **Add.** Add a new profile to the list.
- **Edit.** Edit the selected profile.
- **Delete.** Delete the selected profile from the list.
- **Statistics.** View the statistics for the selected profile.
- **Action.** Drop-down list that contains additional commands. Currently allows you to import a profile that was exported from another Web App Firewall configuration.

Creating Web App Firewall profiles

September 14, 2021

You can create a Web App Firewall profile in one of two ways: by using the command line, and by using the GUI. Creating a profile by using the command line requires that you specify options on the command line. The process is similar to that of configuring a profile, and with a few exceptions the two commands take the same parameters.

Creating a profile by using the GUI requires that you specify only two options. You specify basic or advanced *defaults*, the default configuration for the various security checks and settings that are part of a profile, and choose the profile *type* to match the type of content that the profile is intended to protect. You can also, optionally, add a comment. After you create the profile, you must then configure it by selecting it in the data pane, and then clicking **Edit**.

If you plan to use the learning feature or to enable and configure many advanced protections, you must choose advanced defaults. In particular, if you plan to configure either of the SQL injection checks, either of the cross-site scripting checks, any check that provides protection against Web form attacks, or the cookie consistency check, you must plan to use the learning feature. Unless you include the proper exceptions for your protected websites when configuring these checks, they can block legitimate traffic. Anticipating all exceptions without creating any that are too broad is difficult. The learning feature makes this task much easier. Otherwise, basic defaults are quick and must provide the protection that your web applications need.

There are three profile types:

- **HTML.** Protects standard HTML-based websites.
- **XML.** Protects XML-based web services and websites.

- **Web 2.0 (HTML XML).** Protects websites that contain both HTML and XML elements, such as ATOM feeds, blogs, and RSS feeds.

There are also a few restrictions on the name that you can give to a profile. A profile name cannot be the same as the name assigned to any other profile or action in any feature on the NetScaler appliance. Certain action or profile names are assigned to built-in actions or profiles, and can never be used for user profiles. A complete list of disallowed names can be found in the Web App Firewall Profile Supplemental Information. If you attempt to create a profile with a name that has already been used for an action or a profile, an error message is displayed and the profile is not created.

To create a Web App Firewall profile by using the command line interface

At the command prompt, type the following commands:

- `add appfw profile <name> [-defaults ( **basic** | **advanced** )]`
- `set appfw profile <name> -type ( **HTML** | **XML** | **HTML XML** )`
- `set appfw profile <name> -comment "<comment>"`
- `save ns config`

Example

The following example adds a profile named pr-basic, with basic defaults, and assigns a profile type of HTML. This is the appropriate initial configuration for a profile to protect an HTML website.

```
1 add appfw profile pr-basic -defaults basic -comment "Simple profile for
      websites."
2 set appfw profile pr-basic -type HTML
3 save ns config
4 <!--NeedCopy-->
```

To create a Web App Firewall profile by using the GUI

Complete the following procedure to create a Web App Firewall profile:

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, click **Add**.
3. In the **Create Web App Firewall Profile** page, set the following basic parameters:
   a) Name
   b) Profile Type
   c) Comments
   d) Defaults

e) Description

4. Click **OK**.
5. In the **Advanced Settings** section, complete the following configurations:

   a) Security Checks
   b) Profile Settings
   c) Dynamic Profiling
   d) Relaxation Rules
   e) Deny Rules
   f) Learned Rule
   g) Extended Logging



6. In the **Security Checks** section, select a security protection and click Action Settings.
7. In the security check page, set the parameters.

   > **Note:**
   > The **Active Rule** setting is available only for **HTML SQL Injection** check to allow > or deny signature rules.

8. Click **OK** and **Close**.
9. In the **Profile Settings** section, set the profile parameters. For more information, see Configure Web App Firewall Profile settings topic.
10. In the **Dynamic Profiling** section, select a security check to add dynamic profile settings. For more information, see Dynamic Profile topic.
11. In the **Relaxation Rules** section, click **Edit** to add a relaxation rule for a security check. For more information, see Relaxation Rule for details.
12. In the **Deny Rules** section, add a deny rule for the HTML SQL Injection check. For more information, see HTML Deny Rules topic.
13. In the **Learnt Rule** section, set the learning settings. For more information, see Web App Firewall Learning topic.

14. In the **Extended logging** section, click **Add** for masking sensitive data. For more information, see Extended logging topic.
15. Click **Done**, and then click **Close**.



Enforce HTTP RFC compliance

September 14, 2021

Citrix Web App Firewall inspects the incoming traffic for HTTP RFC compliance and drops any request that has RFC violations by default. However, there are certain scenarios, where the appliance might have to bypass or block a non-RFC compliance request. In such cases, you can configure the appliance to bypass or block such requests at global or profile level.

Block or bypass non-RFC compliant requests at global level

The HTTP module tags a request as invalid if it is incomplete or invalid and such requests cannot be processed by WAF. For example, an incoming HTTP request having a host header missing. To block or bypass such invalid requests, you must configure the `malformedReqAction` option in the application firewall global settings.

> **Note:**
>
> If you disable the block option in the `malformedReqAction` parameter, the appliance bypasses the entire app firewall processing for all non-RFC compliance requests and forwards the requests to the next module.

To block or bypass invalid non-RFC complaint HTTP requests by using the command line interface

To block or bypass invalid requests, enter the following command:

```
set appfw settings -malformedreqaction <action>
```

Example:

```
set appfw settings -malformedReqAction block
```

To display malformed request action settings

To display malformed request action settings, enter the following command:

```
show appfw settings
```

Output:

```
1 DefaultProfile: APPFW_BYPASS UndefAction: APPFW_BLOCK SessionTimeout:
      900    LearnRateLimit: 400    SessionLifetime: 0
    SessionCookieName: citrix_ns_id ImportSizeLimit: 134217728
    SignatureAutoUpdate: OFF SignatureUrl:"https://s3.amazonaws.com/
    NSAppFwSignatures/SignaturesMapping.xml" CookiePostEncryptPrefix:
    ENC GeoLocationLogging: OFF CEFLogging: OFF      EntityDecoding:
     OFF    UseConfigurableSecretKey: OFF SessionLimit: 100000
    MalformedReqAction: block log stats
2  Done
3 <!--NeedCopy-->
```

To block or bypass invalid non-RFC complaint HTTP requests by using the Citrix ADC GUI

1. Navigate to **Security** > **Citrix Web App Firewall**.
2. In the **Citrix Web App Firewall** page, click **Change Engine Settings** under **Settings**.
3. In the **Configure Citrix Web App Firewall Settings** page, select the **Log Malformed Request** option as Block, Log, or Stats.
4. Click **OK** and **Close**.

> **Note:**
>
> If you unselect the block action or do not select any malformed request action, the appliance bypasses the request without intimating the user.

Block or bypass non-RFC compliant requests at profile level

Other non-RFC compliant requests can be configured to block or bypass at profile level. You must configure the RFC profile either in Block or Bypass mode. By doing this, any invalid traffic that matches the Web App Firewall profile is either bypassed or blocked accordingly.

> **Note:**
>
> When you set the RFC profile in "Bypass" mode, you must make sure you disable the transformation option in the **HTML Cross-Site Scripting Settings** and in the **HTML SQL Injection Settings** sections. If you enable the option and set the rfc profile in Bypass mode, the appliance displays a warning message, "Transform cross-site scripts" and "Transform SQL special characters" are both currently ON. Recommend turning it off when used with `APPFW_RFC_BYPASS`.
>
> **Important:**
>
> Also, the appliance displays a warning note, "Appfw Security checks enabled might not be applicable to requests which violates RFC checks when this profile is set. Enabling any transformation setting is not recommended as requests might be partially transformed that contains RFC violations."

To configure an RFC profile in the Web App Firewall profile by using the command line interface

At the command prompt, type the following commands:

```
set appfw profile <profile_name> -rfcprofile <rfcprofile_name
```

Example

```
set appfw profile P1 -rfcprofile APPFW_RFC_BLOCK
```

> **Note:**
>
> By default, the rfc profile is bound to the Web App Firewall profile in Block mode.

To configure an RFC profile in the Web App Firewall profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the **Profiles** page, select a profile and click **Edit**.

3. In the **Web App Firewall Profile** page, click **Profile settings** from the **Advanced Settings** section.
4. In the **HTML Settings** section, set the RFC profile in APPFW_RFC_BYPASS mode.
   The system displays a warning message, “Appfw Security checks enabled might not be applicable to requests which violates RFC checks when this profile is set. Enabling any transformation setting is not recommended as requests might be partially transformed that contains RFC violations”.

Configuring Web App Firewall profiles

September 14, 2021

To configure a user-defined Web App Firewall profile, first configure the security checks, which are called *deep protections* or *advanced protections* in the Web App Firewall wizard. Certain checks require configuration if you are to use them at all. Others have default configurations that are safe but limited in scope; your websites might need or benefit from a different configuration that takes advantage of more features of certain security checks.

After you have configured the security checks, you can also configure some other settings that control the behavior, not of a single security check, but the Web App Firewall feature. The default configuration is sufficient to protect most websites, but you must review them to make sure that they are right for your protected websites.

> **Note:**
>
> The profile name length and all import object name length can be set up to 127 characters.

For more information about the Web App Firewall security checks, see Advanced Protections.

To configure an Web App Firewall profile by using the command line

At the command prompt, type the following commands:

- `set appfw profile <name> <arg1> [<arg2> ...]`

  where:

  - `<arg1>` = a parameter and any associated options.
  - `<arg2>` = a second parameter and any associated options.
  - … = additional parameters and options.

  For descriptions of the parameters to use when configuring specific security checks, see Advanced Protections.

- `save ns config`

Example

The following example shows how to enable blocking for the HTML SQL Injection and HTML Cross-Site Scripting checks in a profile named pr-basic. This command enables blocking for those actions while making no other changes to the profile.

```
set appfw profile pr-basic -crossSiteScriptingAction block -
    SQLInjectionAction block
<!--NeedCopy-->
```

Bind relaxation rule to a Web App Firewall profile

When Web App Firewall detects a violation, the user has the ability to bypass the action applied through relaxation rules. Relaxation rule is an exception applied to the detected security violation. For example, the Start URL relaxation rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-site scripting can also be easily detected.

To bind security exemption or relaxation rules by using the CLI

At the command prompt, type:

```
bind appfw profile <name> ((-startURL <expression> [-resourceId  <
    string>]) | -denyURL <expression> | (-fieldConsistency <string>   <
    formActionURL> [-isRegex ( REGEX | NOTREGEX )]) | (-
    cookieConsistency <string> [-isRegex ( REGEX | NOTREGEX )]) | (-
    SQLInjection <string> <formActionURL> [-isRegex ( REGEX | NOTREGEX )
    ] [-location <location>] [-valueType <valueType> <valueExpression
    >....
<!--NeedCopy-->
```

To bind security exemption or relaxation rules by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select a profile and click **Edit**.
3. In the **Citrix Web App Firewall Profile** page, click **Relaxation Rules** from the **Advanced Setting** section.
4. In the **Relaxation Rules** section, click **StartURL** and click **Edit**.
5. In the **Start URL Relaxation Rules** page, click **Add**.

6. In the **Start URL Relaxation Rule** page, set the following parameters:
   a) Enabled. Select the checkbox to enable the relaxation rule
   b) Start URL. Enter the regular expression value
   c) Comments. Provide a short description about the relaxation rule.
7. Click **Create** and **Close**.



To configure an Web App Firewall profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**.
3. In the **Configure Web App Firewall Profile** dialog box, on the **Security Checks** tab, configure the security checks.
   - To enable or disable an action for a check, in the list, select or clear the check box for that action.
   - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.

You can also select a check and, at the bottom of the dialog box, click Open to display the **Configure Relaxation** dialog box or **Configure Rule** dialog box for that check. These dialog boxes also vary from check to check. Most of them include a **Checks** tab and a **General** tab. If the check supports relaxations or user-defined rules, the **Checks** tab includes an **Add** button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click **Open** to modify it.

- To review learned exceptions or rules for a check, select the check, and then click **Learned Violations**. In the **Manage Learned Rules** dialog box, select each learned exception or rule in turn.
  - To edit the exception or rule, and then add it to the list, click **Edit & Deploy**.
  - To accept the exception or rule without modification, click **Deploy**.
  - To remove the exception or rule from the list, click **Skip**.
- To refresh the list of exceptions or rules to be reviewed, click **Refresh**.
- open the Learning Visualizer and use it to review learned rules, click **Visualizer**.
- review the log entries for connections that matched a check, select the check, and then click Logs. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see Logs, Statistics, and Reports.
- To completely disable a check, in the list, clear all of the check boxes to the right of that check.

4. On the **Settings** tab, configure the profile settings.
   - To associate the profile with the set of signatures that you previously created and configured, under Common Settings, choose that set of signatures in the **Signatures** drop-down list.

     > **Note:**
     >
     > You may must use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.

   - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.

> **Note:**
>
> You must first upload the error object that you want to use in the Imports pane. For more information about importing error objects, see Imports.

- To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types. »More….

5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile, as described in Configuring and Using the Learning Feature.
6. Click **OK** to save your changes and return to the **Profiles** pane.

Web Application Firewall profile settings

September 14, 2021

Following are the profile settings that you must configure on the appliance.

At the command prompt, type:

```
add appfw profile <name> [-invalidPercentHandling <invalidPercentHandling
>] [-checkRequestHeaders ( ON | OFF )] [-URLDecodeRequestCookies ( ON |
 OFF )] [-optimizePartialReqs ( ON | OFF )] [-errorURL <expression>] [-
logEveryPolicyHit ( ON | OFF )] [-stripHtmlComments <stripHtmlComments>] [-
stripXmlComments ( none | all )] [-postBodyLimitSignature <positive_integer
>][-fileUploadMaxNum <positive_integer>][-canonicalizeHTMLResponse ( ON
| OFF )][-percentDecodeRecursively ( ON | OFF )] [-multipleHeaderAction
<multipleHeaderAction> ...][-inspectContentTypes <inspectContentTypes>
...][-semicolonFieldSeparator ( ON | OFF )]
```

Example:

```
add appfw profile profile1 [-invalidPercentHandling secure_mode] [-checkRequestHeaders
 ON] [-URLDecodeRequestCookies OFF] [-optimizePartialReqs OFF]
```

Where,

invalidPercentHandling. Configure the method for handling percent-encoded names and values.

Available settings function as follows:

asp_mode - Strips and Parses Invalid Percent for Parsing. Example:- `curl -v "http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)` is stripped of and the rest of the content is inspected and action taken for the SQLInjection check.

secure_mode - We detect the Invalid Percent coded value and ignore it. Example:- `curl -v "http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)` is detected, counters are incremented and content is passed as is to the server.
apache_mode - This mode works similar to secure mode.
Possible values: apache_mode, asp_mode, secure_mode
Default value: secure_mode

optimizePartialReqs. When OFF/ON (without safe object), a Citrix ADC appliance sends the partial request to the back-end server. This partial response sent back to the client. OptimizePartialReqs makes sense when the Safe object is configured. The appliance sends requests for full response from the server when OFF, requests only partial response when ON.

Available settings are as follows:

ON - Partial requests by the client result in partial requests to the back-end server.
OFF - Partial requests by the client are changed to full requests to the back-end server
Possible values: ON, OFF
Default value: ON

URLDecodeRequestCookies. URL Decode request cookies before subjecting them to SQL and cross-site scripting checks.

Possible values: ON, OFF
Default value: OFF

Signature Post Body Limit (Bytes). Limits the request payload (in bytes) inspected for signatures with the location specified as 'HTTP_POST_BODY'.

Default value: 8096
Minimum value: 0
Maximum Value: 4294967295

Post Body Limit (Bytes). Limits the request payload (in bytes) inspected by Web Application Firewall.

Default value: 20000000
Minimum value: 0
Maximum Value: 10 GB

For more information about the Security setting and its GUI procedure, see Configure Web App Firewall Profile topic.

postBodyLimitAction. PostBodyLimit honors error settings when you specify the maximum size of HTTP body to be allowed. To honor error settings you must configure one or more Post Body Limit actions. The configuration is also applicable for requests where the transfer encoding header is chunked.

```
set appfw profile <profile_name> -PostBodyLimitAction block log stats
```


Where,
Block - This action blocks connection that violates the security check and it is based on the maximum size of the configured HTTP body (post body limit). You must always enable the option.

Log - Log violations of this security check.

Stats - Generate statistics for this security check.

> **Note:**
>
> The log format for post body limit action is now changed to follow the standard audit logging format, for example:
>
> ```
> ns.log.4.gz:Jun 25 1.1.1.1. <local0.info> 10.101.10.100 06/25/2020:10:10:28 GMT 0-PPE-0 : default APPFW APPFW_POSTBODYLIMIT 1506 0 : <Netscaler IP> 4234-PPE0 - testprof ><URL> Request post body length(<Post Body Length >)exceeds post body limit.
> ```

inspectQueryContentTypes Inspect request query and web forms for injected SQL and cross-site scripts for the following content types.

```
set appfw profile p1 -inspectQueryContentTypes HTML XML JSON OTHER
```

Possible values: HTML, XML, JSON, OTHER

By default, this parameter is set as “InspectQueryContentTypes: HTML JSON OTHER” for both basic and advanced appfw profiles.

Example for inspect query content type as XML:

```
1 > set appfw profile p1 -type XML
2 Warning: HTML, JSON checks except “InspectQueryContentTypes” & “
     Infer Content-Type XML Payload Action” will not be applicable when
     profile type is not HTML or JSON respectively.
3 <!--NeedCopy-->
```

Example for inspect query content type as HTML:

```
1 > set appfw profile p1 -type HTML
2 Warning: XML, JSON checks except “InspectQueryContentTypes” & “Infer
      Content-Type XML Payload Action” will not be applicable when
     profile type is not XML or JSON respectively
3 Done
4 <!--NeedCopy-->
```

Example for inspect query content type as JSON:

```
1 > set appfw profile p1 -type JSON
```



```
Warning: HTML, XML checks except “InspectQueryContentTypes” & “Infer
    Content-Type XML Payload Action will not be applicable when profile
    type is not HTML or XML respectively
Done
<!--NeedCopy-->
```

errorURL expression. The URL that the Citrix Web App Firewall uses as an error URL. Maximum Length: 2047.

> **Note:**
>
> For blocking violations in a requested URL, if the error URL is similar to the signature URL the appliance resets the connection.

logEveryPolicyHit - Log every profile match, regardless of security checks results.
Possible values: ON, OFF.
Default value: OFF.

stripXmlComments - Strip XML comments before forwarding a webpage sent by a protected website in response to a user request.
Possible values: none, all, exclude_script_tag.
Default value: none

postBodyLimitSignature - Maximum allowed HTTP post body size for signature inspection for location HTTP_POST_BODY in the signatures, in bytes.
The changes in value can impact CPU and latency profile.
Default value: 2048.
Minimum value: 0
Maximum Value: 4294967295

fileUploadMaxNum - Maximum allowed number of file uploads per form-submission request. The maximum setting (65535) allows an unlimited number of uploads.
Default value: 65535
Minimum value: 0
Maximum value: 65535

canonicalizeHTMLResponse - Perform HTML entity encoding for any special characters in responses sent by your protected websites.
Possible values: ON, OFF
Default value: ON

percentDecodeRecursively - Configure whether the application firewall should use percentage recursive decoding.
Possible values: ON, OFF
Default value: ON

multipleHeaderAction - One or more multiple header actions. Available settings function as follows:

- Block. Block connections that have multiple headers.
- Log. Log connections that have multiple headers.
- KeepLast. Keep only the last header when multiple headers are present.

inspectContentTypes – One or more InspectContentType lists.

- application/x-www-form-urlencoded
- multipart/form-data
- text/x-gwt-rpc

Possible values: none, application/x-www-form-urlencoded, multipart/form-data, text/x-gwt-rpc

semicolonFieldSeparator - Allow ';' as a form field separator in URL queries and POST form bodies.
Possible values: ON, OFF
Default value: OFF

Changing an Web App Firewall profile type

September 14, 2021

If you chose the wrong profile type for an Web App Firewall profile, or the type of content on the protected website has changed, you can change the profile type.

> **Note** When you change the profile type, you lose all configuration settings and learned relaxations or rules for the features that the new profile type does not support. For example, if you change the profile type from Web 2.0 to XML, you lose any configuration options for Start URL, Form Field Consistency Check, and the other HTML-specific security checks. The configuration for any options that is supported by both the old and the new profile types remains unchanged.

To change an Web App Firewall profile type by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> -type ( **HTML** | **XML** | **HTML XML** )`
- `save ns config`

Example

The following example changes the type of a profile named pr-basic, from HTML to HTML XML, which is equivalent to the Web 2.0 type in the GUI.


```
set appfw profile pr-basic -type HTML XML
save ns config
<!--NeedCopy-->
```

To change an Web App Firewall profile type by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Policies**.
2. In the details pane, click **Action**, and then **Change Profile Type**.
3. In the **Change Web App Firewall Profile Type** dialog box, **Profile Type** drop-down list, select a new profile type.
4. Click **OK** to save your changes and return to the **Profiles** pane.

Exporting and importing an Web App Firewall profile

September 14, 2021

You can replicate the entire configuration of an Web App Firewall profile (including all bound objects, such as HTML error object, XML error object, WSDL or XML schema, signatures, and so on) across multiple appliances. You can select a target profile and export the configuration to save it in your computer's local file system, or you can transfer the archived configuration to store it on a server. Similarly, you can browse your computer's local file system or import the archive from the server to select a previously exported profile and import it into your NetScaler appliance.

The option to export the entire profile configuration and then import it into another appliance can be useful in various use cases. For example, you might want to configure an Web App Firewall profile in a test bed set-up to test and validate that it is working as expected. Once you are satisfied, you can export the profile and import the profile configuration to your production NetScaler appliances. This functionality is also useful for backing up your configuration. You can export the profile before making changes, so that you can easily roll back the configuration to a known state if necessary.

> **Note**
>
> Web App Firewall profiles that are exported and archived from one build cannot be restored to a system running a different build, because changes introduced in the newer releases can lead to compatibility issues. If you attempt to restore an archived profile to a different build than the one from which it was exported, an error message is logged in ns.log.

The export and import profile functionality is available in both the GUI (GUI) and the command line interface (CLI). The GUI is recommended, because it offers easy to use **Action** options. With a click of a button you can **Export** or **Import** the entire configuration of a profile.

Exporting Web App Firewall profiles with the CLI

If you use CLI to **export** a profile, you must **archive** the configuration and then **export** it. To **import** a profile, you must **import** the archive into the NetScaler appliance and then run the **restore** command to extract the configuration. The following set of CLI commands can be used for exporting, importing and managing the profile configurations.

CLI commands to export archives:

- `archive appfw profile <name> <archivename> [-comment <string>]`
- `export appfw archive <name> <target>`

CLI commands to import archives:

- `import appfw archive <src> <name> [-comment <string>]`
- `restore appfw profile <archivename>`

CLI commands to manage archives:

- `show appfw archive`
- `rm appfw archive <name>`

Exporting a profile from one appliance and importing it to another requires five steps in CLI. The first 3 steps are performed on the source appliance on which profile configuration is originally created, and the next 2 steps are performed on the target appliance on which the profile configuration is to be replicated.

Export profile from the source NetScaler appliance:

Step 1: Create an archive of the configured profile.

Step 2: Export the archive to the NetScaler file system.

Step 3: Use a file transfer utility such as scp to transfer the exported archive file from NetScaler appliance A to the target NetScaler appliance.

Import profile to the target NetScaler appliance:

Step 4: Run the import command to import the archived file. You can import the archive from your NetScaler's local file system, or you can use HTTP or HTTPS protocol to import the archive from a server by using the URL.

Step 5: Run the restore command to restore the profile configuration from the imported archive

To export an Web App Firewall profile by using the command line interface:

First, **archive** the profile's configuration, and then **export** the archive to a target location. At the command prompt, type the following commands:

```
archive appfw profile <profileName> <archiveName>
```

where:

- <profileName> is the name of the profile to archive.
- <archiveName> is the name of the archive file to create.

Execution of the above command creates 2 instances of the archive file. One in /var/tmp folder and another in the /var/archive/appfw folder.

```
export appfw archive <archiveName> <target>
```

where:

- <archiveName> is the name of the archive to export. (The same name as in the previous command.
- <target> is a file path starting with local: as the prefix, followed by <archiveName>.

Execution of the export command saves the exported archive file on the file system of your NetScaler appliance in the /var/tmp folder.

Examples:

```
> archive appfw profile test_pr archived_test_pr
```

```
> export appfw archive archived_test_pr local:dutA_test_pr
```

After the above two commands are run, the /var/tmp folder contains the archived_test_pr file and the exported copy, dutA_test_pr, which are identical in size. From the CLI, you can drop into the shell to navigate to the folder to verify that these files are there.

After exporting the archive file, you can use **scp** or some other such file transfer utility to transfer a copy of the archive file from the NetScaler appliance on which they were created to your target NetScaler appliance.

Importing Web App Firewall Profiles with the CLI

After you have successfully scp'd the archived file from the source appliance to the target appliance, you are ready to **import** the profile's archive, and then run the **restore** command to replicate the profile's configuration on the target appliance.

Log onto the target appliance. Drop into the shell and cd to the /var/tmp folder to verify that the size of the scp'd file on this appliance matches the size of the original archived file on the source appliance. Exit the shell to return to the command line.

To import a profile by using the CLI:

At the command prompt, type the following commands:

```
import appfw archive <src> <name> [-comment <string>]
```

where

- <src> is the location of the archive file after it has been transferred from the source appliance on which it was created. You can use a local file system and file name. If you have placed the archive on a server, you can use a URL to import the archived file. If the path or file name contains spaces, enclose the URL in straight double quotation marks.
- <name> is the name of the archive file to be imported.
- <string> is an optional description of the purpose of the Archive.

```
restore appfw profile <archiveName>
```

Examples:

A. Import from local file followed by restore:

```
> import appfw archive local:dutA_test_pr dut2_test_pr
```

```
> restore appfw profile dut2_test_pr
```

B. Import from URL followed by restore:

```
import appfw archive http://10.217.30.16/FFC/Profile_ImportExport/
dutA_test_pr.tgz my_archive
restore appfw profile my_archive
```

This example restores the test_pr profile along with all bound objects (such as signatures, html error page, relaxation rules and so on) on the target NetScaler appliance.

You can use the following CLI commands to access man pages for additional details.

- man archive appfw profile
- man export appfw archive
- man import appfw archive
- man restore appfw profile
- man show appfw archive
- man rm appfw archive

Exporting and Importing Web App Firewall Profiles with the GUI

The GUI is easier to use than the CLI. The utility performs both archive and export operations when you click **Export**. Similarly it runs both import and restore when you click **Import**. The GUI can access the local file system of the computer from which you access the utility. You can export a copy of the archive and save it on your local computer. You can then import this copy directly in the target appliance without having to manually transfer the archive file from one appliance to the other(s).

To export an Web App Firewall profile by using the GUI:

1. Navigate to **Configuration** > **Security** > **Web App Firewall** > **Profiles**.
2. In the details pane, select a profile to export. Click **Actions** and select **Export** to download and save a copy in your computer's local file system.

To import an Web App Firewall profile by using the GUI:

1. Navigate to **Configuration** > **Security** > **Web App Firewall** > **Profiles**.
2. In the details pane, click **Actions** and select **Import**. In the Import Web App Firewall Profile pane, the Import From* selection box gives you 2 options:

URL: You can choose to import an archive by specifying a **URL**. When this option is selected, you must provide an absolute path for the archived file in the **URL** input box.

File: You can choose to import an archive from the local **File**. When this option is selected, a **Local File** selection field is displayed. You can browse your computer's local files to select the target archive file.

Click **Create** to import the specified archive. Successful completion of the import operation creates the profile configuration on the target appliance.

Highlights

- You can replicate the entire configuration (including all import objects as well as configured relaxation rules for the profile) on multiple appliances, without needing to repeat configuration steps, by using export and import profile functionality.
- The imported objects, such as signatures, WSDL, Schema, error page and so on, are included in the archived tar file and replicated on the target appliance.
- Customized field types are included in the archived tar file and replicated on the target appliance.
- The policy bindings of the archived profile are not replicated when the configuration is restored. You must configure the policy and bind it to the profile after importing the profile to the appliance.
- The name of the archive file can be up to 31 character long. As with profile names, an archive name must begin with an alphanumeric character or underscore and contain only alphanumeric and underscore (_), number (#), period (.), space ( ), colon (:), at (@), equals (=) or hyphen (-) characters.
- Comments associated with the archive must be descriptive enough to convey the purpose of the archived configuration. The maximum allowed length for a comment is 255 characters.
- The `clear config -force basic` command does not remove the archived profiles.
- The import and export profile functionality is supported in high availability (HA) deployments.

Debugging Tips

- Monitor the /var/log/ns.log during command executions to see if there are any ERROR messages.

- Additional logs (_restore.log, remove.log, import.log) are generated in the /var/tmp/folder. They can help debug issues during the corresponding operations. When these logs reach one MB in size, the log messages are purged to shrink the log file to one fourth of the original size.
- If the import command fails when you are using the URL option instead of the local file system, verify that DNS name server and route settings are accurately configured.
- If you are using the HTTPS protocol to import the archive, the command might fail if the HTTPS server requires client certificate authentication.

Ease of troubleshooting with Web Application Firewall logs

September 14, 2021

When there is a security attack, it is important to capture detailed WAF logging on the appliance. For this, you can configure the “VerboseLogLevel” parameter on an Application Firewall profile.

Consider a web traffic having a security attack. When the appliance receives the traffic, violation details such as HTTP header details, log pattern, and pattern payload information are logged and sent to the ADM server. The ADM server monitors the detailed logs and displays it on the Security Insight page for monitoring and tracking purpose.

Configuring verbose log level by using the command interface

To capture detailed WAF logs, configure the following command.
At the command interface, type:

```
set appfw profile <profile_name> -VerboseLogLevel (pattern|patternPayload|
patternPayloadHeader)
```

Example

```
set appfw profile profile1 -VerboseLogLevel patternPayloadHeader
```

The available log levels are:

1. Pattern. Logs only violation pattern.
2. Pattern payload. Logs violation pattern and 150 bytes of extra field element payload.
3. Pattern payload header. Logs violation patter, 150 bytes of extra field element payload and HTTP header information.

Configuring verbose log level by using the Citrix ADC GUI

Follow the procedure below to configure the verbose log level in the WAF profile.

1. On the navigation pane, navigate to **Security** > **Profiles**.
2. In the **Profiles** page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Profile Settings** under **Advanced Settings**.
4. In the **Profile Settings** section, select the detailed WAF log level in the Verbose Log Level field.
5. Click **OK** and **Done**.



File upload protection

September 14, 2021

Many attackers try to upload malicious code, virus, or malware as file attachments during multi-form submission. It is important to protect our network and overcome such threats. To prevent such malicious file uploads, a Citrix ADC admin can now configure a set of allowable file upload formats in the WAF profile. By doing this, you restrict file uploads to specific formats and protect the appliance

against malicious file uploads. But, the protection works only when you disable the “ExcludeFileUploadFormChecks” option in the WAF profile.

How file upload works

When you configure allowable file upload formats, the component interaction is as follows:

- Client request has a form submission with a file upload type, for example PDF.
- As part of security check, WAF inspects the request payload and validates the file type (based on magic signature numbers).
- If the file type is an allowable file format, the corresponding action based on file type binding is applied.
- To validate the file type the appliance inspects the payload and checks for the known magic numbers at known offsets. Each file type has a sequence of magic numbers that validates the file type.
- Only if the validation passes, WAF identifies the file as an allowable format and the associated action is applied.

Configure file type upload by using Citrix ADC CLI

To configure allowable file formats, the appliance uses a WAF profile that is bound to file upload parameters.

1. Configure Web Application Firewall profile

To configure a web application firewall profile, type the following:

```
set appfw profile <profile_name> [-fileUploadTypesAction <fileUploadTypesAction
>]<fileUploadTypesAction> = ( none | block | log | stats )
```

Example

```
set appfw profile profile1 -fileUploadTypesAction block
```

1. Bind Web Application Firewall profile with file upload parameters. The command binds the specified exemption (relaxation) or rule to the specified application firewall profile.

To bind a profile with file upload parameters, type the following:

```
bind appfw profile <profile_name> - fileUploadType <form_field > <form_action_url
> -fileType <fileType> ( pdf | msdoc | text | image | any)
```

[-isNameRegex REGEX (REGEX | NOTREGEX)]
> **Note:**
>

> Form field name is a regular expression type. The default value is ‘NOTREGEX‘.
Example
‘> bind appfw profile test -fileuploadType thefile ”http://10.10.10.10/fileupload_sample/upload.php” -filetype image -isNameRegex‘
-->

Configure file upload security protection by using Citrix ADC GUI

Follow the procedure below to set the file upload settings.

1. In the navigation pane, navigate to **Security** > **Profiles**.
2. In the Profiles page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Security Checks** under **Advanced Settings**.
4. In the **Security Checks** section, go to **File Upload Types** settings.



5. Select the check box and click **Action Settings**.
6. In the **File Upload Types Settings** page, set the file upload action.
7. Click **OK**.
8. In the **Citrix Web App Firewall Profile** page, click **OK** and **Done**.



Configure file upload relaxation rule by using Citrix ADC GUI

You can relax a file upload security protection to avoid false positives. For example, the appliance might block file uploads but you can add a relaxation rule to allow file uploads from specific websites. By doing this, the appliance bypasses security inspection for the specified form field and allow users to upload files from the website mentioned in the action URL.

Follow the procedure below to create a relaxation rule.

1. In the navigation pane, navigate to **Security** > **Citrix Web App Firewall** < **Profiles**.
2. In the Profiles page, click **Add**.
3. In the **Citrix Web App Firewall Profile** page, click **Relaxation Rules** under **Advanced Settings**.
4. In the **Relaxation Rules** section, select **File Upload Types** and click **Edit**.



5. In the **File Upload Types Rexalation Rules** page, click **Add**.
6. In the **File Upload Types Relaxation Rule** page, set the following parameters:
   a) Enabled. Select this check box to enable the relaxation rule.
   b) Form Field Name. Enter the field name that does not require security check.
   c) Action URL. The form submission URL that must be exempted from security check.

d) File Type. File type that must be allowed for the user to upload.
e) Comments. A brief description about the file upload.

7. Click **Create**.



8. In the **Citrix Web App Firewall Profile** page, click **OK** and **Done**.



Configuring and using the Learning feature

October 26, 2021

The learning feature is a repetitive pattern filter that observes activity on a website or application protected by the Web App Firewall, to determine what constitutes normal activity on that website or

application. It then generates a list of up to 2,000 suggested rules or exceptions (relaxations) for each security checks that include support for the learning feature. Users normally find it easier to configure relaxations by using the learning feature than by entering the necessary relaxations manually.

The security checks that support the learning feature are:

- Start URL check
- Cookie Consistency check
- Form Field Consistency check
- Field Formats check
- CSRF Form Tagging check
- HTML SQL Injection check
- HTML Cross-Site Scripting check
- XML Denial-of-Service check
- XML Attachment check
- Web Services Interoperability check

You perform two different types of activities when using the learning feature. First, you enable and configure the feature to use it. You can use learning on all traffic to your protected web applications, or you can configure a list of IPs (called the *Add Trusted Learning Clients* list) from which the learning feature must generate recommendations. Second, after the feature has been enabled and has processed a certain amount of traffic to your protected websites, you review the list of suggested rules and relaxations (learned rules) and mark each with one of the following designations:

- **Edit & Deploy.** The rule is pulled into the Edit dialog box so that you can modify it, and the modified form is deployed.
- **Deploy.** The unmodified learned rule is placed on the list of rules or relaxations for this security check.
- **Skip.** The learned rule is placed on a list of rules or relaxations that are not deployed. The learned rule is removed when skipped. However, as they are not added to relaxations, they might get learned again.

Learning is not performed only when relaxations are in place, except for field format rules. When rules are skipped, they are only removed from learned database. As relaxations are not added, they might get learned again. When the rules are deployed, they are removed from the learned database and also relaxations are added for the rules. As relaxations are added, they would not be learned again. For field format protection, learning is performed irrespective of relaxations.

Although you can use the command line interface for basic configuration of the learning feature, the feature is designed primarily for configuration through the Web App Firewall wizard or the GUI. You can perform only limited configuration of the learning feature by using the command line.

The wizard integrates configuration of learning features with configuration of the Web App Firewall as a whole, and is therefore the easiest method for configuring this feature on a new Citrix ADC appliance

or when managing a simple Web App Firewall configuration. The GUI visualizer and manual interface both provide direct access to all learned rules for all security checks, and are therefore often preferable when you must review learned rules for many security checks.

The learning database is limited to 20 MB in size, which is reached after approximately 2,000 learned rules or relaxations are generated per security check for which learning is enabled. If you do not regularly review and either approve or ignore learned rules and this limit is reached, an error is logged to the NetScaler log and no more learned rules are generated until you review the existing learned rules and relaxations.

If learning stops because the database has reached its size limit, you can restart learning either by reviewing the existing learned rules and relaxations or by resetting the learning data. After learned rules or relaxations are approved or ignored, they are removed from the database. After you reset the learning data, all existing learning data is removed from the database and it is reset to its minimum size. When the database falls below 20 MB in size, learning restarts automatically.

To configure the learning settings by using the command line interface

Specify the Web App Firewall profile to be configured and, for each security check that you want to include in that profile, specify the minimum threshold or the percent threshold. The minimum threshold is an integer representing the minimum number of user sessions that the Web App Firewall must process before it learns a rule or relaxation (default: 1). The percent threshold is an integer representing the percentage of user sessions in which the Web App Firewall must observe a particular pattern (URL, cookie, field, attachment, or rule violation) before it learns a rule or relaxation (default: 0). Use the following commands:

- `set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPercentThres <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-CSRFtagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-fieldConsistencyPercentThreshold <positive_integer>] [-crossSiteScriptingMinThreshold <positive_integer>] [-crossSiteScriptingPerce <positive_integer>] [-SQLInjectionMinThreshold <positive_integer>] [-SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThreshold <positive_integer>] [-fieldFormatPercentThreshold <positive_integer>] [-XMLWSIMinThreshold <positive_integer>] [-XMLWSIPercentThreshold <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer>] [-XMLAttachmentPercentThreshold <positive_integer>]`
- `save ns config`

Example

The following example enables and configures the learning settings in the profile or the HTML SQL Injection security check. This is an appropriate initial test bed learning configuration, where you have complete control over the traffic that is sent to the Web App Firewall.

```
set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
save ns config
<!--NeedCopy-->
```

To reset learning settings to their defaults by using the command line interface

To remove any custom configuration of the learning settings for the specified profile and security check, and return the learning settings to their defaults, at the command prompt type the following commands:

- `unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFtagMinThreshold] [-CSRFtagPercentThresh] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-fieldFormatMinThreshold] [-fieldFormatPercentThreshold] [-XMLWSIMinThreshold] [-XMLWSIPercentThreshold] [-XMLAttachmentMinThreshold] [-XMLAttachmentPercen]`
- `save ns config`

To display the learning settings for a profile by using the command line interface

At the command prompt, type the following command:

```
show appfw learningsettings <profileName>
```

To display unreviewed learned rules or relaxations for a profile by using the command line interface

At the command prompt, type the following command:

```
show appfw learningdata <profileName> <securityCheck>
```


To remove specific unreviewed learned rules or relaxations from the learning database by using the command line interface

At the command prompt, type the following command:

```
rm appfw learningdata <profileName> (-startURL <expression> | -cookieConsistency
 <string> | (-fieldConsistency <string> <formActionURL>)| (-crossSiteScripting
 <string> <formActionURL>)| (-SQLInjection <string> <formActionURL>)| (-
fieldFormat <string><formActionURL>)| (-CSRFTag <expression> <CSRFFormOriginURL
>)| -XMLDoSCheck <expression> | -XMLWSICheck <expression> | -XMLAttachmentCheck
 <expression>)[-TotalXMLRequests]
```

Example

The following example removes all unreviewed learned relaxations for the profile, HTML SQL Injection security check, that apply to the last name form field.

```
1 rm appfw learningdata pr-basic -SQLInjection LastName
2 <!--NeedCopy-->
```

To remove all unreviewed learned data by using the command line interface

At the command prompt, type the following command:

```
reset appfw learningdata
```

To export learning data by using the command line interface

At the command prompt, type the following command:

```
export appfw learningdata <profileName> <securitycheck>[-target <string>]
```

Example

The following example exports learned relaxations for the profile and the HTML SQL Injection security check to a comma-separated values (CSV) format file in the /var/learnt_data/ directory under the file name specified in the -target parameter.

```
1 export appfw learningdata pr-basic SQLInjection -target sqli_ld
2 <!--NeedCopy-->
```


To configure the Learning feature by using the GUI

1. Navigate to **Security** > **Web App Firewall** > **Profiles**.
2. In the **Profiles** pane, select the profile, and then click **Edit**.
3. Click **Learnt Rules** under **Advanced Settings** section.
4. In the **Learnt Rules** section, select a security check and click **Settings**.
5. In the **Security Check Settings** page, set the following parameters:
   a) **Minimum number threshold.** Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1
   b) **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
6. Click **OK** and **Close**.



7. Click **Remove All Learned Data** to remove all learned data and reset the learning feature, so that it must start its observations again from the beginning.

> **Note:**
>
> This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.

8. To restrict the learning engine to traffic from a specific set of IPs, click **Trusted Learning Clients**, and add the IP addresses that you want to use to the list.
    a) To add an IP address or IP address range to the Trusted Learning Clients list, click **Add**.
    b) In the **Add Trusted Learning Clients** dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
    c) In the Comments text area, type a comment that describes this IP address or range.
    d) Click **Create** to add your new IP address or range to the list.
    e) To modify an existing IP address or range, click the IP address or range, and then click **Open**. Except for the name, the dialog box that appears is identical to the **Add Trusted Learning Clients** dialog box.
    f) To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click **Disable or Enable**, as appropriate.
    g) To remove an IP address or range completely, click the IP address or range, and then click **Remove**.
9. Click **Close** to return to the Configure Web App Firewall Profile page.
10. Click **Done**.

To review learned rules or relaxations by using the GUI

1. Navigate to **Security** > **Web App Firewall** > **Profiles**.
2. In the **Profiles** pane, select the profile, and then click **Edit**.
3. Click **Learnt Rules** under **Advanced Settings** section.
4. In the **Learnt Rules** section, select a security check and click **Settings**.
5. To review the learned data hierarchically as a branching tree, enabling you to choose general patterns that match many of the learned patterns, click **Visualizer**.
6. If you have chosen to review actual learned patterns, perform the following steps.
7. Select the first learned relaxation and choose how to handle it.
    a) To modify and then accept the relaxation, click **Edit & Deploy**, edit the relaxation regular expression, and then click **OK**.
    b) To accept the relaxation without modifications, click **Deploy**.
    c) To remove the relaxation from the list without deploying it, click **Skip**.

d) Repeat the previous step to review each additional learned relaxation.

8. Click **Close** to return to the **Manage Learned Rules** dialog box.
9. Click **Done**.



Dynamic profiling

September 14, 2021

The learning feature is a pattern filter that observes and learns activities on the back-end server. Based on the observation, the learning engine generates up to 2000 rules or exceptions (relaxations) for each security check. To automate the process and auto deploy the relaxation rules, Citrix ADC appliance uses dynamic profiling.

With dynamic profiling, the appliance records the learnt data for a pre-defined threshold and sends an SNMP alert to the user. If the user does not skip the data within a grace period, the appliance auto deploys it as a relaxation rule. Earlier, the user had to manually deploy the relaxation rules. Currently, dynamic profiling is available only for the follow security checks:

1. HTML SQL injection
2. HTML Cross Site scripting
3. Field format
4. Start URL
5. Content-type
6. Field formats
7. CSRF form tagging
8. Cookie consistency
9. Deny URL
10. Buffer Overflow
11. Credit Card

For example, consider the HTML SQL Injection security check enabled with dynamic profiling. You can use learning for a list of IPs (called the Trusted Learning Clients list) from which the learning feature must generate recommendations. To configure a list of trusted clients, see Learning Trusted Clients topic. If the incoming traffic has violations, it is recorded as a learnt data. If the learned data is recorded in the learning engine, the appliance sends an SNMP alert to the user. If the user does not recognize a false positive and does not skip the learnt data within a grace period, the appliance auto deploys it as a relaxation rule.

> **Note:**
> After you configure the dynamic profile, you must periodically review the appliance configuration for the auto-deployment of the relaxation rules and save it on the appliance.

Configure dynamic profiling by using the Citrix ADC command interface

Dynamic profiling is available for Start URL, HTML Cross-Site Scripting, Field Format, or HTML SQL Injection security checks. To configure dynamic profiling, you must complete the following steps.

1. Configure dynamic learning
2. Configure auto deployment grace period

Configure dynamic learning

As a first step, you must configure dynamic learning on your appliance. At the command prompt, type:

```
set appfw profile <profile_name> dynamicLearning <security_checks>
```

Example

```
set appfw profile test1 dynamicLearning SQLInjection CrossSiteScripting
fieldFormat startURL
```

Configure auto deployment grace period

Once you enable the feature on specific security checks, you must configure the grace period for the auto deployment.

```
set appfw learningsettings <profile name> -crossSiteScriptingAutoDeployGracePeriod
 <seconds>
```

```
set appfw learningsettings <profile name> fieldFormatAutoDeploymentGracePeriod
 <seconds>
```

```
set appfw learningsettings <profile name> SQLInjectionAutoDeploymentGracePeriod
 <seconds>
```



```
set appfw learningsettings <profile name> -startURLAutoDeployGracePeriod <
seconds>
```

Example

```
set appfw learningsettings test1 -crossSiteScriptingAutoDeployGracePeriod 30
set appfw learningsettings test1 -startURLAutoDeployGracePeriod 7
set appfw learningsettings test1 -fieldFormatAutoDeploymentGracePeriod 10
set appfw learning settings test1 -SQLInjectionAutoDeploymentGracePeriod 12
```

> **Note:**
>
> Here, the auto deployment grace period is in minutes.

Configuring dynamic profiling by using the Citrix ADC GUI

1. Navigate to **Security > Citrix Web App Firewall > Profile**.
2. In the details pane, select a profile and click **Edit**.
3. In the **Citrix Web App Profile** page, click **Dynamic Profiling** under **Advanced Settings**.



4. In the **Dynamic Profiling** section, select a security check and click **Edit**.



5. In the **Dynamic Profiling and Learning Settings** page, set the grace period the security check.



6. Click **OK** and **Done**.

Export and import of relaxation rules

When you enable dynamic profiling, the learnt data is auto deployed as relaxation rules. Along with this, the appliance also enables you to export the dynamic profiling based relaxation rules and regular relaxation rules. You can export the rules from staging environment and import it to the production environment.

> **Note:**
>
> When you import rules to the production environment, you must ensure the process is additive and does not override the existing configuration.

How to export and import relaxation rules

To export and import the relaxation rules, you must complete the following steps:

1. You must first export the dynamic profiling-based data. For this, the export option is available for the relaxation rules in the WAF profile. When you select this option, you export the dynamic profiling relaxation rules and regular relaxation rules. You can use the export option to download the configuration as a compressed bundle on the appliance.
2. Once you have exported the data from the staging environment, you must import it to another Citrix ADC appliance. For this, you must use the import option available in the relaxation rules of the WAF profile. When you select this option, the appliance imports the specified relaxation rules bundled and restores it to the WAF profile of the selected appliance.

> **Note:**
>
> If you are going to import relaxation rules in a WAF profile, there are two types of action:
> Augment – This action ensures import is additive, thus not overriding any existing configuration.
> Overwrite – This action overwrites the existing configuration with configuration present in the compressed export bundle."

Import archived relaxation rules file by using CLI

To import the relaxation rules, you must import the archive into the Citrix ADC appliance and then run the restore command to extract the configuration. The following set of CLI commands can be used for exporting, importing, and managing the configurations.

To import the archived file from the specific location and restore, at the command prompt, type:

```
import appfw archive <src> <name> [-comment <string>]
```

Where,
"src": Indicates the source of the tar archive file in the form, `<protocol>://<host>[:<port>][/<path>]`

“name”: Indicates name of archive.
“comment”: Comments associated with this archive.

```
restore appfw profile <archivename> [-relaxationRules] [-importProfileName
 <string>] [-matchUrlString <string>] [-replaceUrlString <string>] [-
overwrite] [-augment]
```

Where,
archivename: Indicates source for tar archive. This is a mandatory argument.
“relaxationRules”: Option to import all appfw relaxation rules.

importProfileName: Indicates profile name created or updated to associate the relaxation rules during restore operation.
“matchUrlString”: Indicates action URL string to match in archived relaxation rules.

replaceUrlString: Indicates string to replace in action URL while restoring relaxation rules.

overwrite: Existing rules action to purge existing relaxation rules and replace during import.

augment: Existing rules action to augment relaxation rules during import.

Example:

```
import appfw archive local: dutA_test_pr.tgz demo
restore appfw profile dutA_test_pr
```

Export the archived file to the selected appliance by using the CLI

If you use CLI to export the appfw relaxation rules, you must archive the configuration and then export it.
To archive and export the archived file, at the command prompt, type:

```
archive appfw profile <name> <archivename> [-comment <string>]
```

Where,
archive name: Indicates source for tar archive. This is a mandatory argument.
name: Indicates the appfw profile name containing the relaxation rules to export

```
export appfw archive <name> <target>
```

Where,
Name. Name of tar archive. This is a mandatory argument. Maximum Length: 31
Target. Path to the file to be exported. This is a mandatory argument. Maximum Length: 2047

Example:

```
> archive appfw profile test_pr archived_test_pr

> export appfw archive archived_test_pr local:dutA_test_pr
```


To export relaxation rules by using Citrix ADC GUI

Follow the steps given below to export relaxation rules:

1. Navigate to **Security** > **Citrix Web App Firewall**.
2. In the details page, click **Citrix Web App Firewall Profiles** link under **Configuration Summary** section.
3. In the **Citrix Web App Firewall Profile** page, click the **Relaxation Rules** link under **Advanced Settings** section.
4. In the **Relaxation Rules** section, click **Export All Relaxation Rules**. The action applies to all security checks and on the ones which dynamic learning is enabled on that profile.



To import relaxation rules by using Citrix ADC GUI

Complete the steps to import relaxation rules:

1. Navigate to **Security** > **Citrix Web App Firewall**.
2. In the details page, click the **Citrix Web App Firewall Profiles** link under **Configuration Summary** section.
3. In the **Citrix Web App Firewall Profile** page, click the **Relaxation Rules** link under **Advanced Settings** section.
4. In the **Relaxation Rules** section, click **Import All Relaxation Rules**.
5. In the **Configure Citrix Web App Firewall Profile** page, set the following parameters:
    a) Local file. Name of the compressed archived file containing the relaxation rules.
    b) Profile Name. Name of the profile to which the relaxation rules are bound.
    c) Matching URL String. Portion of the URL that matches.
    d) Replace URL String. Portion of the URL that replaces the URL string.
    e) Existing Rule Action. Select if the rule must overwrite existing rules or augment the existing rules.

6. Click **OK**.



Supplemental Information about profiles

September 14, 2021

Following is supplemental information about particular aspects of Web App Firewall profiles. This information explains how to include special characters in a security check rule or relaxation, and how to use variables when configuring profiles.

Configuration variable support

Instead of using static values, to configure the Web App Firewall's security checks and settings, you can now use standard Citrix ADC named variables. By creating variables, you can more easily export and then import configurations to new Citrix ADC appliances, or update existing Citrix ADC appliances from a single set of configuration files. This simplifies updates when you use a test bed setup to develop a complex Web App Firewall configuration that is tuned for your local network and servers and then transfer that configuration to your production Citrix ADC appliances.

You create Web App Firewall configuration variables in the same manner as you do any other Citrix ADC

named variables, following standard Citrix ADC conventions. To create a named expression variable by using the GUI, you use the Add Expression dialog box. To create a named expression variable by using the Citrix ADC command line, you use the add expression command followed by the appropriate parameter.

The following URLs and expressions can be configured with variables instead of static values:

- **Start URL** (-starturl)
- **Deny URL** (-denyurl)
- **Form Action URL** for *Form Field Consistency Check* (-fieldconsistency)
- **Action URL** for *XML SQL Injection Check* (-xmlSQLInjection)
- **Action URL** for X*ML Cross-Site Scripting Check* (-xmlcross-site scripting)
- **Form Action URL** for *HTML SQL Injection Check* (-sqlInjection)
- **Form Action URL** for *Field Format Check* (-fieldFormat)
- **Form Origin URL** and **Form Action URL** for *Cross-Site Request Forgery (CSRF) Check* (-csrfTag)
- **Form Action URL** for *HTML Cross-Site Scripting Check* (-crossSiteScripting)
- **Safe Object** (-safeObject)
- **Action URL** for *XML Denial-of-Service (XDoS) check* (-XMLDoS)
- **URL** for *Web Services Interoperability check* (-XMLWSIURL)
- <**URL** for *XML Validation check* (-XMLValidationURL)
- **URL** for *XML Attachment check* (-XMLAttachmentURL)

For more information, see Policies and Expressions.

To use a variable in the configuration, you enclose the variable name between two at (@) symbols and then use it exactly as you would the static value that it replaces. For example, if you are configuring the Deny URL check by using the GUI and want to add the named expression variable myDenyURL to the configuration, you would type @myDenyURL@ into the Add Deny URL dialog box, Deny URL text area. To do the same task by using the Citrix ADC command line, you would type add appfw profile <name> -denyURLAction @myDenyURL@.

PCRE character encoding format

The Citrix ADC operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your Web App Firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your Web App Firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII

characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.

- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.
- **ASCII control characters.** Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters must never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The Citrix ADC appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, "English US," the Web App Firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The Web App Firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The Web App Firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The Web App Firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The Web App Firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The Web App Firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The Web App Firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- **Unicode (UTF-8).** The Web App Firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the Web App Firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols

and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the Web App Firewall configuration is "xNN" for ASCII characters; "\xNN\xNN" for non-ASCII characters used in English, Russian, and Turkish; and "\xNN\xNN\xNN" for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an Web App Firewall regular expression as a UTF-8 character, you would type \x21. If you want to include an á, you would type \xC3\xA1.

> **Note:**
>
> Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character's UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as x20 to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the Web App Firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters.

  Actual URL: `http://www.josénuñez.com`

  Encoded URL: `^http://www\[.\]jos\xC3\xA9nu\xC3\xB1ez\[.\]com$`

- Another URL containing extended ASCII characters.

  Actual URL: `http://www.example.de/trömso.html`

  Encoded URL: `^http://www[.]example\[.]de/tr\xC3\xB6mso[.]html$`

- A form field name containing extended ASCII characters.

  `Actual Name: nome_do_usuário`

  Encoded Name: `^nome_do_usu\xC3\xA1rio$`

- A safe object expression containing extended ASCII characters.

  Unencoded Expression [A-Z]{3,6}¥[1-9][0-9]{6,6}

  Encoded Expression: [A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful website that contains this information is located at the following URL:

http://www.utf8-chartable.de/unicode-utf8-table.pl

For the characters in the table on this website to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Inverted PCRE Expressions

In addition to matching content that contains a pattern, you can match content that does not contain a pattern by using an inverted PCRE expression. To invert an expression, you simply include an exclamation point (!) followed by white space as the first character in the expression.

> **Note:** If an expression consists only of an exclamation point with nothing following, the exclamation point is treated as a literal character, not syntax indicating an inverted expression.

The following Web App Firewall commands support inverted PCRE expressions:

- Start URL (URL)
- Deny URL (URL)
- Form Field Consistency (form action URL)
- Cookie Consistency (form action URL)
- Cross Site Request Forgery (CSRF) (form action URL)
- HTML Cross-site Scripting (form action URL)
- Field Format (form action URL)
- Field Type (type)
- Confidential Field (URL)

Note: If the security check contains an isRegex flag or check box, it must be set to YES or checked to enable regular expressions in the field. Otherwise the contents of that field are treated as literal and no regular expressions (inverted or not) are parsed.

Disallowed Names for Web App Firewall Profiles

The following names are assigned to built-in actions and profiles on the Citrix ADC appliance, and cannot be used as names for a user-created Web App Firewall profile.

- AGRESSIVE
- ALLOW
- BASIC

- CLIENTAUTH
- COMPRESS
- CSSMINIFY
- DEFLATE
- DENY
- DNS-NOP
- DROP
- GZIP
- HTMLMINIFY
- IMGOPTIMIZE
- JSMINIFY
- MODERATE
- NOCLIENTAUTH
- NOCOMPRESS
- NONE
- NOOP
- NOREWRITE
- RESET
- SETASLEARNNSLOG_ACT
- SETNSLOGPARAMS_ACT
- SETSYSLOGPARAMS_ACT
- SETTMSESSPARAMS_ACT
- SETVPNPARAMS_ACT
- SET_PREAUTHPARAMS_ACT
- default_DNS64_action
- dns_default_act_Cachebypass
- dns_default_act_Drop
- nshttp_default_profile
- nshttp_default_strict_validation
- nstcp_default_Mobile_profile
- nstcp_default_XA_XD_profile
- nstcp_default_profile
- nstcp_default_tcp_interactive_stream
- nstcp_default_tcp_lan
- nstcp_default_tcp_lan_thin_stream
- nstcp_default_tcp_lfp
- nstcp_default_tcp_lfp_thin_stream
- nstcp_default_tcp_lnp
- nstcp_default_tcp_lnp_thin_stream

- nstcp_internal_apps

Custom error status and message for HTML, XML, and JSON error object

September 14, 2021

When the Citrix Web App Firewall detects a violation, the appliance handles the error scenario using either a redirect URL or the error object (imported into the profile and enabled). If the scenario is handled using an error object configuration, the WAF profile provides a custom response status code and message. You can customize the response error details for an HTML, XML, or JSON error object in the WAF profile.

> **Note:**
>
> By default, the error code and error message are set as "200" and "OK" if error object settings are configured.

When handling error scenarios, it is important for the appliance to respond with appropriate HTTP response status code and message for resolving issues. By providing a custom error status message and custom error status code, the appliance can provide better user intervention to resolve a problem when a violation occurs. For example, if you set the response error code to "404" and the status message to "Not Found", the user can inspect the response status code and message to check if a violation has occurred. This can help the user to filter responses that contain the error object

Configure custom status code and message for HTML error object in a WAF profile by using the CLI

At the command prompt, type:

```
1 set appfw profile <profile-name> -HTMLErrorStatusCode <value> -
     HTMLErrorStatusMessage <value> -useHTMLErrorObject ON
2 <!--NeedCopy-->
```

Example:

```
set appfw profile profile_1 -HTMLErrorStatusCode 404 -HTMLErrorStatusMessage
  "Not Found"  -useHTMLErrorObject ON
```

Configure custom status code and message for XML error object in a WAF profile by using the CLI

At the command prompt, type:


```
set appfw profile <profile-name> -XMLErrorStatusCode <value> -
    XMLErrorStatusMessage <value>
<!--NeedCopy-->
```

Example:

```
set appfw profile profile_1 -XMLErrorStatusCode 406 - XMLErrorStatusMessage
  “Not Acceptable”
```

Configure custom status code and message for JSON error object in a WAF profile by using the CLI

At the command prompt, type:

```
set appfw profile <profile-name> -JSONErrorStatusCode <value> -
    JSONErrorStatusMessage <value>
<!--NeedCopy-->
```

Example:

```
set appfw profile profile_1 -JSONErrorStatusCode 500 - JSONErrorStatusMessage
  “Internal Server Error”
```

Configure custom status code and message for HTML, JSON, or XML error object in a WAF profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, click **Edit**.
3. In the **Create Web App Firewall Profile** page, click **Profile Settings** from the **Advanced Settings** section.
4. In the **Profile settings** section, set the following parameters.

   a. HTML Error Object. Select the option for handing error scenarios using an HTML error object. Import the error object from a URL, file, or text.
   b. HTML Error Status Code. Provide a custom error status code.
   c. HTML Error Status Message. Provide a customer error message.
5. Click **OK** and **Done**.

> **Note:**
>
> The same procedure is applicable for JSON and XML custom error object settings.



Policy labels

September 14, 2021

A policy label consists of a set of policies, other policy labels, and virtual server-specific policy banks. The Web App Firewall evaluates each policy bound to the policy label in order of priority. If the policy matches, it filters the connection as specified in the associated profile. Then it does whatever the Goto parameter specifies, which can be to terminate policy evaluation, go to the next policy, or go to the policy with the specified priority. If the Invoke parameter is set, it terminates processing of the current policy label and begins to process the specified policy label or virtual server.

To create an Web App Firewall policy label by using the command line

At the command prompt, type the following commands:

- `add appfw policylabel <labelName> http_req`
- `save ns config`

Example

The following example creates a policy label named policylbl1.

```
1 add appfw policylabel policylbl1 http_req
2 save ns config
3 <!--NeedCopy-->
```

To bind a policy to a policy label by using the command line

At the command prompt, type the following commands:

- `bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpressi >] [-invoke (<labelType> <labelName>)]`
- `save ns config`

Example

The following example binds the policy policy1 to the policy label policylbl1 with a priority of 1.

```
bind appfw policylabel policylbl1 policy1 1
save ns config
<!--NeedCopy-->
```

To configure an Web App Firewall policy label by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Policy Labels**.
2. In the details pane, do one of the following:
   - To add a new policy label, click **Add**.
   - To configure an existing policy label, select the policy label and the click **Open**.

   The **Create Web App Firewall Policy Label** or the **Configure Web App Firewall Policy Label** dialog box opens. The dialog boxes are nearly identical.
3. If you are creating a new policy label, in the Create Web App Firewall Policy Label dialog box, type a name for your new policy label.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
4. Select **Insert Policy** to insert a new row and display a drop-down list with all existing Web App Firewall policies.
5. Select the policy you want to bind to the policy label, or select New Policy to create a new policy and follow the instructions in To create and configure a policy by using the GUI. The policy that you selected or created is inserted into the list of globally bound Web App Firewall policies.
6. Make any additional adjustments.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Web App Firewall Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.

- To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression

7. Repeat steps 5 through 7 to bind any additional Web App Firewall policies you want to the policy label.
8. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that you have successfully created or modified the policy label.

Policies

September 14, 2021

The Web App Firewall uses two types of policies: firewall policies and auditing policies. Firewall policies control which traffic is sent to the Web App Firewall. Auditing policies control the log server to which Web App Firewall logs are sent.

Firewall policies can be complex because the policy rule can consist of multiple expressions in the Citrix ADC expressions language, which is a full-fledged object oriented programming language capable of defining with extreme precision exactly which connections to filter. Because firewall policies operate within the context of the Web App Firewall, they must meet certain criteria that are connected to how the Web App Firewall functions and what traffic is appropriately filtered by it. As long as you keep these criteria in mind, however, firewall policies are similar to policies for other Citrix ADC features. The instructions here do not attempt to cover all aspects of writing firewall policies, but only provide an introduction to policies and cover those criteria that are unique to the Web App Firewall.

Auditing policies are simple because the policy rule is always ns_true. You need only specify the log server that you want to send logs to, the logging levels that you want to use, and a few other criteria that are explained in detail.

Web App Firewall Policies

September 14, 2021

A firewall policy is a rule associated with a profile. The rule is an expression or group of expressions that define the types of request/response pairs that the Web App Firewall is to filter by applying the profile. Firewall policy expressions are written in the Citrix ADC expressions language, an object-oriented programming language with special features to support specific Citrix ADC functions. The profile is the set of actions that the Web App Firewall is to use to filter request/response pairs that match the rule.

Firewall policies enable you to assign different filtering rules to different types of web content. Not all web content is alike. A simple website that uses no complex scripting and accesses and handles no private data might require only the level of protection provided by a profile created with basic defaults. Web content that contains JavaScript-enhanced web forms or accesses an SQL database probably requires more tailored protection. You can create a different profile to filter that content and create a separate firewall policy that can determine which requests are attempting to access that content. You then associate the policy expression with a profile you created and globally bind the policy to put it into effect.

The Web App Firewall processes only HTTP connections, and therefore uses a subset of the overall Citrix ADC expressions language. The information here is limited to topics and examples that are likely to be useful when configuring the Web App Firewall. Following are links to additional information and procedures for firewall policies:

- For procedures that explain how to create and configure a policy, see Creating and Configuring Web App Firewall Policies.
- For a procedure that explains in detail how to create a policy rule (expression), see To create or configure an Web App Firewall rule (expression).
- For a procedure that explains how to use the Add Expression dialog box to create a policy rule, see To add a firewall rule (expression) by using the Add Expression dialog box.
- For a procedure that explains how to view the current bindings for a policy, see Viewing a Firewall Policy's Bindings.
- For procedures that explain how to bind an Web App Firewall policy, see Binding Web App Firewall Policies.
- For detailed information about the Citrix ADC expressions language, see Policies and Expressions.

> **Note**
>
> Web App Firewall evaluates the policies based on the configured priority and goto expressions. At the end of the policy evaluation, the last policy that evaluates to true is used and the security configuration of the corresponding profile is invoked for processing the request.
>
> For example, Consider a scenario where there are 2 policies.
>
> - Policy_1 is a generic policy with Expression=ns_true and has a corresponding profile_1 which is a basic profile. The priority is set to 100.
> - Policy_2 is more specific with Expression=HTTP.REQ.URL.CONTAINS("XYZ") and has a corresponding profile_2 which is an advance profile. The GoTo Expression is set to NEXT and the priority is set to 95 which is a higher priority compared to Policy_1.
>
> In this scenario, if the target string "XYZ" is detected in the URL of the processed request, Policy_2 match is triggered as it has a higher priority even though Policy_1 is also a match. However, as

> per the GoTo expression configuration of Policy_2, the policy evaluation continues and the next policy Policy_1 is also processed. At the end of the policy evaluation, Policy_1 evaluates as true and the basic security checks configured in Profile_1 are invoked.
>
> If the Policy_2 is modified and the GoTo Expression is changed from **NEXT** to **END**, the processed request that has the target string "XYZ", triggers the Policy_2 match due to priority consideration and as per the GoTo expression configuration, the policy evaluation ends at this point. Policy_2 evaluates as true and the advanced security checks configured in Profile_2 are invoked.
>
> **NEXT**
> **END**
>
> Policy evaluation is completed in one pass. Once the policy evaluation is completed for the request and the corresponding profile actions are invoked, the request does not go through another round of policy evaluation.

Creating and configuring Web App Firewall policies

September 14, 2021

A firewall policy consists of two elements: a *rule*, and an associated *profile*. The rule selects the HTTP traffic that matches the criteria that you set, and sends that traffic to the Web App Firewall for filtering. The profile contains the filtering criteria that the Web App Firewall uses.

The policy rule consists of one or more expressions in the Citrix ADC expressions language. The Citrix ADC expressions syntax is a powerful, object-oriented programming language that enables you to precisely designate the traffic that you want to process with a specific profile. For users who are not familiar with the Citrix ADC expressions language syntax, or who prefer to configure their Citrix ADC appliance by using a web-based interface, the GUI provides two tools: the **Prefix** menu and the **Add Expression** dialog box. Both help you to write expressions that select exactly the traffic that you want to process. Experienced users who are thoroughly familiar with the syntax may prefer to use the Citrix ADC command line to configure their Citrix ADC appliances.

> **Note:**
>
> In addition to the default expressions syntax, for backward compatibility the Citrix ADC operating system supports the Citrix ADC classic expressions syntax on Citrix ADC Classic and nCore appliances and virtual appliances. Classic expressions are not supported on Citrix ADC Cluster appliances and virtual appliances. Current Citrix ADC users who want to migrate existing configurations to the Citrix ADC Cluster must migrate any policies that contain classic expressions to the default expressions syntax.

For detailed information about the Citrix ADC expressions languages, see Policies and Expressions.

You can create a firewall policy by using the GUI or the Citrix ADC command line.

To create and configure a policy by using the command line interface

At the command prompt, type the following commands:

- `add appfw policy <name><rule> <profileName>`
- `save ns config`

Example

The following example adds a policy named pl-blog, with a rule that intercepts all traffic to or from the host blog.example.com, and associates that policy with the profile pr-blog. This is an appropriate policy to protect a blog hosted on a specific host name.

```
1 add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
     ")" pr-blog
2 <!--NeedCopy-->
```

To create and configure a policy by using the GUI

1. Navigate to **Security** > **Web App Firewall** > **Policies**.
2. In the details pane, do one of the following:
   - To create a firewall policy, click **Add**. The **Create Web App Firewall Policy** is displayed.
   - To edit an existing firewall policy, select the policy, and then click **Edit**.

   The **Create Web App Firewall Policy** or **Configure Web App Firewall Policy** is displayed.
3. If you are creating a firewall policy, in the **Create Web App Firewall Policy** dialog box, Policy Name text box, type a name for your new policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

   If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.
4. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a profile to associate with your policy by clicking New, and you can modify an existing profile by clicking Modify.
5. In the Expression text area, create a rule for your policy.
   - You can type a rule directly into the text area.

- You can click Prefix to select the first term for your rule, and follow the prompts.
- You can click Add to open the Add Expression dialog box, and use it to construct the rule.

6. Click **Create** or **OK**, and then click **Close**.

To create or configure a Web App Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the Web App Firewall filters by using the profile associated with the policy. Like other Citrix ADC policy rules (or *expressions*), the Web App Firewall rules use Citrix ADC expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the **Web App Firewall** wizard or the Citrix ADC GUI to create your policy rule:
   - If you are configuring a policy in the **Web App Firewall** wizard, in the navigation pane, click **Web App Firewall**, then in the details pane click **Web App Firewall Wizard**, and then navigate to the **Specify Rule** screen.
   - If you are configuring a policy manually, in the navigation pane, expand **Web App Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, the **Create Web App Firewall Profile** dialog box, or the **Configure Web App Firewall Profile** dialog box, click **Prefix**, and then choose the prefix for your expression from the drop-down list. Your choices are:
   - **HTTP.** Choose an HTTP protocol if you want to examine some aspect of the request that pertains to the protocol.
   - **SYS.** Choose protected websites if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** Choose a client that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** Choose a client to which the request was sent and if you want to examine some aspect of the recipient of the request.

   After you choose a prefix, the Web App Firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.
3. Choose your next term.

   If you chose the HTTP protocol as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The Web App Firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For

help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you have decided which term you want, double-click it to insert it into the **Expression** window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(“”), type the header name between the quotation marks.
5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

   Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

     ```
     1 HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     2 <!--NeedCopy-->
     ```

     For `shopping.example.com`, substitute the name of the web host that you want to match.

   - **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

     ```
     1 HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
     2 <!--NeedCopy-->
     ```

     For www.example.com, substitute the name of the web host. For the folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

   - **Specific type of content: GIF images.** To match GIF format images:

     ```
     1 HTTP.REQ.URL.ENDSWITH(".png")
     2 <!--NeedCopy-->
     ```

     To match other format images, substitute another string in place of .png.

   - **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

     ```
     1 HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
     2 <!--NeedCopy-->
     ```

     To match all JavaScript with .js extensions:


```
HTTP.REQ.URL.ENDSWITH(".js")
<!--NeedCopy-->
```

For more information about creating policy expressions, see Policies and Expressions.

> **Note:**
>
> If you use the command line to configure a policy, remember to escape any double quotation marks within Citrix ADC expressions. For example, the following expression is correct if entered in the GUI:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

If entered at the command line, however, you must type the following command instead:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

To add a firewall rule (expression) by using the Add Expression dialog box

The **Add Expression** dialog box (also referred to as the Expression Editor) helps users who are not familiar with the Citrix ADC expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the **Web App Firewall** wizard or the Citrix ADC GUI:
   - If you are configuring a policy in the **Web App Firewall** wizard, in the navigation pane, click **Web App Firewall**, then in the details pane click **Web App Firewall Wizard**, and then navigate to the **Specify Rule** screen.
   - If you are configuring a policy manually, in the navigation pane, expand **Web App Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, in the **Create Web App Firewall Profile** dialog box, or in the **Configure Web App Firewall Profile** dialog box, click **Add**.
3. In the **Add Expression dialog** box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
   - **HTTP.** Choose HTTP protocol if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
   - **SYS.** Choose protected websites if you want to examine some aspect of the request that pertains to the recipient of the request.

- **CLIENT.** Choose the computer that sent the request if you want to examine some aspect of the sender of the request.
- **SERVER.** Choose the computer to which the request was sent and to examine some aspect of the recipient of the request.

4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the Web App Firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the **Preview Expression** window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The **Preview Expression** window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or more terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click **OK** to close the **Add Expression** dialog box. Your expression is inserted into the **Expression** text area.

Binding Web App Firewall policies

September 14, 2021

After you have configured your Web App Firewall policies, you bind them to Global or a bind point to put them into effect. After binding, any request or response that matches an Web App Firewall policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the Citrix ADC OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the Web App Firewall feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you configure the Web App Firewall to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you configure the Web App Firewall to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other

policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you bind your policies.

For more information about binding policies on the Citrix ADC appliance, see "Policies and Expressions."

To bind an Web App Firewall policy by using the command line interface

At the command prompt, type the following commands:

- `bind appfw global <policyName>`
- `bind appfw profile <profile_name> -crossSiteScripting data`

Example

The following example binds the policy named pl-blog and assigns it a priority of 10.

```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

Configure log expressions

The log expression support for binding Web App Firewall is added to log HTTP header information when a violations occurs.

Log expression is binded at the Application profile, and binding contains the expression that needs to be evaluated and sent to logging frameworks when violation occurs.

The Web App Firewall violation log record with http header information is recorded. You can specify a custom log expression and it helps in analysis and diagnosis when violations are generated for the current flow (request/response).

Example configuration

```
bind appfw profile <profile> -logexpression <string> <expression>
add policy expression headers "\" HEADERS(100):\"+HTTP.REQ.FULL_HEADER"
add policy expression body_100 "\"BODY:\"+HTTP.REQ.BODY(100)"
bind appfw profile test -logExpression log_body body_100
bind appfw profile test -logExpression log_headers  headers
bind appfw profile test -logExpression "\"URL:\"+HTTP.REQ.URL+\" IP:\"+
    CLIENT.IP.SRC"
<!--NeedCopy-->
```


Example logs

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12.1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=POST request=http://10.217.222.44/test/credit.html msg= HEADERS(100):POST /test/credit.html HTTP/1.1^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Host: 10.217.222.44^M Accept: /^M Content-Length: 33^M Content-Type: application/x-www-form-urlencoded^M ^M  cn1=58 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12.1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=POST request=http://10.217.222.44/test/credit.html msg=BODY:ata\=asdadasdasdasddddddddddddddd cn1=59 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12.1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=POST request=http://10.217.222.44/test/credit.html msg=URL:/test/credit.html IP:10.217.222.128 cn1=60 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Other violation logs
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12.1|APPFW|APPFW_STARTURL|6|src=10.217.222.128 spt=26409 method=POST request=http://10.217.222.44/test/credit.html msg=Disallow Illegal URL. cn1=61 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12.1|APPFW|APPFW_SAFECOMMERCE|6|src=10.217.222.128 spt=26409 method=POST request=http://10.217.222.44/test/credit.html msg=Maximum number of potential credit card numbers seen cn1=62 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```


Note

1. Only auditlog support is available. Support for logstream and visibility in security insight would be added in future release versions.
2. If auditlogs are generated, then only 1024 bytes of data can be generated per log message.
3. If log streaming is used, then the limits are based on maximum supported size of log stream/ipfix protocol size limitations. Maximum support size for log stream is larger than 1024 bytes.

To bind an Web App Firewall policy by using the GUI

1. Do one of the following:
   - Navigate to **Security** > **Web App Firewall**, and in the details pane, click Web App Firewall policy manager.
   - Navigate to **Security** > **Web App Firewall** > **Policies** > **Firewall Policies**, and in the details pane, click **Policy Manager**.
2. In the **Web App Firewall Policy Manager** dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
   - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the Citrix ADC appliance, and are applied before any other policies.
   - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
   - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
   - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the Citrix ADC appliance.
   - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
   - **None.** Do not bind the policy to any bind point.
3. Click **Continue**. A list of existing Web App Firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.

- To modify the policy expression, double click that field to open the **Configure Web App Firewall Policy** dialog box, where you can edit the policy expression.
- To set the Goto Expression, double click **field** in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
- To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression

6. Repeat steps 3 through 6 to add any additional Web App Firewall policies you want to globally bind.
7. Click **OK**. A message appears in the status bar, stating that the policy has been successfully bound.

Viewing a policy bindings

September 14, 2021

You can quickly check to determine what bindings are in place for any firewall policy by viewing the bindings in the GUI.

To view bindings for an Web App Firewall policy

1. Navigate to **Security > Citrix Web App Firewall > Policies > Firewall Policies**
2. In the details pane, select the policy that you want to check, and then click Show Bindings. The Binding Details for Policy: Policy message box is displayed, with a list of bindings for the selected policy.
3. Click **Close**.

Supplemental information about Web App Firewall policies

September 14, 2021

Following is supplemental information about particular aspects of Web App Firewall policies that system administrators who manage the Web App Firewall might need to know.

Correct but unexpected behavior

Web application security and modern websites are complex. In a number of scenarios, a Citrix ADC policy might cause the Web App Firewall to behave differently in certain situations than a user who

is familiar with policies would normally expect. Following are a number of cases where the Web App Firewall may behave in an unexpected fashion.

- **Request with a missing HTTP Host header and an absolute URL.** When a user sends a request, in the majority of cases the request URL is relative. That is, it takes as its starting point the Referer URL, the URL where the user's browser is located when it sends the request. If a request is sent without a Host header, and with a relative URL, the request is normally blocked both because it violates the HTTP specification and because a request that fails to specify the host can under some circumstances constitute an attack. If a request is sent with an absolute URL, however, even if the Host header is missing, the request bypasses the Web App Firewall and is forwarded to the web server. Although such a request violates the HTTP specification, it poses no possible threat because an absolute URL contains the host.

Auditing policies

September 14, 2021

Auditing policies determine the messages generated and logged during a Web App Firewall session. The messages are logged in SYSLOG format to the local NSLOG server or to an external logging server. Different types of messages are logged based on the level of logging selected.

To create an auditing policy, you must first create either an NSLOG server or a SYSLOG server. And then you create the policy and specify log type and the server to which logs are sent.

To create an auditing server by using the command line interface

You can create two different types of auditing server: an NSLOG server or a SYSLOG server. The command names are different, but the parameters for the commands are the same.

To create an auditing server, at the command prompt, type the following commands:

- `add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat (MMDDYYYY | DDMMYYYY)] [-logFacility <logFacility>] [-tcp (NONE | ALL)] [-acl (ENABLED | DISABLED)] [-timeZone (GMT_TIME | LOCAL_TIME)] [-userDefinedAuditlog (YES | NO)] [-appflowExport (ENABLED | DISABLED)]`
- `save ns config`

Example

The following example creates a syslog server named syslog1 at IP 10.124.67.91, with log levels of emergency, critical, and warning, log facility set to LOCAL1, that logs all TCP connections:


```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency
   critical warning -logFacility
LOCAL1 -tcp ALL
save ns config
<!--NeedCopy-->
```

To modify or remove an auditing server by using the command line interface

- To modify an auditing server, type the set audit <type> command, the name of the auditing server, and the parameters to be changed, with their new values.
- To remove an auditing server, type the rm audit <type> command and the name of the auditing server.

Example

The following example modifies the syslog server named syslog1 to add errors and alerts to the log level:

```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency
   critical warning alert error
-logFacility LOCAL1 -tcp ALL
save ns config
<!--NeedCopy-->
```

To create or configure an auditing server by using the GUI

1. Navigate to **Security** > **Citrix Web App Firewall** > **Policies** > **Auditing** > **Nslog**.
2. In the Nslog Auditing page, click **Servers** tab.
3. Do one of the following:
   - To add a new auditing server, click **Add**.
   - To modify an existing auditing server, select the server, and then click **Edit**.
4. In the **Create Auditing Server** page, set the following parameters:
   - Name
   - Server Type
   - IP Address
   - Port
   - Log Levels
   - Log Facility

- Date Format
- Time Zone
- TCP Logging
- ACL Logging
- User Configurable Log Messages
- AppFlow Logging
- Large Scale NAT Logging
- ALG messages logging
- Subscriber logging
- SSL Interception
- URL Filtering
- Content Inspection Logging

5. Click **Create** and **Close**.



To create an auditing policy by using the command line interface

You can create an NSLOG policy or a SYSLOG policy. The type of policy must match the type of server. The command names for the two types of policy are different, but the parameters for the commands are the same.

At the command prompt, type the following commands:

- `add audit syslogPolicy <name> <-rule > <action>`
- `save ns config`

Example

The following example creates a policy named syslogP1 that logs Web App Firewall traffic to a syslog server named syslog1.

```
add audit syslogPolicy syslogP1 rule "ns_true"action syslog1
save ns config
```

To configure an auditing policy by using the command line interface

At the command prompt, type the following commands:

- `set audit syslogPolicy <name> [-rule <expression>] [-action <string>]`
- `save ns config`

Example

The following example modifies the policy named syslogP1 to log Web App Firewall traffic to a syslog server named syslog2.

```
set audit syslogPolicy syslogP1 rule "ns_true"action syslog2
save ns config
```

To configure an auditing policy by using the GUI

1. Navigate to **Security** > **Citrix Web App Firewall** > **Policies**.
2. In the details pane, click **Audit Nslog Policy**.
3. In the Nslog Auditing page, click **Policies** tab and do one of the following:
   - To add a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Edit**.
4. In the **Create Auditing Nslog Policy** page, set the following parameters:

- Name
- Auditing Type
- Expression Type
- Server

5. Click **Create**.



Imports

September 14, 2021

Several Web App Firewall features make use of external files that you upload to the Web App Firewall when configuring it. Using the GUI, you manage those files in the Imports pane, which has four tabs corresponding to the four types of files you can import: HTML error objects, XML error objects, XML schemas, and Web Services Description Language (WSDL) files. Using the Citrix ADC command line, you can import these types of files, but you cannot export them.

HTML error object

When a user's connection to an HTML or Web 2.0 page is blocked, or a user asks for a non-existent HTML or Web 2.0 page, the Web App Firewall sends an HTML-based error response to the user's browser. When configuring which error response the Web App Firewall must use, you have two choices:

- You can configure a redirect URL, which can be hosted on any Web server to which users also have access. For example, if you have a custom error page on your Web server, 404.html, you can configure the Web App Firewall to redirect users to that page when a connection is blocked.
- You can configure an HTML error object, which is an HTML-based Web page that is hosted on the Web App Firewall itself. If you choose this option, you must upload the HTML error object to the Web App Firewall. You do that in the Imports pane, on the HTML Error Object tab.

The error object must be a standard HTML file that contains no non-HTML syntax except for Web App Firewall error object customization variables. It cannot contain any CGI scripts, server-parsed code, or PHP code. The customization variables enable you to embed troubleshooting information in the error object that the user receives when a request is blocked. While most requests that the Web App Firewall blocks are illegitimate, even a properly configured Web App Firewall can occasionally block legitimate requests, especially when you first deploy it or after you make significant changes to your protected websites. By embedding information in the error page, you provide the user with the information that he or she needs to give to the technical support person so that any issues can be fixed.

The Web App Firewall error page customization variables are:

- ${NS_TRANSACTION_ID}. The transaction ID that the Web App Firewall assigned to this transaction.
- ${NS_APPFW_SESSION_ID}. The Web App Firewall session ID.
- ${NS_APPFW_VIOLATION_CATEGORY}. The specific Web App Firewall security check or rule that was violated.
- ${NS_APPFW_VIOLATION_LOG}. The detailed error message associated with the violation.
- ${COOKIE The contents of the specified cookie. For <CookieName>, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.
  Note: If you have blocking enabled for the Cookie Consistency Check, any blocked cookies are not displayed on the error page because the Web App Firewall blocks them.

To use these variables, you embed them in the HTML or XML of the error page object as if they were an ordinary text string. When the error object is displayed to the user, for each customization variable the Web App Firewall substitutes the information to which the variable refers. An example HTML error page that uses custom variables is shown below.

```
1 <!doctype html public "-//w3c//dtd html 4.0//en"> <html> <head> <
    title>Page Not Accessible</title> </head> <body> <h1>Page Not
    Accessible</h1> <p>The page that you accessed is not available. You
     can:</p> <ul> <li>return to the <b><a href="[homePage]">home page
    </a></b>, re-establish your session, and try again, or,</li> <li>
    report this incident to the help desk via <b><a href="mailto:[
```



```
      helpDeskEmailAddress]">email</a></b> or by calling [
      helpDeskPhoneNumber].</li>  </ul>  <p>If you contact the help desk,
      please provide the following information:</p>  <table cellpadding=8
      width=80%>  <tr><th align="right" width=30%>Transaction ID:</th><td
      align="left" valign="top" width=70%>${
  NS_TRANSACTION_ID }
  </td></tr>  <tr><th align="right" width=30%>Session ID:</th><td align=
      "left" valign="top" width=70%>${
  NS_APPFW_SESSION_ID }
  </td></tr>  <tr><th align="right" width=30%>Violation Category:</th><
      td align="left" valign="top" width=70%>${
  NS_APPFW_VIOLATION_CATEGORY }
  </td></tr>  <tr><th align="right" width=30%>Violation Log:</th><td
      align="left" valign="top" width=70%>${
  NS_APPFW_VIOLATION_LOG }
  </td></tr>  <tr><th align="right" width=30%>Cookie Name:</th><td align
      ="left" valign="top" width=70%>${
  COOKIE("[cookieName]") }
  </td></tr>  </table>  <body>  <html>
<!--NeedCopy-->
```

To use this error page, copy it into a text or HTML editor. Substitute the appropriate local information for the following variables, which are enclosed in square brackets to distinguish them from the Citrix ADC variables. (Leave those unchanged.):

- [homePage]. The URL for your website's home page.
- [helpDeskEmailAddress]. The email address that you want users to use to report blocking incidents.
- [helpDeskPhoneNumber]. The phone number that you want users to call to report blocking incidents.
- [cookieName]. The name of the cookie whose contents you want to display on the error page.

XML error object

When a user's connection to an XML page is blocked, or a user asks for a nonexistent XML application, the Web App Firewall sends an XML-based error response to the user's browser. You configure the error response by uploading an XML-based error page to the Web App Firewall in the Imports Pane, on the XML Error Object tab. All XML error responses are hosted on the Web App Firewall. You cannot configure a redirect URL for XML applications.

> **Note**:
> You can use the same customization variables in an XML error object as in an HTML error object.

XML Schema

When the Web App Firewall performs a validation check on a user's request for an XML or Web 2.0 application, it can validate the request against the XML schema or design type document (DTD) for that application and reject any request that does not follow the schema or DTD. Both an XML schema and a DTD are standard XML configuration files that describe the structure of a specific type of XML document.

WSDL

When the Web App Firewall performs a validation check on a user's request for an XML SOAP-based web service, it can validate the request against the web services type definition (WSDL) file for that web service. A WSDL file is a standard XML SOAP configuration file that defines the elements of a specific XML SOAP web service.

Importing and exporting files

September 14, 2021

You can import HTML or XML error objects, XML schemas, DTDs, and WSDLs to the Web App Firewall by using the GUI or the command line. You can edit any of these files in a web-based text area after importing them, to make small changes directly on the Citrix ADC instead of having to make them on your computer and then reimport them. Finally, you can export any of these files to your computer, or delete any of these files, by using the GUI.

> **Note:**
>
> You cannot delete or export an imported file by using the command line.

To import a file by using the command line interface

At the command prompt, type the following commands:

- `import appfw htmlerrorpage <src> <name>`
- `<save> ns config`

Example

The following example imports an HTML error object from a file named error.html and assigns it the name HTMLError.


```
import htmlerrorpage error.html HTMLError
save ns config
<!--NeedCopy-->
```

To import a file by using the GUI

Before you attempt to import an XML schema, DTD, or WSDL file, or an HTML or XML error object from a network location, verify that the Citrix ADC can connect to the Internet or LAN computer where the file is located. Otherwise, you cannot import the file or object.

1. Navigate to **Security > Citrix Web App Firewall > Imports**.
2. Navigate to **Application Firewall** > **Imports**.
3. In the **Application Firewall Imports** pane, select the tab for the type of file you want to import, and then click **Add**.

   The tabs are HTML Error Page, XML Error Page, XML Schema or WSDL. The upload process is identical on all four tabs from the user point of view.
4. Fill in the dialog fields.
   - **Name**—A name for the imported object.
   - **Import From**—Choose the location of the HTML file, XML file, XML schema or WSDL that you want to import in the drop-down list:
     - **URL:** A web URL on a website accessible to the appliance.
     - **File:** A file on a local or networked hard disk or other storage device.
     - **Text:** Type or paste the text of the custom response directly into a text field in the GUI.

     The third text box changes to the appropriate value. The three possible values are provided below.
   - **URL**—Type the URL into the text box.
   - **File**—Type the path and filename to the HTML file directly, or click Browse and browse to the HTML file.
   - **Text**—The third field is removed, leaving a blank space.
5. Click **Continue**. The File Contents dialog is displayed. If you chose URL or File, the File Contents text box contains the HTML file that you specified. If you chose Text, the File Contents text box is empty.
6. If you chose Text, type or copy and paste the custom response HTML that you want to import.
7. Click **Done**.
8. To delete an object, select the object, and then click **Delete**.

To export a file by using the GUI

Before you attempt to export an XML schema, DTD, or WSDL file, or an HTML or XML error object, verify that the Web App Firewall appliance can access the computer where the file is to be saved. Otherwise, you cannot export the file.

1. Navigate to **Security** > **Web App Firewall** > **Imports**.
2. In the **Web App Firewall Imports** pane, select the tab for the type of file you want to export.

   The export process is identical on all four tabs from the user point of view.
3. Select the file that you want to export.
4. Expand the Action drop-down list, and select **Export**.
5. In the dialog box, choose **Save File** and click **OK**.
6. In the **Browse** dialog box, navigate to the local file system and directory where you want to save the exported file, and click **Save**.

To edit an HTML or XML Error Object in the GUI

You edit the text of HTML and XML error objects in the GUI without exporting and then reimporting them.

1. Navigate to **Security > Citrix Web App Firewall > Imports**, and then select the tab for the type of file that you want to modify.
2. Navigate to **Application Firewall** > **Imports**, and then select the tab for the type of file that you want to modify.
3. Select the file that you want to modify, and then click **Edit**.

   The text of the HTML or XML error object is displayed in a browser text area. You can modify the text by using the standard browser-based editing tools and methods for your browser.

   Note: The edit window is designed to allow you to make minor changes to your HTML or XML error object. To make extensive changes, you may prefer to export the error object to your local computer and use standard HTML or XML web page editing tools.
4. Click **OK**, and then click **Close**.

Global configuration

September 14, 2021

The Web App Firewall global configuration affects all profiles and policies. The Global Configuration items are:

- **Engine Settings.** A collection of global settings—session cookie name, session time-out, maximum session lifetime, logging header name, undefined profile, default profile, and import size limit—that pertain to all connections that the Web App Firewall processes, rather than to a specific subset of connections.
- **Confidential Fields.** A set of form fields in web forms that contain sensitive information that must not be logged to the Web App Firewall logs. Form fields such as password fields on a logon page or credit card information on a shopping cart checkout form are normally designated as confidential fields.
- **Field Types.** The list of web form field types used by the Field Formats security check. Each of these field types is defined by a PCRE-compliant regular expression that defines the type of data and the minimum/maximum length of data that must be allowed in that type of form field.
- **XML Content Types.** The list of content types recognized as XML and subjected to XML-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.
- **JSON Content Types.** The list of content types recognized as JSON and subjected to JSON-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.

Engine settings

September 14, 2021

The engine settings affect all requests and responses that the Citrix Web App Firewall processes. Following are the settings:

- **Cookie name**—The name of the cookie that stores the Citrix ADC session ID.
- **Session timeout**—The maximum inactive period allowed. If a user session shows no activity for this length of time, the session is terminated and the user is required to reestablish it by visiting a designated start page.
- **Cookie post-encrypts prefix**—The string that precedes the encrypted portion of any encrypted cookies.
- **Maximum session lifetime**—The maximum amount of time, in seconds, that a session is allowed to remain live. After this period is reached, the session is terminated and the user is required to reestablish it by visiting a designated start page. This setting cannot be less than the session timeout. To disable this setting, so that there is no maximum session lifetime, set the value to zero (0).
- **Logging header name**—The name of the HTTP header that holds the Client IP, for logging.

- **Undefined profile**—The profile applied when the corresponding policy action evaluates as undefined.
- **Default profile**—The profile applied to connections that do not match a policy.
- **Import size limit**—The maximum byte count of all files imported to the appliance, including signatures, WSDLs, schemas, HTML, and XML error pages. During an import, if the size of the imported object causes the cumulative count of all imported files to exceed the configured limit, the import operation fails. And the appliance displays the following error message: "*ERROR: Import failed - exceeding the configured total size limit on the imported objects*".
- **Learn message rate limit**—The maximum number of requests and responses per second that the learning engine is to process. Any additional requests or responses over this limit are not sent to the learning engine.
- **Entity decoding**—Decode HTML entities when running Web App Firewall checks.
- **Log malformed request**—Enable logging of malformed HTTP requests.
- **Use configurable secret key**—Use a configurable secret key for Web App Firewall operations. This secret key is used for signing and verifying data. When "useConfigurableSecretKey" is turned ON, you must use the key enabled in the "set ns encryptionParams" parameter.
- **Reset learned data**—Remove all learned data from the Web App Firewall. Restarts the learning process by collecting fresh data.

Two settings, *Reset Learned Data* and *Signatures Auto-Update*, are found in different places depending on whether you use the command interface or the Citrix ADC GUI to configure your Citrix Web App Firewall. When using the command interface, you configure Reset Learned Data by using the reset appfw learning data command. This takes no parameters and has no other functions. You can configure the signature auto-Update in the set appfw settings command. The -signatureAutoUpdate parameter enables or disables auto-updating of the signatures, and -signatureUrl configures the URL which hosts the updated signatures file.

When using the Citrix ADC GUI, you configure Reset Learned Data in **Security** > **Citrix Web App Firewall** > **Engine Settings**. The **Reset Learned Data** option is at the bottom of the dialog box. You configure Signatures Auto-Update for each set of signatures in **Security** > **Citrix Web App Firewall** > **Signatures**, by selecting the signatures file, clicking the right mouse button and selecting **Auto Update Settings**.

Normally, the default values for the **Web App Firewall** settings are correct. If the default settings cause a conflict with other servers or cause premature disconnection of your users, however, you have to modify them.

The **Web App Firewall** session limit is configurable using the following command:

```
> set appfw settings -sessionLimit 500000

Done

```



```
Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

To configure engine settings by using the command line interface

At the command prompt, type the following commands:

- `set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger>] [-sessionLifetime <positiveInteger>][-clientIPLoggingHeader <headerName>] [-undefaction <profileName>] [-defaultProfile <profileName>] [-importSizeLimit <positiveInteger>] [-logMalformedReq (ON | OFF)] [-signatureAutoUpdate (ON | OFF)] [-signatureUrl <expression>] [-cookiePostEncryptPrefix <string>] [-entityDecoding (ON | OFF)] [-useConfigurableSecretKey (ON | OFF)][-learnRateLimit <positiveInteger>]`
- `save ns config`

Example

```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout
    3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -
    undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096
save ns config
<!--NeedCopy-->
```

To configure engine settings by using the Citrix ADC GUI

1. Navigate to **Security** > **Citrix Web App Firewall**
2. In the details pane, click **Change Engine Settings** under **Settings**.
3. In the **Web App Firewall Engine Settings** dialog box, set the following parameters:
   - Cookie Name
   - Session Timeout
   - Cookie Post Encrypt Prefix
   - Maximum Session Lifetime
   - Logging Header Name
   - Undefined Profile
   - Default Profile

- Import Size Limit
- Learn Messages Rate Limit
- Entity Decoding
- Log Malformed Request
- Use Secret Key
- Learn Message Rate Limit
- Signatures Auto Update

4. Click **OK**.



Confidential fields

September 14, 2021

You can designate web-form fields as confidential to protect the information users type into them. Normally, any information a user types into a web form on one of your protected web servers is logged in the Citrix ADC logs. The information typed into a web-form field designated as confidential, however, is not logged. That information is saved only where the website is configured to save such data, normally in a secure database.

Common types of information that you may want to protect with a confidential field designation include:

- Passwords
- Credit card numbers, validation codes, and expiration dates
- Social security numbers
- Tax ID numbers
- Home addresses
- Private telephone numbers

In addition to being good practice, proper use of confidential field designations may be necessary for PCI-DSS compliance on ecommerce servers, HIPAA compliance on servers that manage medical information in the United States, and compliance with other data protection standards.

> **Important**:
>
> In the following two cases, the Confidential Field designation does not function as expected:

- If a Web form has either a confidential field or an action URL longer than 256 characters, the field or action URL is truncated in the Citrix ADC logs.
- With certain SSL transactions, the logs are truncated if either the confidential field or the action URL is longer than 127 characters.

In either of these cases, the Web App Firewall masks a fifteen-character string with the letter “x,” instead of the normal eight character string. To ensure that any confidential information is removed, the user must use form field name and action URL expressions that match the first 256, or (in cases where SSL is used) the first 127 characters.

To configure your Web App Firewall to treat a web-form field on a protected website as confidential, you add that field to the Confidential Fields list. You can enter the field name as a string, or you can enter a PCRE-compatible regular expression specifying one or more fields. You can enable the confidential-field designation when you add the field, or you can modify the designation later.

To add a confidential field by using the command line interface

At the command prompt, type the following commands:

- `add appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX )] [-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

Example

The following example adds all web form fields whose names begin withPassword to the confidential fields list.

```
add appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi|htm|html|htp|js|php)" -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

To modify a confidential field by using the command line interface

At the command prompt, type the following commands:

- `set appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX ) ][-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

Example

The following example modifies the confidential field designation to add a comment.

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi|htm|html|htp|js|php)" -comment "Protect password fields." -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

To remove a confidential field by using the command line interface

At the command prompt, type the following commands:

- `rm appfw confidField <fieldName> <url>`
- `save ns config`

To configure a confidential field by using the GUI

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Settings**, click **Manage Confidential Fields**.
3. In the Manage Confidential Fields dialog box, do one of the following:
    - To add a new form field to the list, click Add.

- To change an existing confidential field designation, select the field, and then click **Edit**. The **Web App Firewall Confidential Fields** dialog box appears.

  > **Note**:
  > If you select an existing confidential field designation and then click **Add**, the **Create Confidential Form Field** dialog box displays the information for that confidential field. You can modify that information to create your new confidential field.

4. In the dialog box, fill out the elements. They are:
   - **Enabled check box.** Select or clear to enable/disable this confidential field designation.
   - **Is form field name a regular expression check box.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **Field Name.** Enter a literal string or PCRE-format regular expression that either represents a specific field name or that matches multiple fields with names that follow a pattern.
   - **Action URL.** Enter a literal URL or a regular expression that defines one or more URLs of the web page(s) on which the web form(s) that contains the confidential field are located.
   - **Comments.** Enter a comment. Optional.
5. Click **Create** or **OK**.
6. To remove a confidential field designation from the confidential fields list, select the confidential field listing you want to remove, then click Remove to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding, modifying, and removing confidential field designations, click **Close**.

Examples

Following are some regular expressions that define form field names that you might find useful:

- `^passwd_ (Applies confidential-field status to all field names that begin with the “passwd_” string.)`
- `^((\[0-9a-zA-Z._-]*||\\x\[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_ (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)`

Following are some regular expressions that define specific URL types that you might find useful. Substitute your own web host(s) and domain(s) for those in the examples.

- If the web form appears on multiple web pages on the web host www.example.com, but all of those web pages are named logon.pl?, you can use the following regular expression:

```
1 https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon
     [.]pl\?
2 <!--NeedCopy-->
```


- If the web form appears on multiple web pages on the web host www.example-español.com, which contains the n-tilde (ñ) special character, you can use the following regular expression, which represents the n-tilde special character as an encoded UTF-8 string containing C3 B1, the hexadecimal code assigned to that character in the UTF-8 charset:

```
https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)* logon[.]pl\?
<!--NeedCopy-->
```

- If the web form containing query.pl appears on multiple web pages on different hosts within the example.com domain, you can use the following regular expression:

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
<!--NeedCopy-->
```

- If the web form containing query.pl appears on multiple web pages on different hosts in different domains, you can use the following regular expression:

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{
2,6 }
/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
<!--NeedCopy-->
```

- If the web form appears on multiple web pages on the web host www.example.com, but all of those web pages are named logon.pl?, you can use the following regular expression:

```
https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
<!--NeedCopy-->
```

Field types

September 14, 2021

A field type is a PCRE-format regular expression that defines a particular data format and minimum/-maximum data lengths for a form field in a web form. Field types are used in the Field Formats check.

The Web App Firewall comes with several default field types, which are:

- integer. A string of any length consisting of numbers only, without a decimal point, and with an optional preceding minus sign (-).

- alpha. A string of any length consisting of letters only.
- alphanum. A string of any length consisting of letters and/or numbers.
- nohtml. A string of any length consisting of characters, including punctuation and spaces, that does not contain HTML symbols or queries.
- any. Anything at all.

> **Important:**
>
> Assigning the any field type as the default field type, or to a field, allows active scripts, SQL commands, and other possibly dangerous content to be sent to your protected websites and applications in that form field. You must use the any type sparingly, if you use it at all.

You can also add your own field types to the Field Types list. For example, you might want to add a field type for a social security number, postal code, or phone number in your country. You might also want to add a field type for a customer identification number or store credit card number.

To add a field type to the Field Types list, you enter the field name as a literal string or PCRE-format regular expression.

To add a field type by using the command line interface

At the command prompt, type the following commands:

- `add appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

Example

The following example adds a field type named SSN that matches US Social Security numbers to the Field Types list, and sets its priority to 1.

```
add appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1
save ns config
<!--NeedCopy-->
```


To modify a field type by using the command line interface

At the command prompt, type the following commands:

- `set appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

Example

The following example modifies the field type to add a comment.

```
set appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1 -comment "US Social Security Number"
save ns config
<!--NeedCopy-->
```

To remove a field type by using the command line interface

At the command prompt, type the following commands:

- `>rm appfw fieldType <name>`
- `save ns config`

To configure a field type by using the GUI

1. Navigate to Security > Application Firewall.
2. In the details pane, under **Settings**, click **Manage Field Types**.
3. In the **Manage Field Types** dialog box, do one of the following:
   - To add a new field type to the list, click **Add**.
   - To change an existing field type, select the field type, and then click **Edit**.
     The **Configure Field Type** dialog box appears.

> **Note:**
>
> If you select an existing field type designation and then click **Add**, the dialog box displays the information for that field type. You can modify that information to create your new

> field type.

4. In the dialog box, fill out the elements. They are:
   - Name
   - Regular Expression
   - Priority
   - Comment
5. Click Create or OK.
6. To remove a field type from the Field Types list, select the field type listing you want to remove, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding, modifying, and removing field types, click **Close**.

Examples

Following are some regular expressions for field types that you might find useful:

`^[1-9][0-9]{ 2,2 } -[0-9 } { 2,2 } -[0-9]{ 4,4 } $` U.S. Social Security numbers

`^\[A-C\]\[0-9\]{ 7,7 } $` California driver's license numbers

`^+[0-9]{ 1,3 } [0-9()-]{ 1,40 } $` International phone numbers with country codes

`^[0-9]{ 5,5 } -[0-9]{ 4,4 } $` U.S. ZIP code numbers

`^[0-9A-Za-z][0-9A-Za-z.+_-]{ 0,25 } @([0-9A-Za-z][0-9A-Za-z_-]*[.]){ 1,4 } [A-Za-z]{ 2,6 } $` Email addresses

XML content types

September 14, 2021

By default, the Web App Firewall treats files that follow certain naming conventions as XML. You can configure the Web App Firewall to examine web content for additional strings or patterns that indicate that those files are XML files. This can ensure that the Web App Firewall recognizes all XML content on your site, even if certain XML content does not follow normal XML naming conventions, ensuring that XML content is subjected to XML security checks.

To configure the XML content types, you add the appropriate patterns to the XML Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing XML content types patterns.

To add an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- `add appfw XMLContentType <XMLContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

Example

The following example adds the pattern .*/xml to the XML Content Types list and designates it as a regular expression.

```
1 add appfw XMLContentType ".*/xml" -isRegex REGEX
2 <!--NeedCopy-->
```

To remove an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- `rm appfw XMLContentType <XMLContenttypevalue>`
- `save ns config`

To configure the XML content type list by using the GUI

1. Navigate to **Security** > **Web App Firewall**.
2. In the details pane, under **Settings**, click **Manage XML Content Types**.
3. In the **Manage XML Content Types** dialog box, do one of the following:
   - To add a new XML content type, click Add.
   - To modify an existing XML content type, select that type and then click Edit.
     The Configure Web App Firewall XML Content Type dialog appears. Note: If you select an existing XML content type pattern and then click Add, the dialog box displays the information for that XML content type pattern. You can modify that information to create your new XML content type pattern.
4. In the dialog box, fill out the elements. They are:
   - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **XML Content Type** Enter a literal string or PCRE-format regular expression that matches the XML content type pattern that you want to add.
5. Click **Create**.
6. To remove an XML content type pattern from the list, select it, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click **Close**.

JSON content types

September 14, 2021

By default, the Web App Firewall treats files with the content type “application/json” as JSON files.The default setting enables the Web App Firewall to recognize JSON content in requests and responses, and to handle that content appropriately.

You can configure the Web App Firewall to examine web content for additional strings or patterns that indicate that those files are JSON files. This can ensure that the Web App Firewall recognizes all JSON content on your site, even if certain JSON content does not follow normal JSON naming conventions, ensuring that JSON content is subjected to JSON security checks.

To configure the JSON content types, you add the appropriate patterns to the JSON Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing JSON content types patterns.

To add a JSON content type pattern by using the command line interface

At the command prompt, type the following commands:

- `add appfw JSONContentType <JSONContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

Example

The following example adds the pattern .*/json to the JSON Content Types list and designates it as a regular expression.

```
1 add appfw JSONContentType ".*/json" -isRegex REGEX
2 <!--NeedCopy-->
```

To configure the JSON content type list by using the GUI

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Settings**, click **Manage JSON Content Types**.
3. In the Manage JSON Content Types dialog box, do one of the following:
   - To add a new JSON content type, click Add.
   - To modify an existing JSON content type, select that type and then click Edit.
     The Configure Web App Firewall JSON Content Type dialog appears.
     Note: If you select an existing JSON content type pattern and then click Add, the dialog

box displays the information for that JSON content type pattern. You can modify that information to create your new JSON content type pattern.

4. In the dialog box, fill out the elements. They are:
   - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **JSON Content Type** Enter a literal string or PCRE-format regular expression that matches the JSON content type pattern that you want to add.
5. Click **Create** or **OK**.
6. To remove a JSON content type pattern from the list, select it, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click **Close**.

Statistics and reports

September 14, 2021

The information maintained in the logs and statistics, and displayed in the reports, provides important guidance for configuring and maintaining the Web App Firewall.

The Web App Firewall statistics

When you enable the statistics action for Web App Firewall signatures or security checks, the Web App Firewall maintains information about connections that match that signature or security check. You can view the accumulated statistics information on the
Monitoring tab by selecting one of the following choices in the Select Group list box:

- **Web App Firewall.** A summary of all statistics information gathered by your Web App Firewall appliance for all profiles.
- **Web App Firewall (per profile).** The same information, but displayed per-profile rather than summarized.

You can use this information to monitor how your Web App Firewall is operating and determine whether there is any abnormal activity or abnormal amounts of hits on a signature or security check. If you see such a pattern of abnormal activity, you can check the logs for that signature or security check to diagnose and take corrective action.

Relaxation hit statistical counter

Based on the relaxation that is applied on the violated traffic, you can also display statistical details such as number of times a violation is occurring on the appliance, number of relaxation rules applied at the time of violation, and its last applied timestamp. By performing this, the centralized learning

engine can automatically deletes unused or redundant relaxation bindings. For more information, see WAF Learn Engine topic.

The relaxation hit statistical counter is available only for the following security checks.

- Starturl
- Denyurl
- Cross-site scripting
- SQL Injection

To display statistics for relaxation rule hit counters by using the CLI

At the command prompt, type:

```
stat appfw profile p1
```

Example:

```
stat appfw profile p1 -fullvalues
```

Starturl Rules Statistics

Rule	hits	Rate	last hit time
87a4…51177	0	0	Thu … 1970
5b83…dc12a	0	0	Thu … 1970
12345	0	0	Thu … 1970

To display statistics for relaxation rule hit counters by using the GUI

Complete the following steps to view the relaxation rule hit counter statistics:

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select a **Web App Firewall profile** and click **Statistics**.
3. The **Citrix Web App Firewall Statistics** page displays the statistics details.
4. You can select Tabular View or switch to Graphical View to display the data in a tabular or graphical format.

The Web App Firewall Reports

The Web App Firewall reports provide information about your Web App Firewall configuration and how it is handling traffic for your protected websites.

The PCI DSS report

The Payment Card Industry (PCI) Data Security Standard (DSS), version 1.2, consists of 12 security criteria that most credit card companies require businesses who accept online payments via credit and debit cards to meet. These criteria are designed to prevent identity theft, hacking, and other types of fraud. If an internet service provider does not meet the PCI DSS criteria, that ISP or merchant might lose authorization to accept credit card payments through the website.

ISPs and online merchants prove that they are in compliance with PCI DSS by having an audit conducted by a PCI DSS Qualified Security Assessor (QSA) Company. The PCI DSS report is designed to assist them both before and during the audit. Before the audit, it shows which Web App Firewall settings are relevant to PCI DSS, how they must be configured, and (most important) whether your current Web App Firewall configuration meets the standard. During the audit, the report can be used to demonstrate compliance with relevant PCI DSS criteria.

The PCI DSS report consists of a list of those criteria that are relevant to your Web App Firewall configuration. Under each criterion, it lists your current configuration options, indicates whether your current configuration complies with the PCI DSS criterion, and explains how to configure the Web App Firewall so that your protected websites are in compliance with the criterion.

The PCI DSS report is located under **System > Reports**. To generate the report as an Adobe PDF file, click Generate PCI DSS Report. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

> **Note:**
>
> To view this and other reports, you must have the Adobe Reader program installed on your computer.

The PCI DSS report consists of the following sections:

- **Description.** A description of the PCI DSS Compliance Summary report.
- **Firewall License and Feature Status.** Tells you whether the Web App Firewall is licensed and enabled on your Citrix ADC appliance.
- **Executive Summary.** A table that lists the PCI DSS criteria and tells you which of those criteria are relevant to the Web App Firewall.
- **Detailed PCI DSS Criteria Information.** For each PCI DSS criterion that is relevant to your Web App Firewall configuration, the PCI DSS report provides a section that contains information about whether your configuration is in compliance and, if it is not, how to bring it into compliance.
- **Configuration.** Data for individual profiles, which you access either by clicking Web App Firewall Configuration at the top of the report, or directly from the Reports pane. The Web App Firewall Configuration report is the same as the PCI DSS report, with the PCI DSS-specific summary

omitted.

The Web App Firewall configuration report

The Web App Firewall Configuration report is located under **System > Reports**. To display it, click **Generate Web App Firewall Configuration Report**. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

The Web App Firewall Configuration report starts with a Summary page, which consists of the following sections:

- **Web App Firewall Policies.** A table that lists your current Web App Firewall policies, showing the policy name, the content of the policy, the action (or profile) it is associated with, and global binding information.
- **Web App Firewall Profiles.** A table that lists your current Web App Firewall profiles and indicates which policy each profile is associated with. If a profile is not associated with a policy, the table displays INACTIVE in that location.

To download all report pages for all policies, at the top of the Profiles Summary page click **Download All Profiles**. You display the report page for each individual profile by selecting that profile in the table at the bottom of the screen. The Profile page for an individual profile shows whether each check action is enabled or disabled for each check, and the other configuration settings for the check.

To download a PDF file containing the PCI DSS report page for the current profile, click **Download Current Profile** at the top of the page. To return to the Profiles Summary page, click **Web App Firewall Profiles**. To go back to the main page, click **Home**. You can refresh the PCI DSS report at any time by clicking **Refresh** in the upper right corner of the browser.

Web App Firewall logs

September 14, 2021

The Web App Firewall generated log messages can be quite useful for keeping track of the configurational changes, Web App Firewall policy invocations, and security check violations.

When the log action is enabled for security checks or signatures, the resulting log messages provide information about the requests and responses that the Web App Firewall has observed while protecting your websites and applications. The most important information is the action taken by the Web App Firewall when a signature or a security check violation was observed. For some security checks, the log message can provide additional useful information, such as the location and the detected pattern that triggered the violation. You can deploy security checks in non-block mode and monitor the logs to determine whether the transactions that are triggering security violations are valid transactions

(false positives). If they are, you can either remove, or reconfigure the signature or security checks, deploy relaxations, or take other appropriate measures to mitigate the false positives before you enable blocking for that signature or security check. An excessive increase in the number of violation messages in logs can indicate a surge in malicious requests. This can alert you that your application might be under attack to exploit a specific vulnerability that is detected and thwarted by Web App Firewall protections.

> **Note:**
>
> Citrix Web App Firewall logging must be used only with external SYSLOG servers.

Citrix ADC (Native) format logs

The Web App Firewall uses the Citrix ADC format logs (also called native format logs) by default. These logs have the same format as those generated by other Citrix ADC features. Each log contains the following fields:

- Timestamp. Date and time when the connection occurred.
- Severity. Severity level of the log.
- Module. Citrix ADC module that generated the log entry.
- Event Type. Type of event, such as signature violation or security check violation.
- Event ID. ID assigned to the event.
- Client IP. IP address of the user whose connection was logged.
- Transaction ID. ID assigned to the transaction that caused the log.
- Session ID. ID assigned to the user session that caused the log.
- Message. The log message. Contains information identifying the signature or security check that triggered the log entry.

You can search for any of these fields, or any combination of information from different fields. Your selection is limited only by the capabilities of the tools you use to view the logs. You can observe the Web App Firewall log messages in the GUI by accessing the Citrix ADC syslog viewer, or you can manually connect to the Citrix ADC appliance and access logs from the command line interface, or you can drop into shell and tail the logs directly from the /var/log/folder.

Example of a Native Format Log message

```
1 Jun 22 19:14:37 <local0.info> 10.217.31.98 06/22/2015:19:14:37 GMT ns
     0-PPE-1 :
2 default APPFW APPFW_cross-site scripting 60 0 :  10.217.253.62 616-PPE1
      y/3upt2K8ySWWId3Kavbxyni7Rw0000
3 pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
4 12345&drinking_pref=on&text_area=%3Cscript%3E%0D%0A&loginButton=
     ClickToLogin&as_sfid=
5 AAAAAAWEXcNQLlSokNmqaYF6dvfqlChNzSMsdyO9JXOJomm2v
```



```
6 BwAMOqZIChv21EcgBc3rexIUcfm0vckKlsgoOeC_BArx1Ic4NLxxkWMtrJe4H7SOfkiv9NL7AG4juPIan

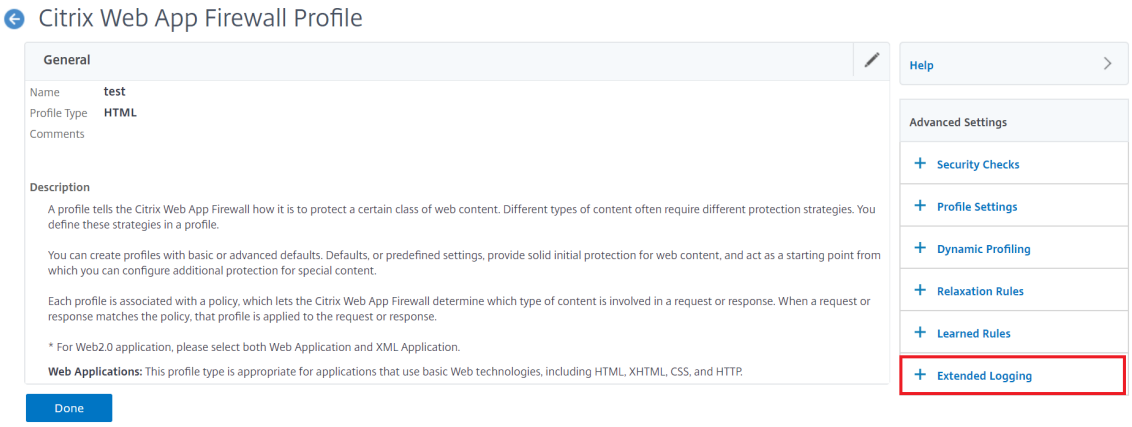
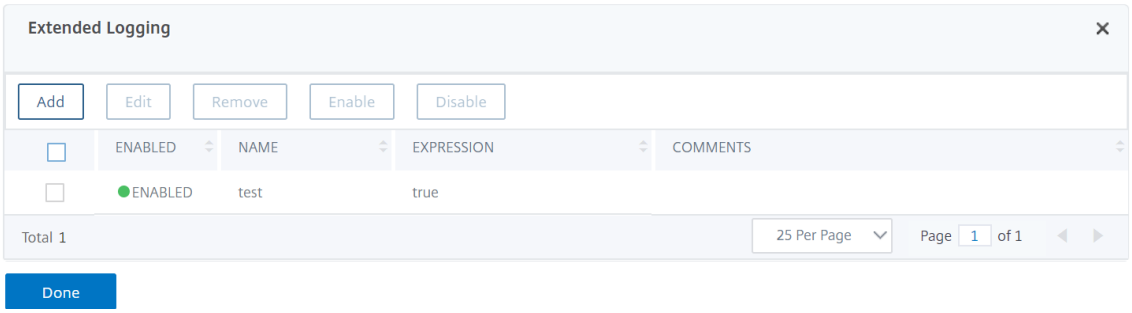
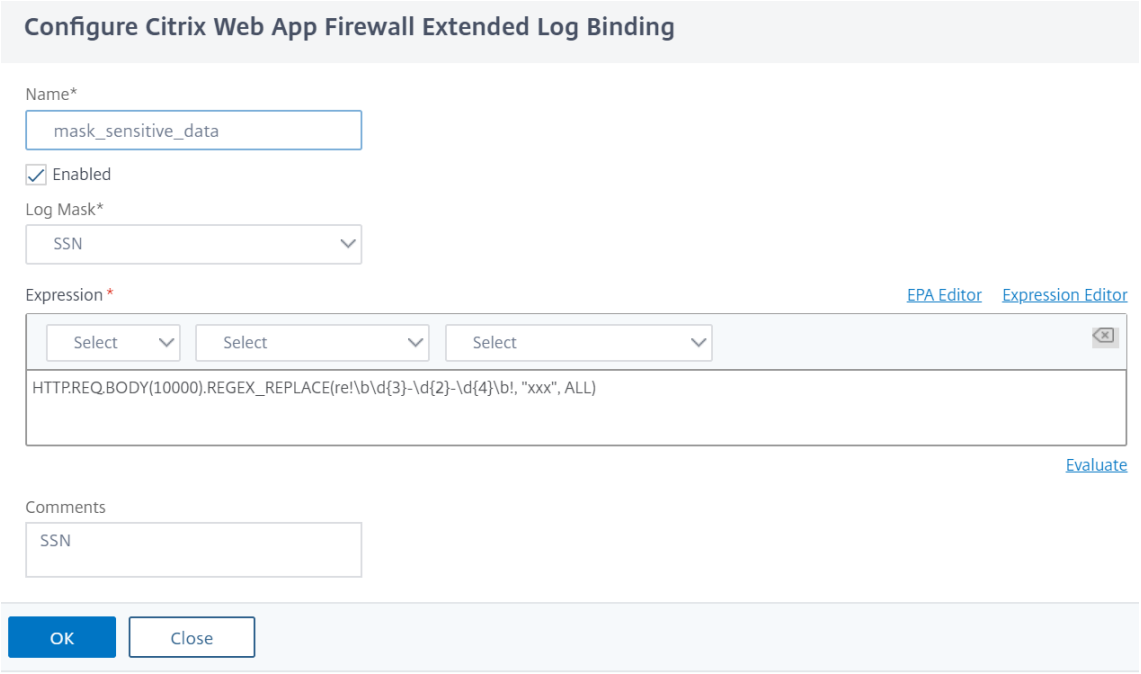
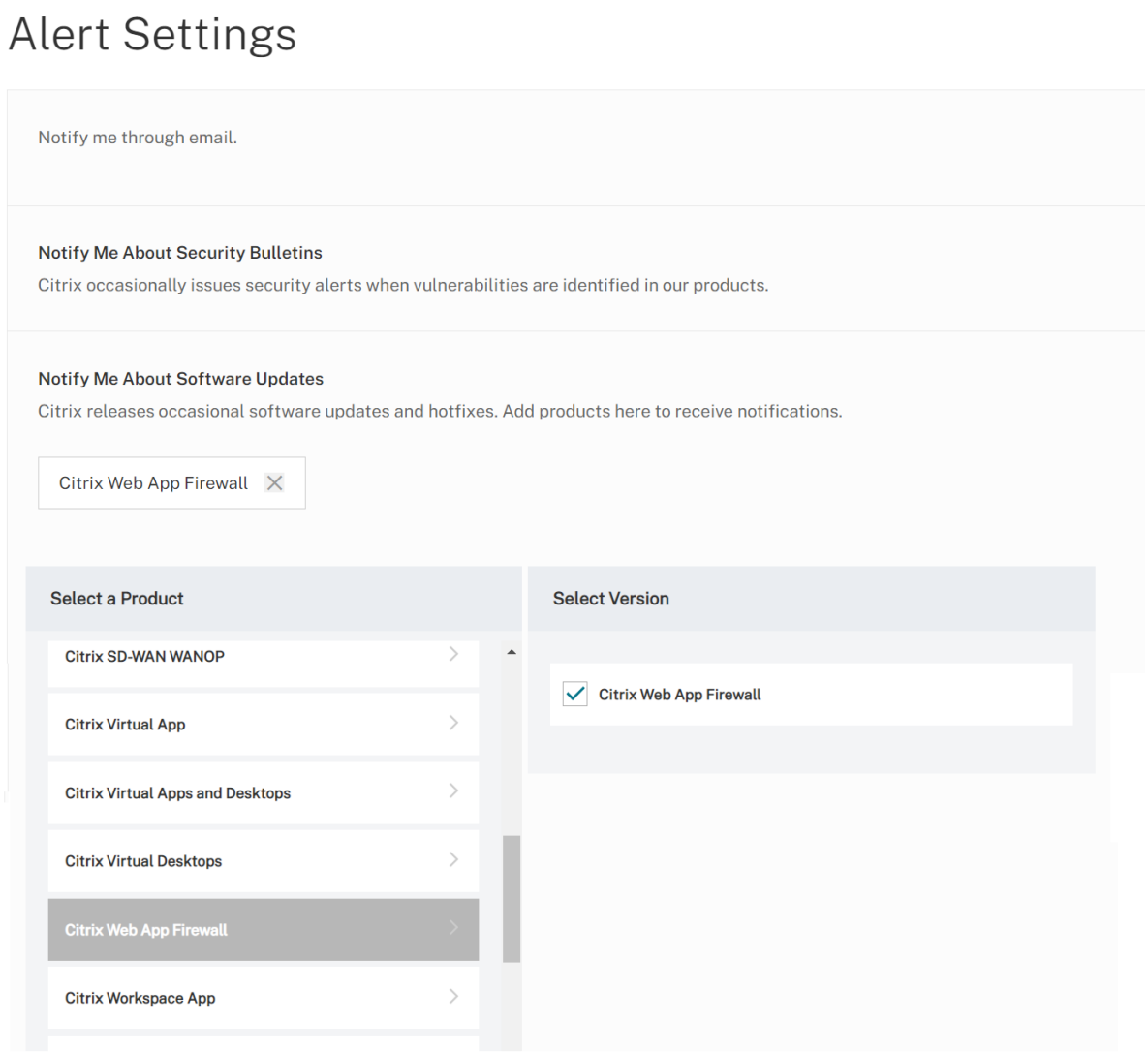
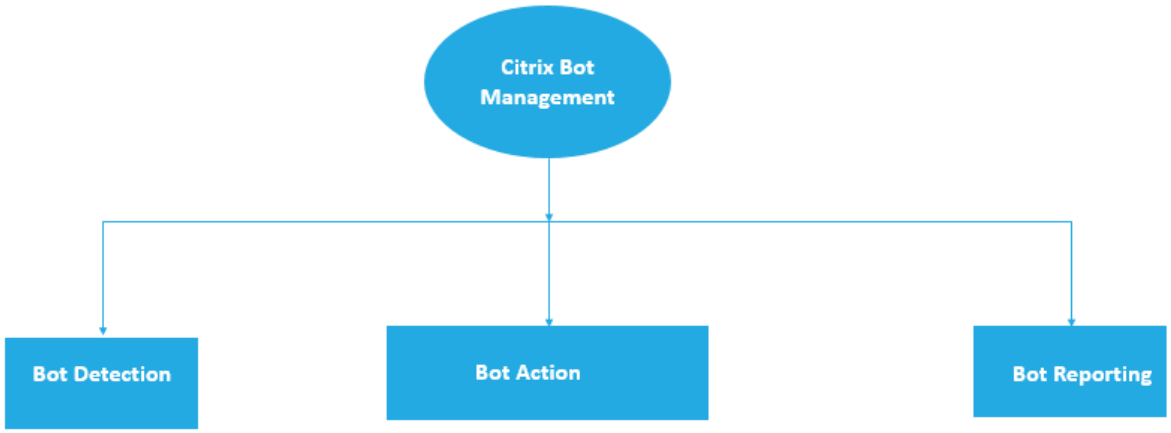
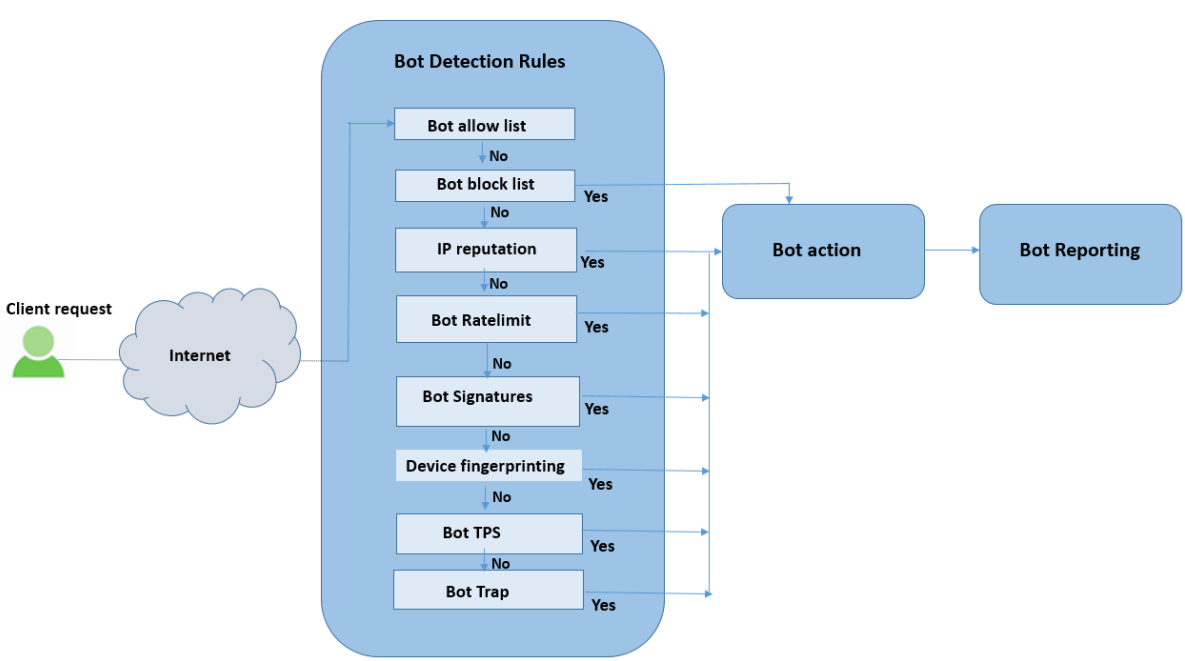
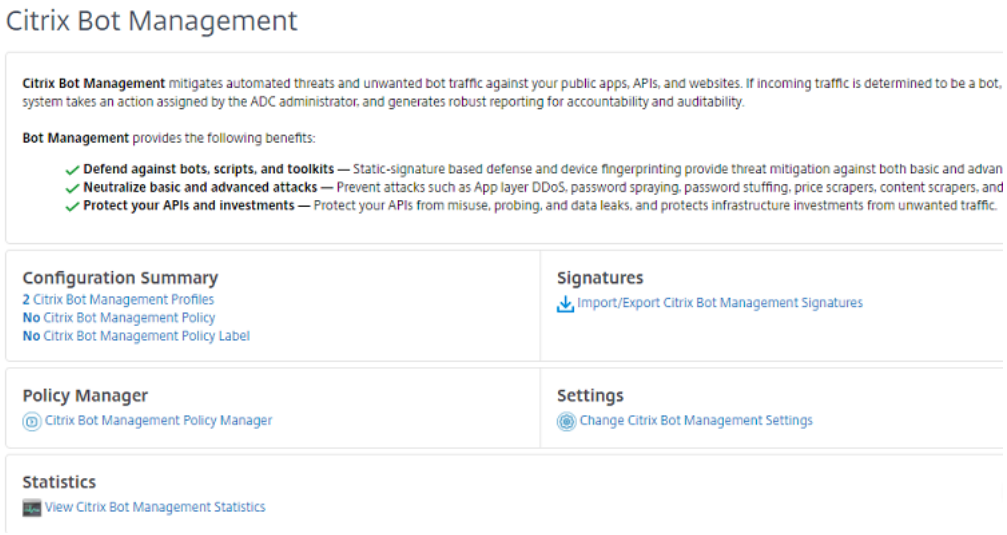
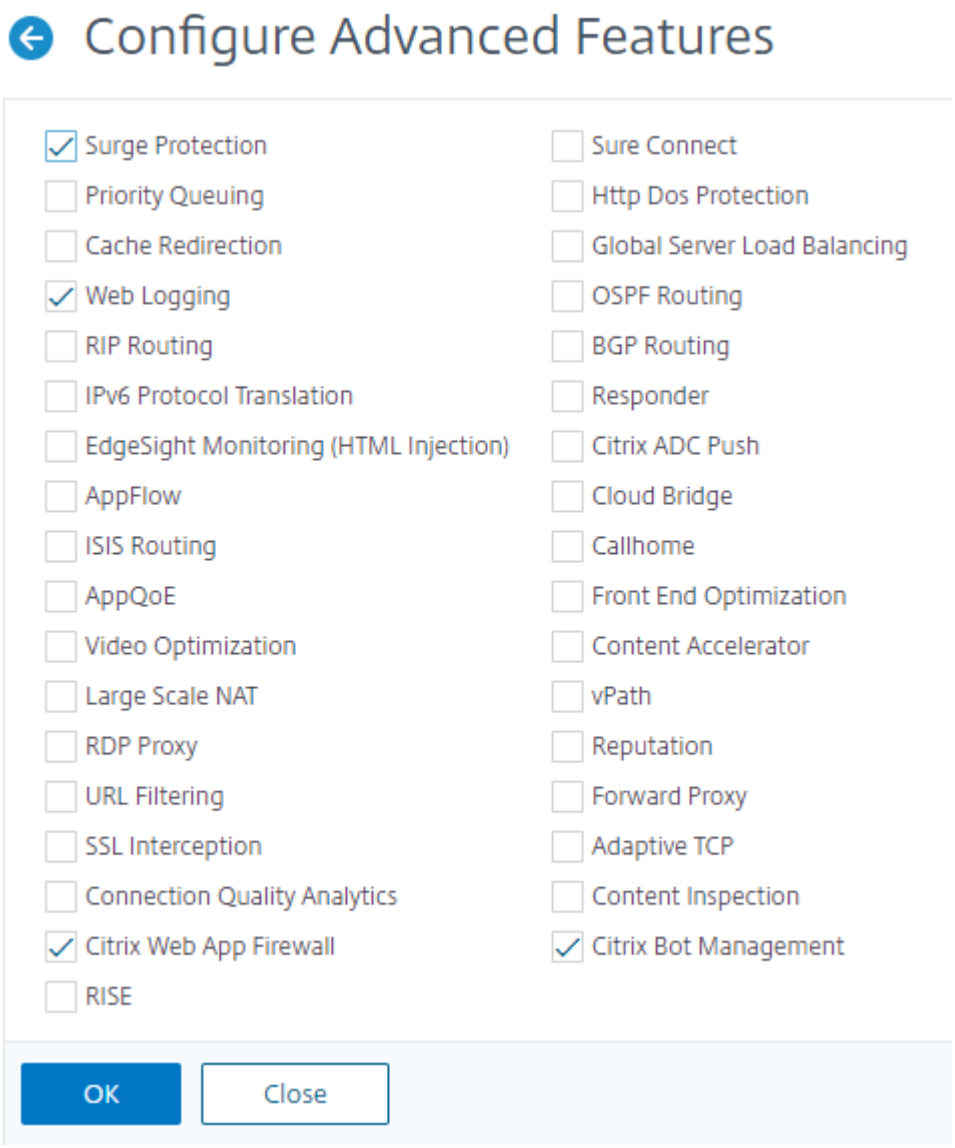
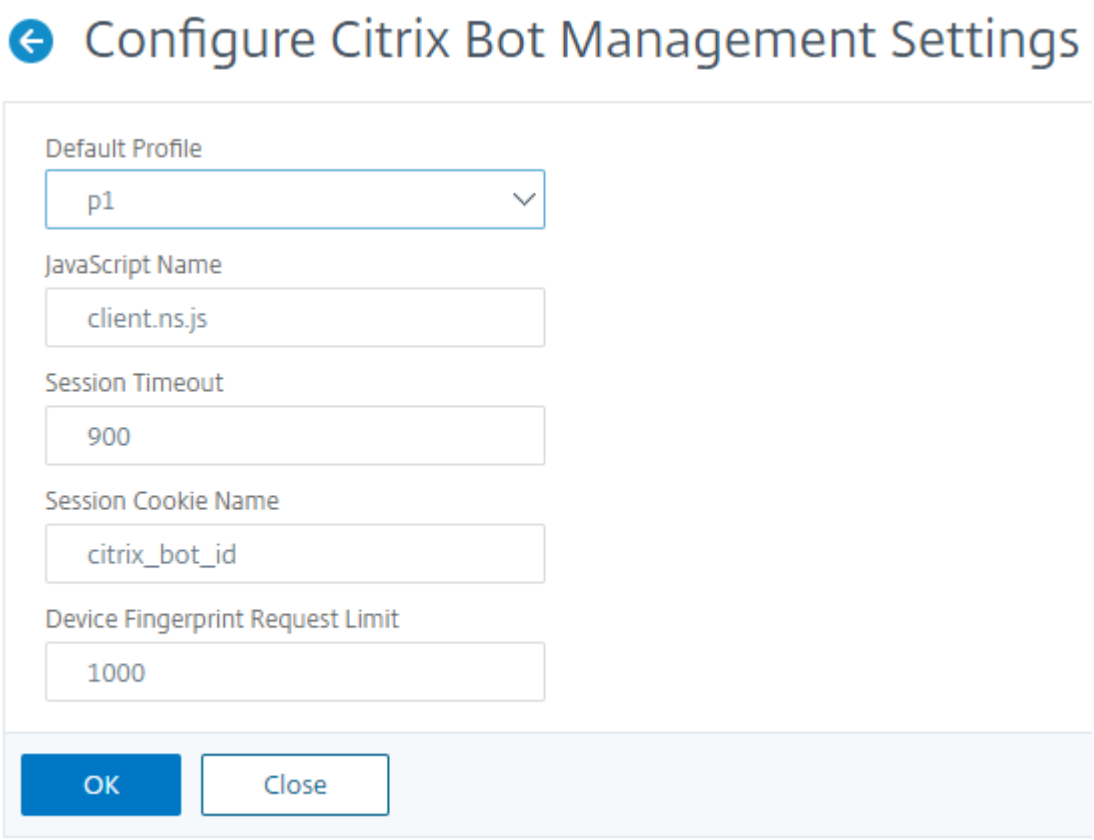
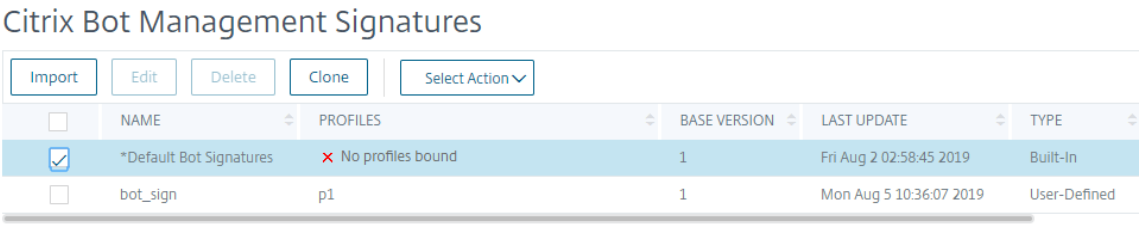
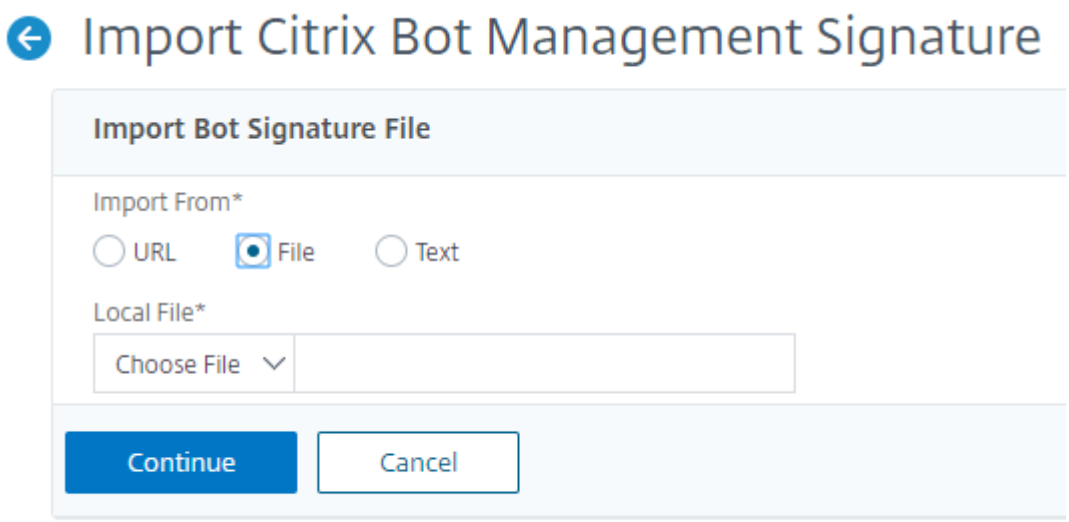
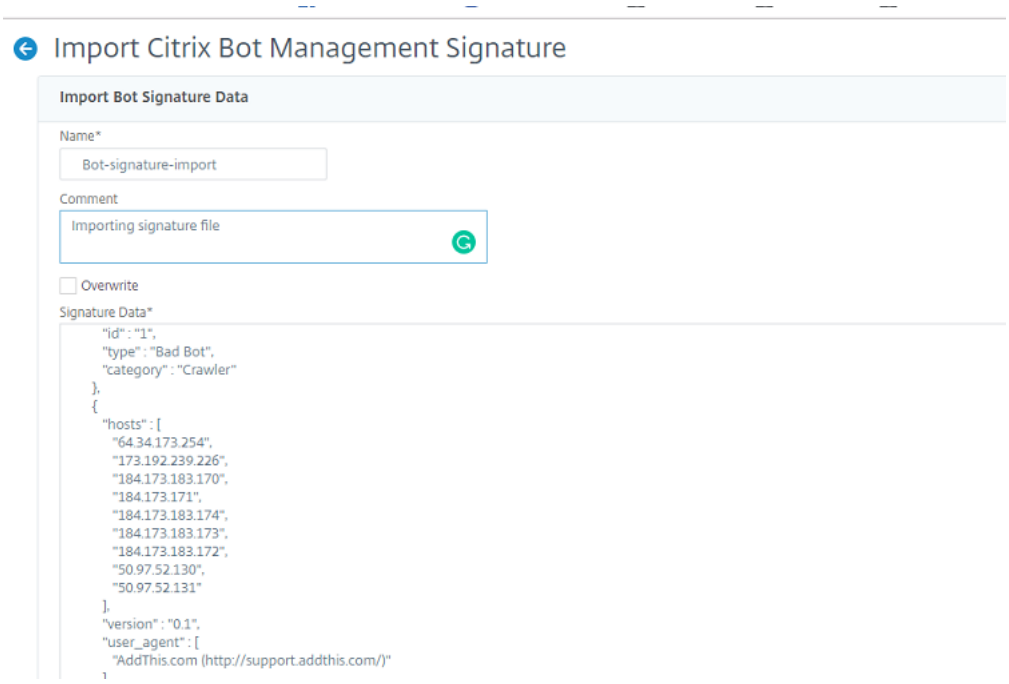
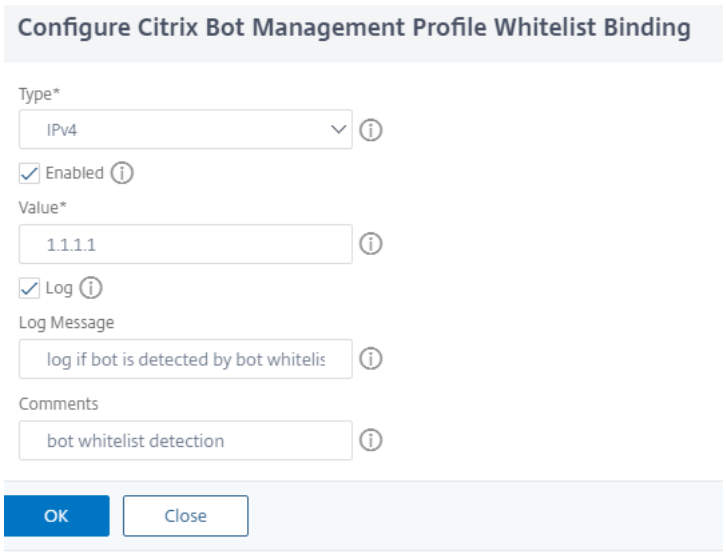
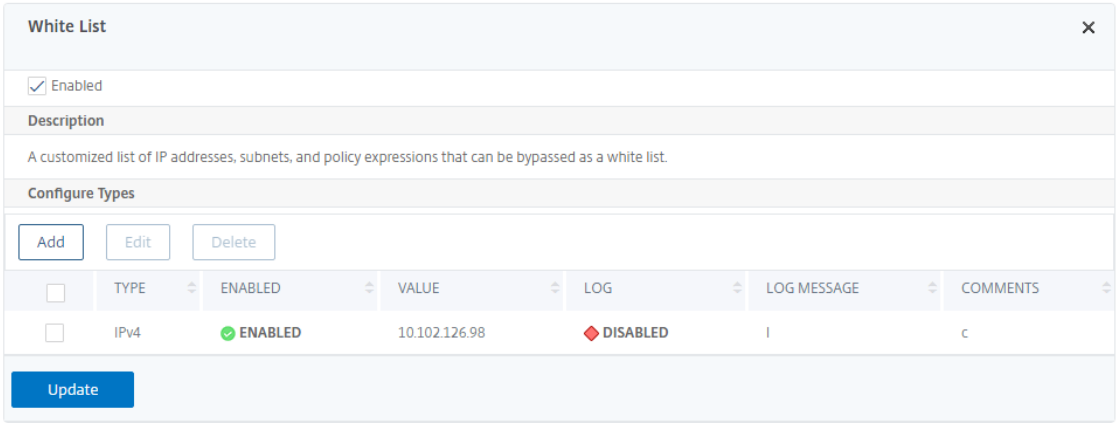
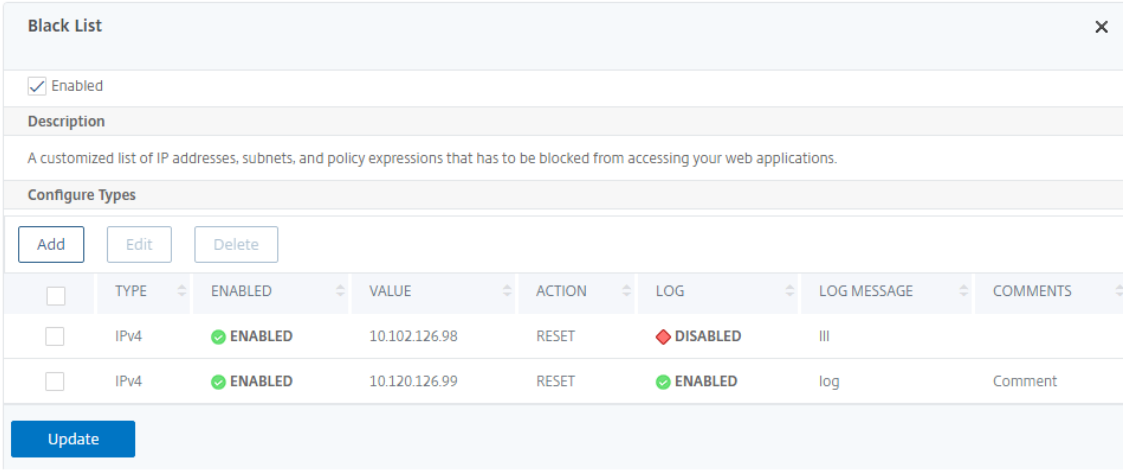
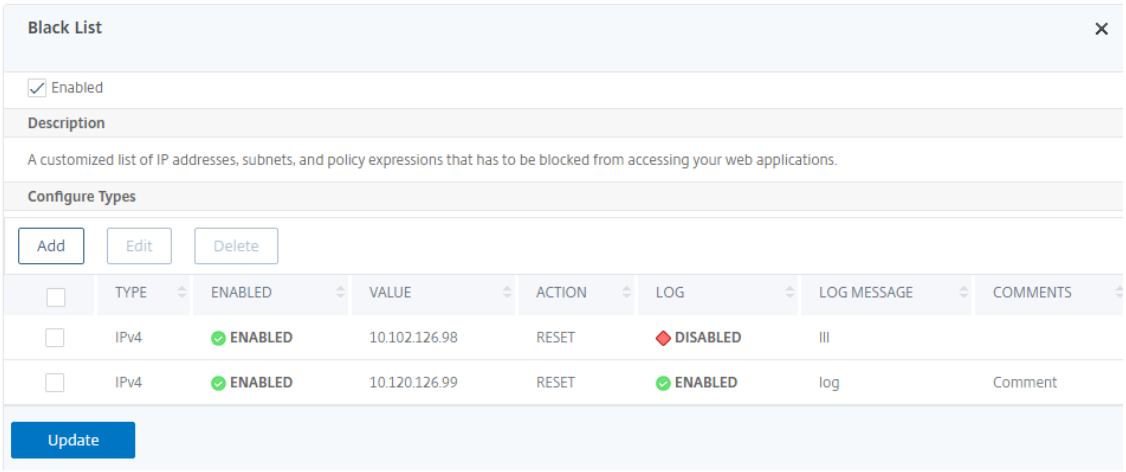
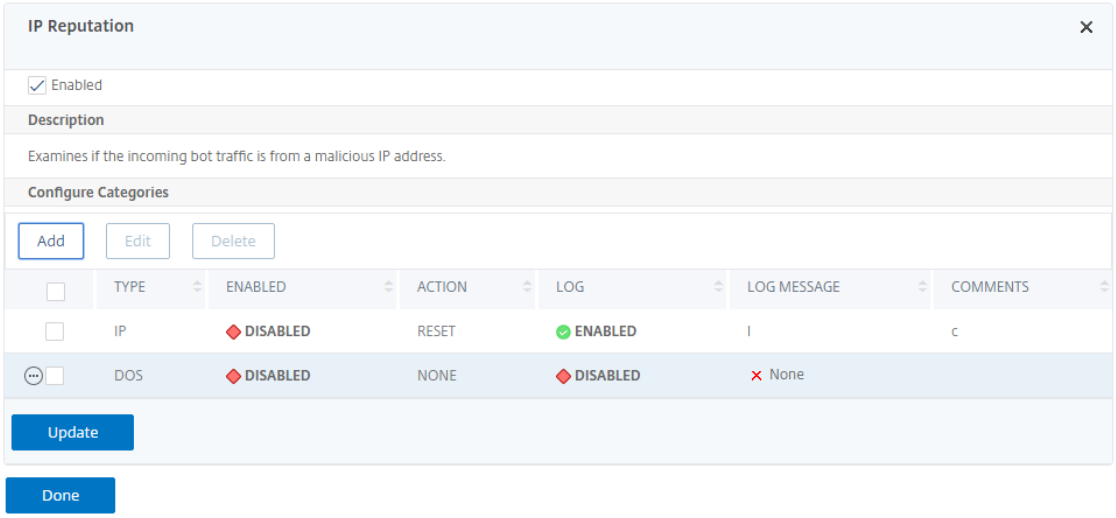
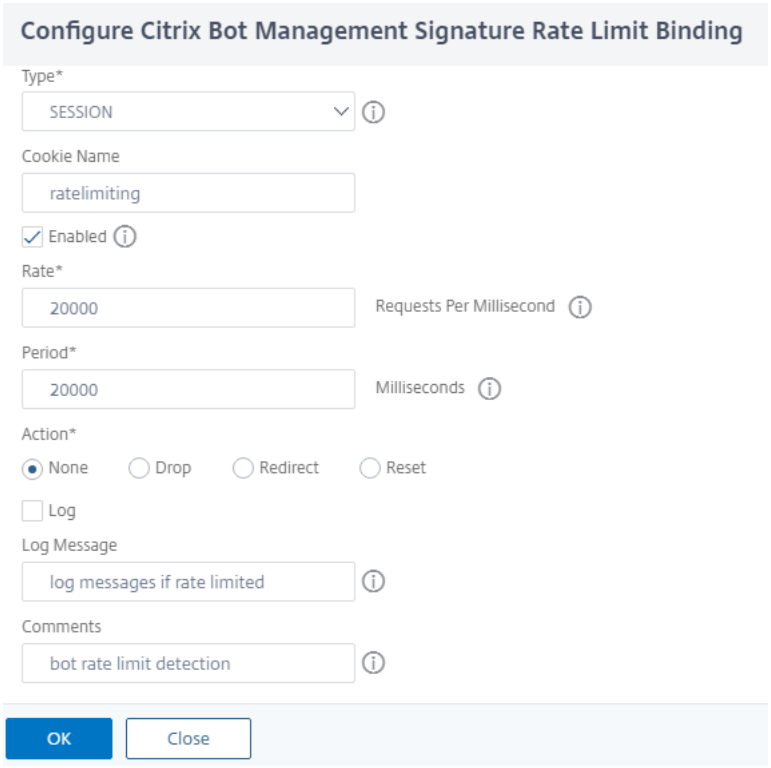
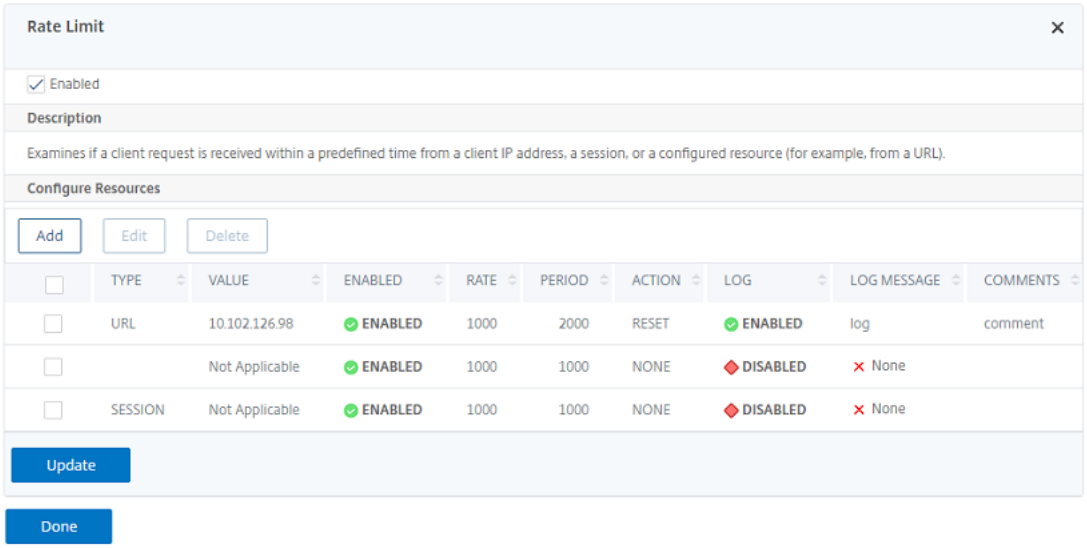
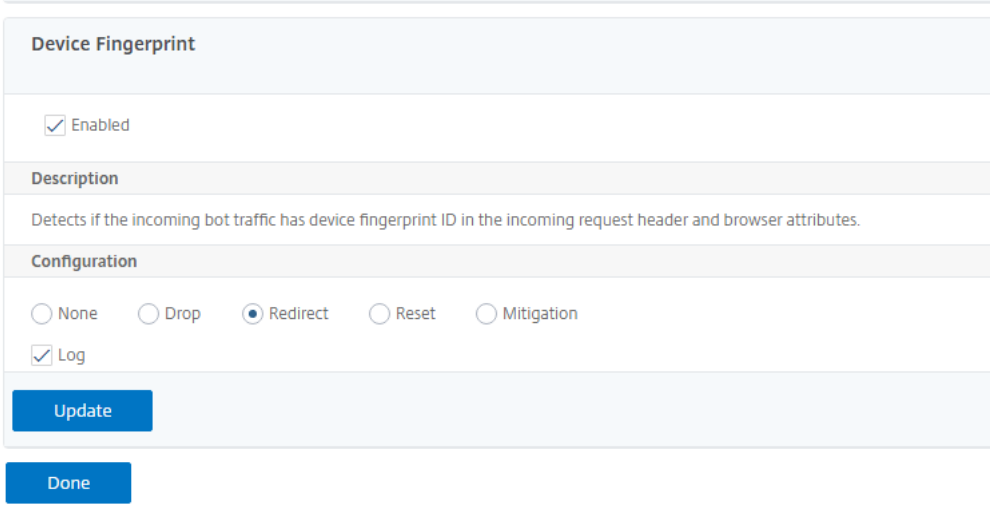
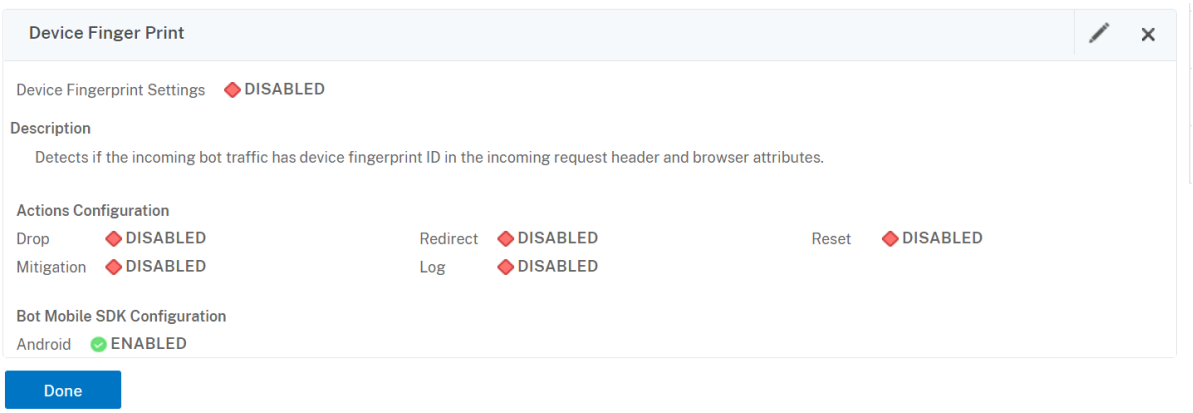
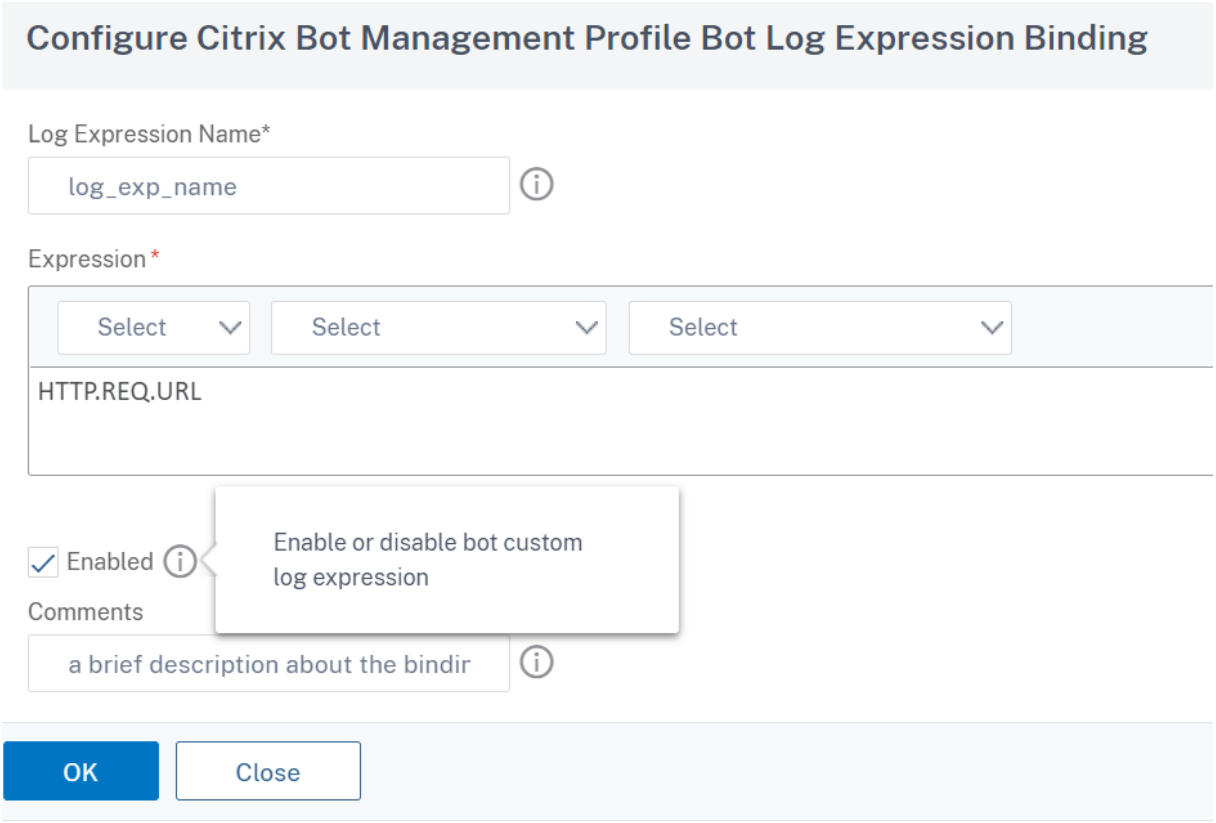
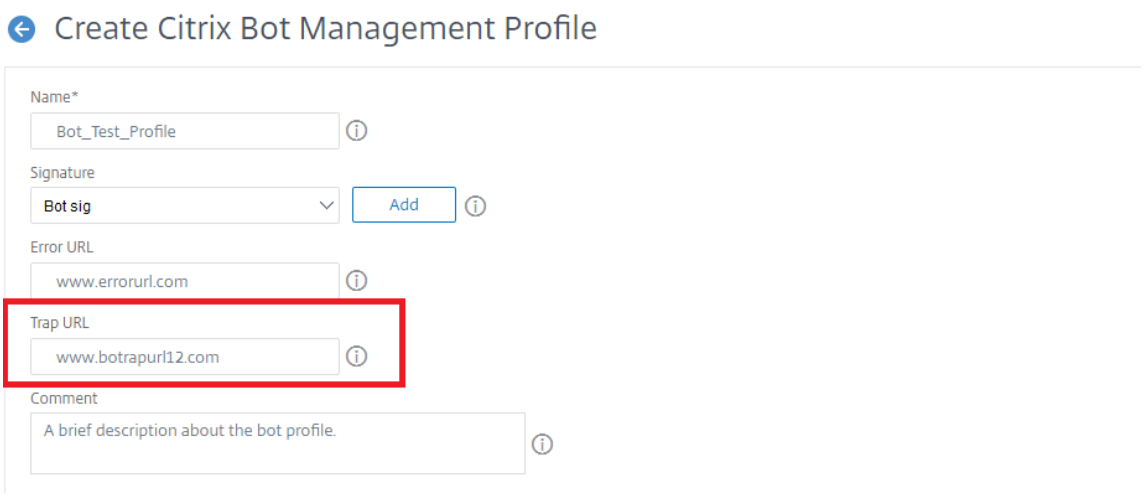
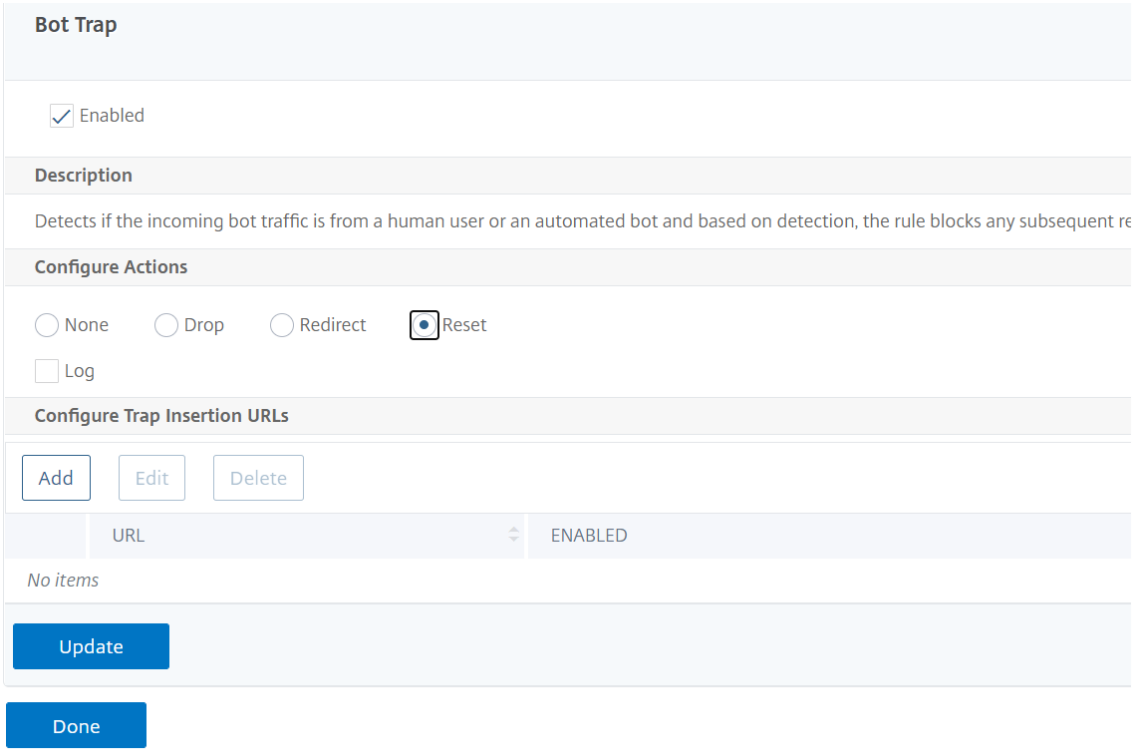
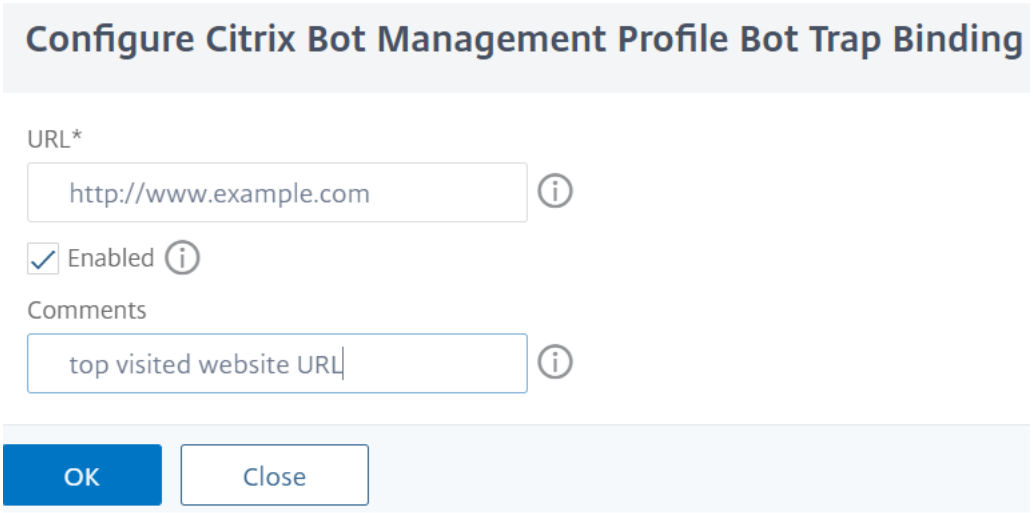
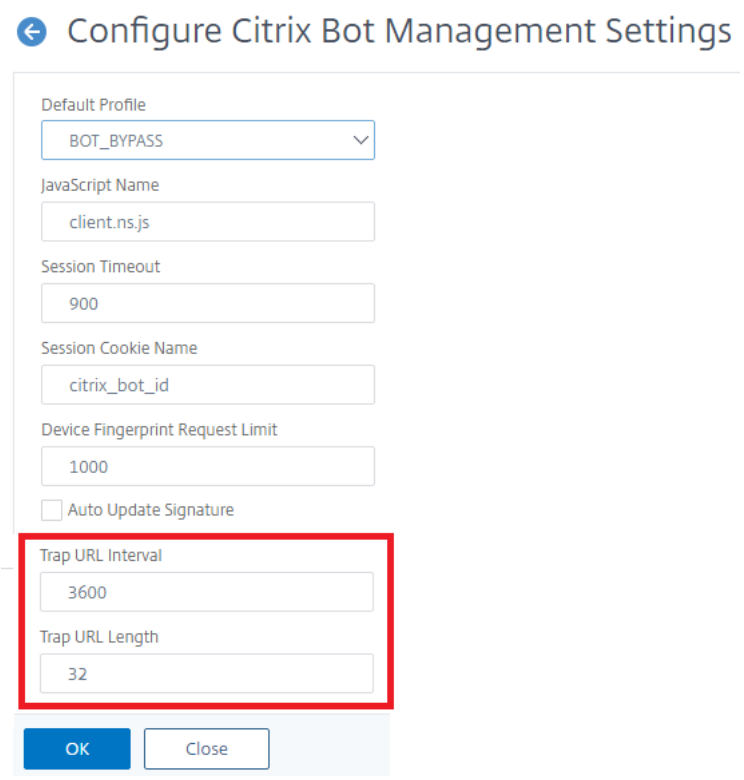
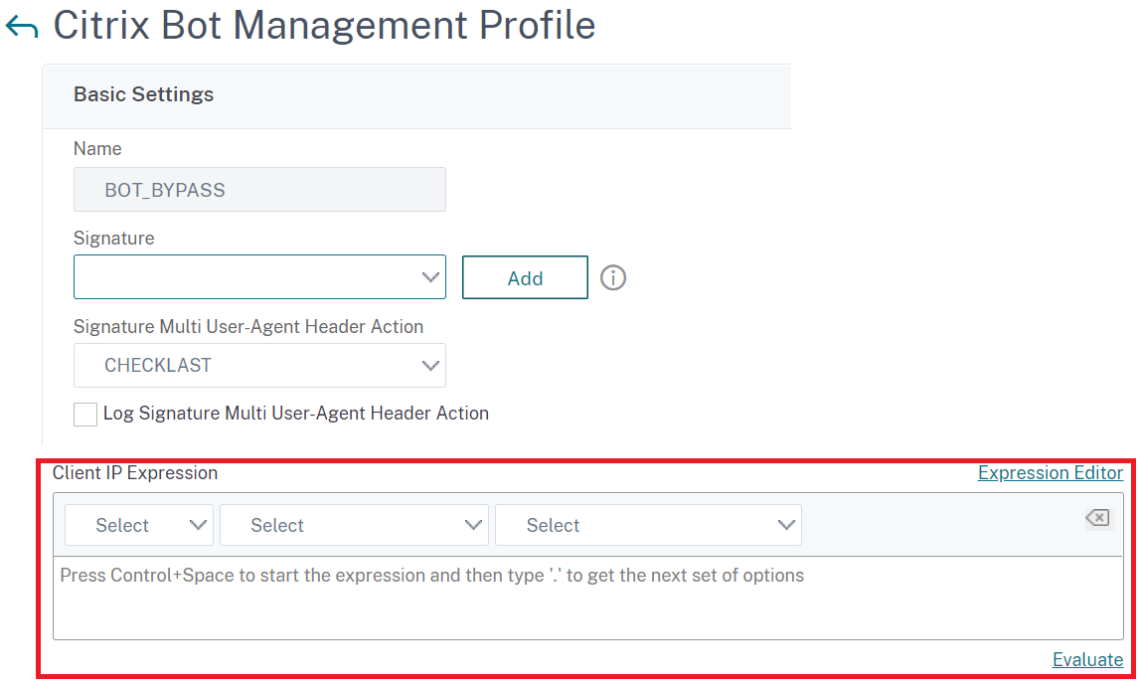
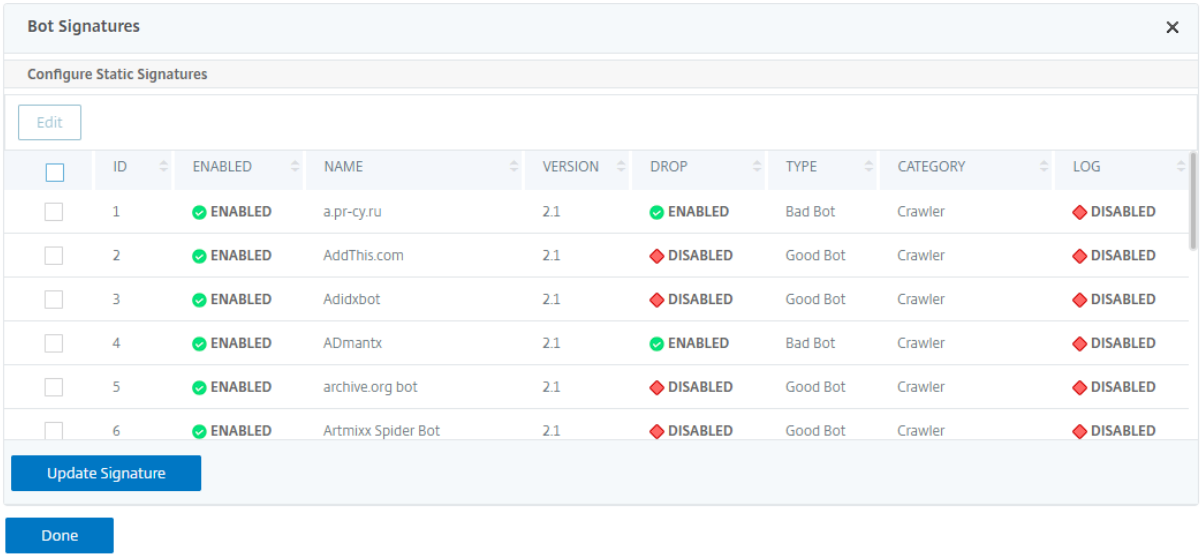
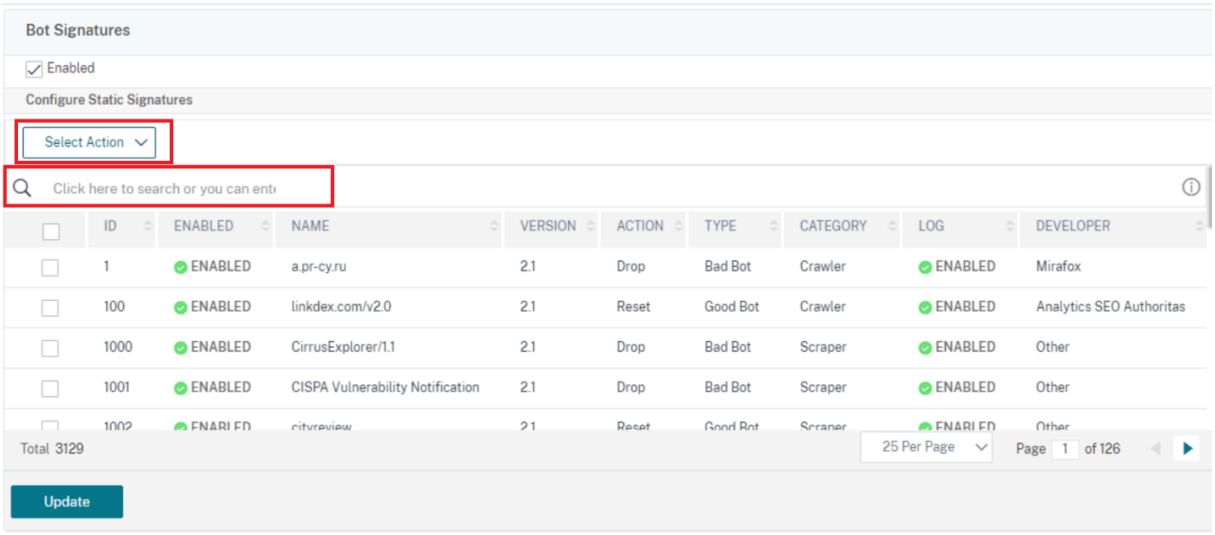
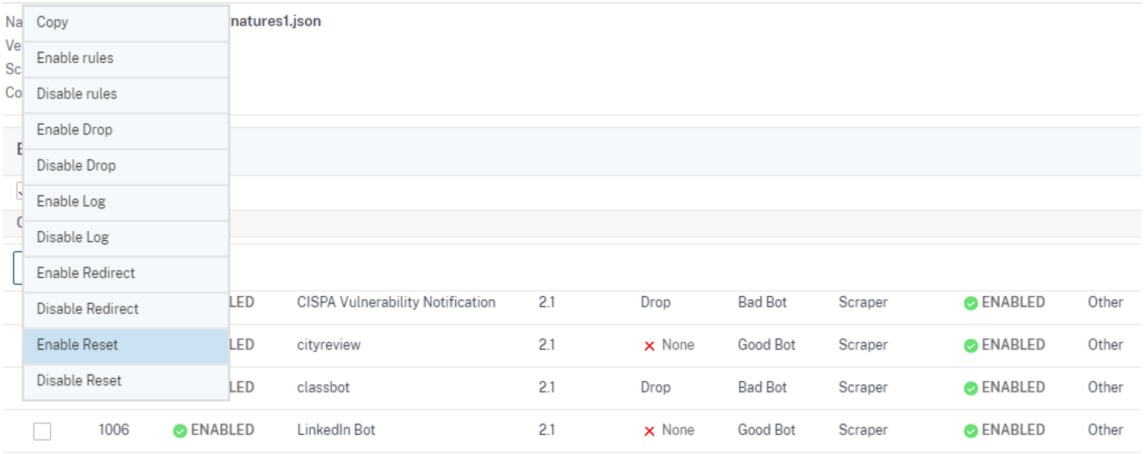
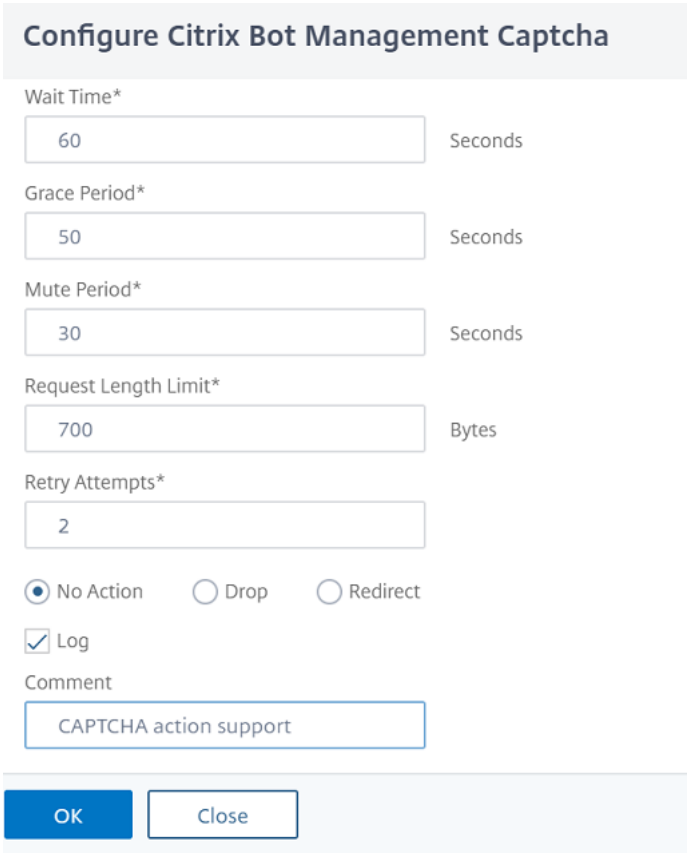
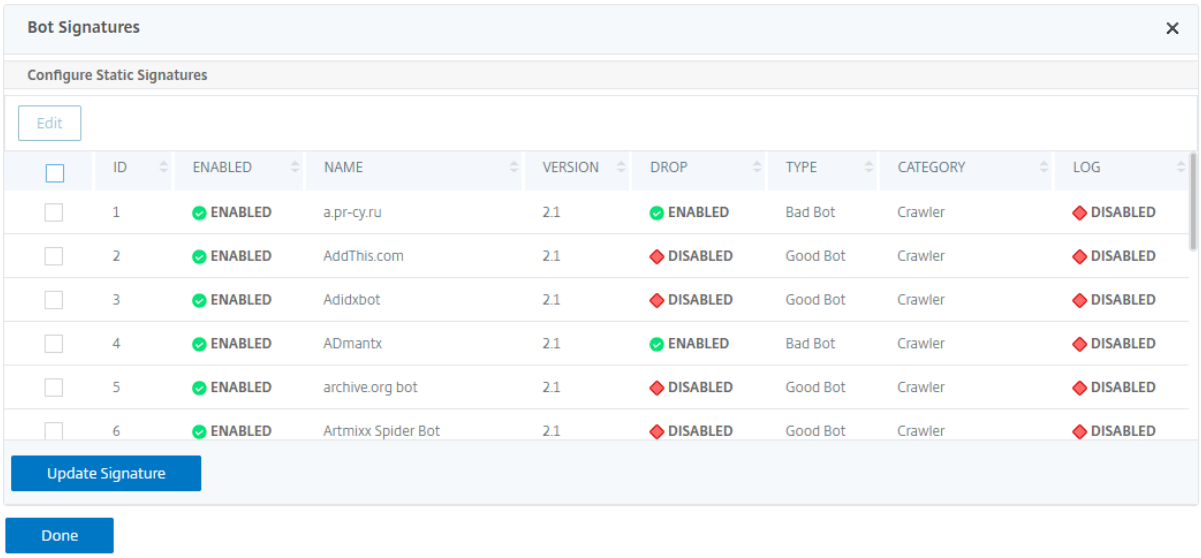
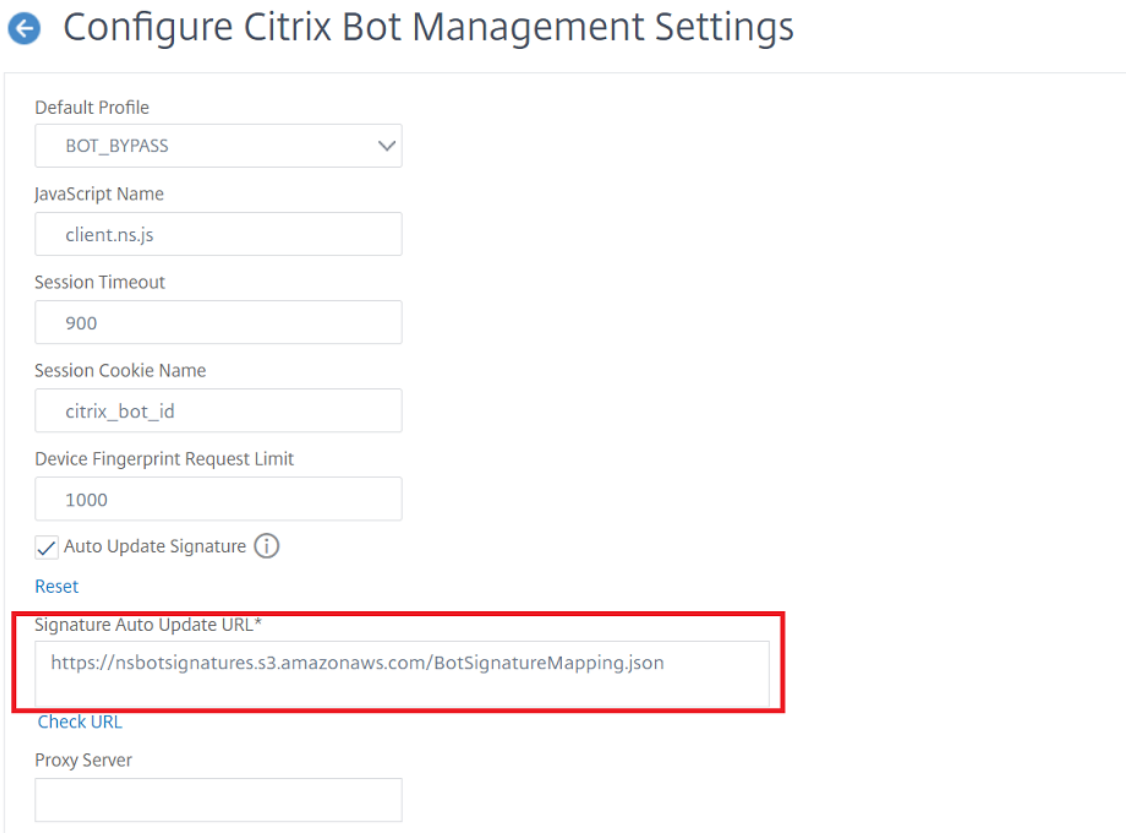
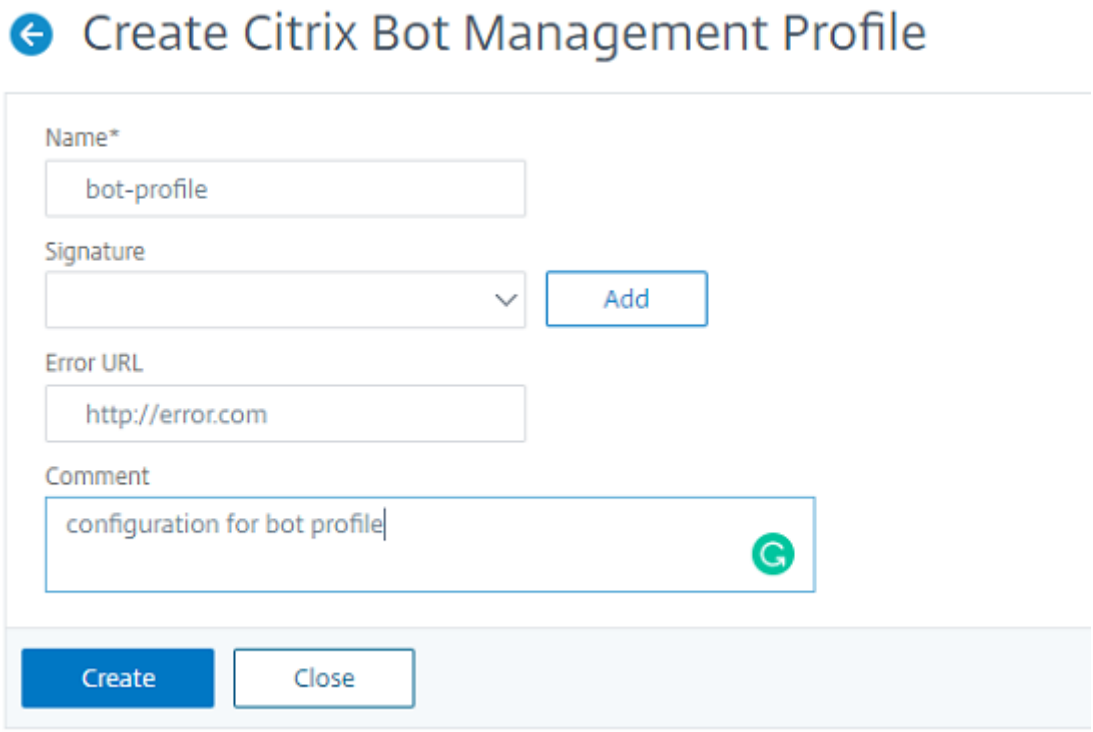
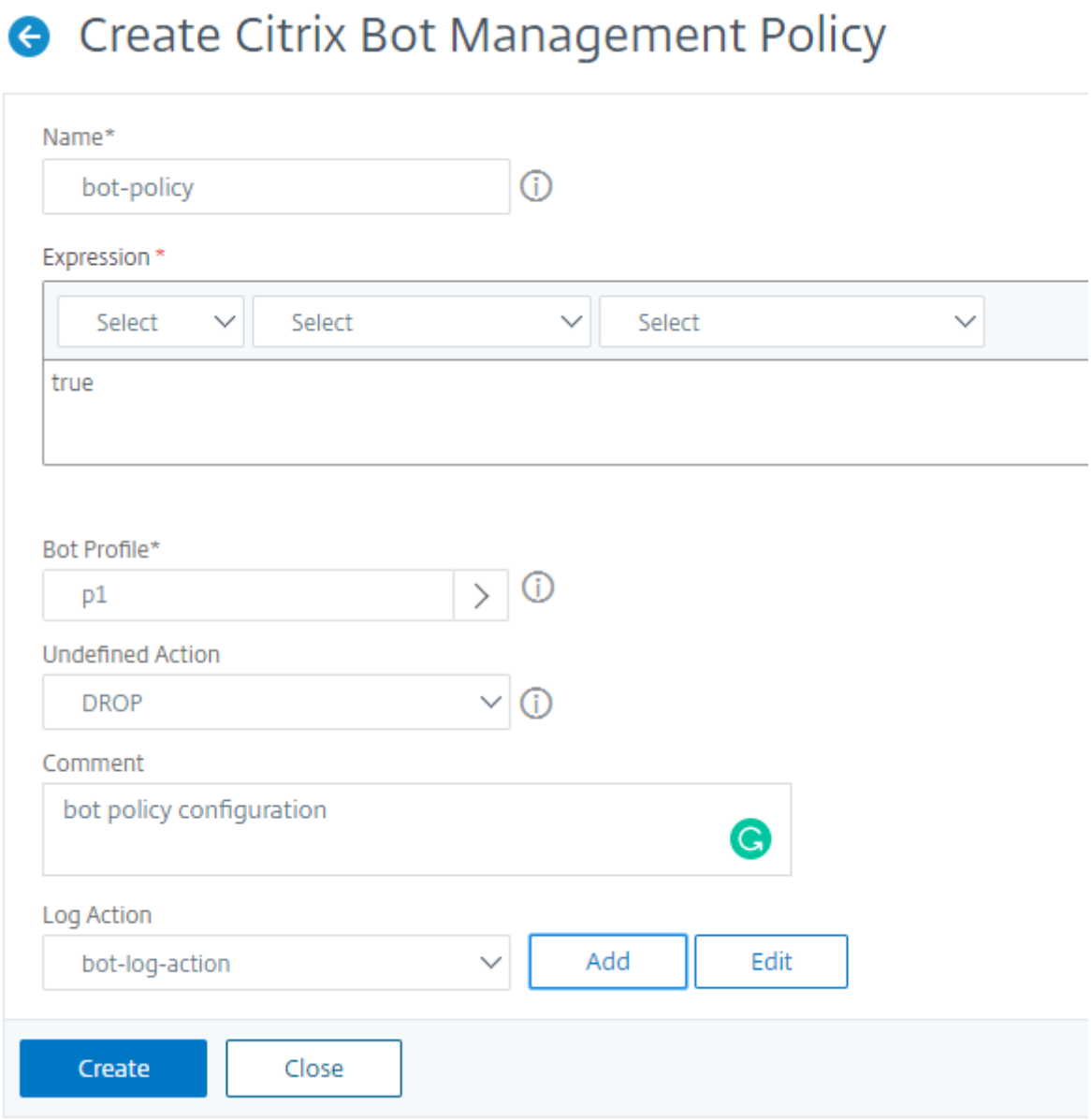
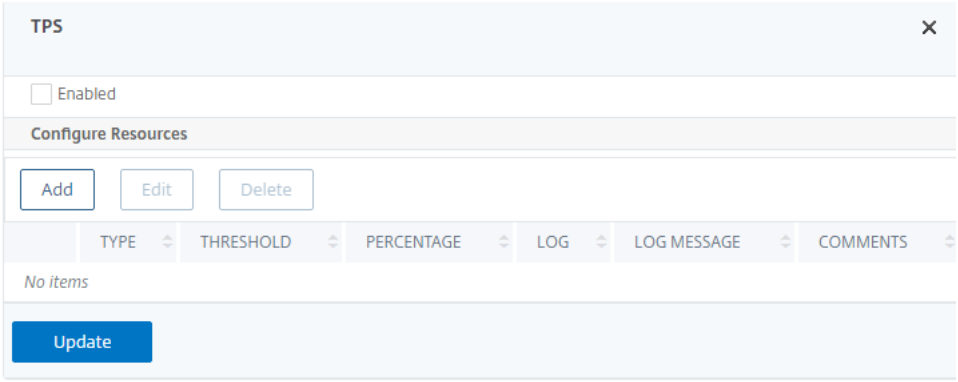
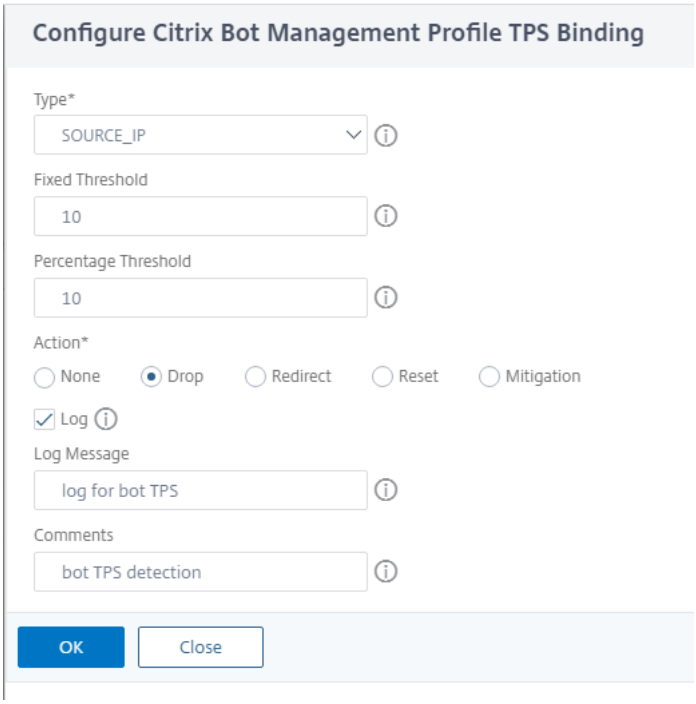
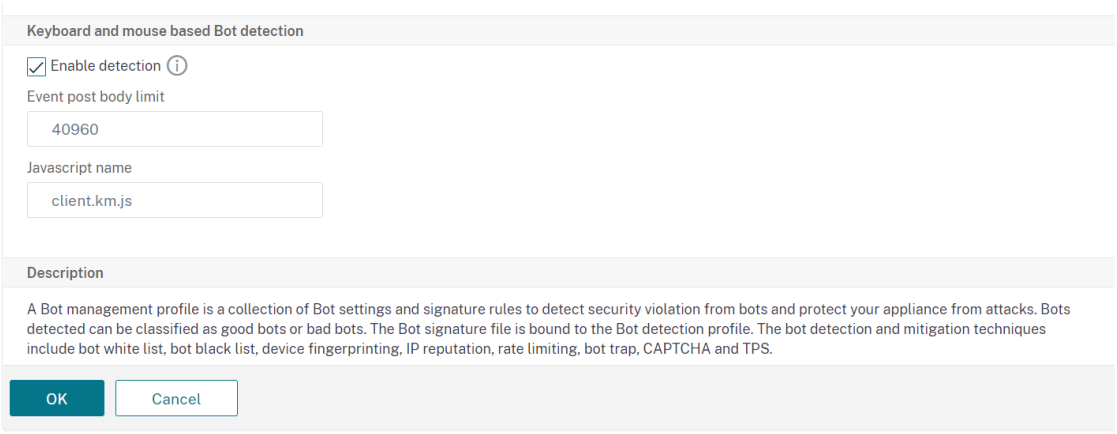
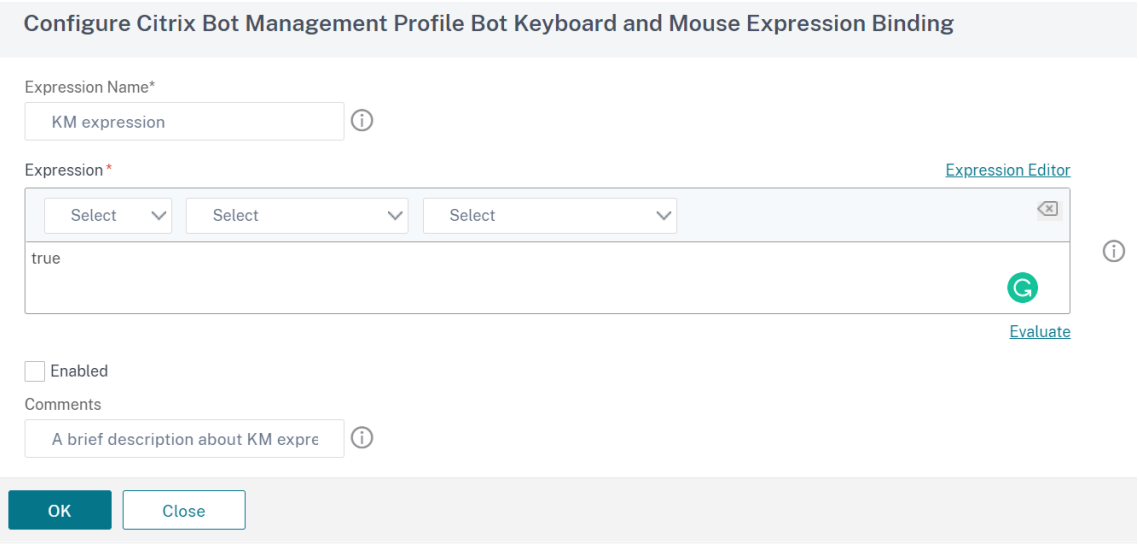
7 %3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c Cross-site script
     check failed for
8 field text_area="Bad tag: script" <blocked>
9 <!--NeedCopy-->
```

Common Event Format (CEF) Logs

The Web App Firewall also supports CEF logs. CEF is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications. CEF enables customers to use a common event log format so that data can easily be collected and aggregated for analysis by an enterprise management system. The log message is broken into different fields so that you can easily parse the message and write scripts to identify important information.

Analyzing the CEF Log Message

In addition to date, timestamp, client IP, log format, appliance, company, build version, module, and security check information, Web App Firewall CEF Log messages include the following details:

- src – source IP address
- spt – source port number
- request – request URL
- act – action (e.g. blocked, transformed)
- msg – message (Message regarding the observed security check violation)
- cn1 – event ID
- cn2 – HTTP Transaction ID
- cs1 – profile name
- cs2 – PPE ID (e.g. PPE1)
- cs3 - Session ID
- cs4 – Severity (e.g. INFO, ALERT)
- cs5 – event year
- cs6 - Signature Violation Category
- method – Method (e.g. GET/POST)

For example, consider the following CEF format log message, which was generated when a Start URL violation was triggered:

```
1 Jun 12 23:37:17 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
     .0
2 |APPFW|APPFW_STARTURL|6|src=10.217.253.62 spt=47606 method=GET
```



```
request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal
    URL. cn1=1340
cn2=653 cs1=pr_ffc cs2=PPE1 cs3=EsdGd3VD0OaaURLcZnj05Y6DOmE0002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

The above message can be broken down into different components. Refer to the CEP log components table.

Example of a request check violation in CEF log format: request is not blocked

```
Jun 13 00:21:28 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_FIELDCONSISTENCY|6|src=10.217.253.62 spt=761 method=GET request=
http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
123456789234&drinking_pref=on&text_area=&loginButton=ClickToLogin&
    as_sfid
=
    AAAAAAWIahZuYoIFbjBhYMP05mJLTwEfIY0a7AKGMg3jIBaKmwtK4t7M7lNxOgj7Gmd3SZc8KUj6CP

7W5kIWDRHN8PtK1Zc-txHkHNx1WknuG9DzTuM7t1THhluevXu9I4kp8%3D&as_fid=
    feeec8758b4174
0eedeeb6b35b85dfd3d5def30c msg=Field consistency check failed for field
     passwd cn1=1401
cn2=707 cs1=pr_ffc cs2=PPE1 cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=
    ALERT cs5=2015 act=
not blocked
<!--NeedCopy-->
```

Example of a response check violation in CEF format: response is transformed

```
Jun 13 00:25:31 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SAFECOMMERCE|6|src=10.217.253.62 spt=34041 method=GET request=
http://aaron.stratum8.net/FFC/CreditCardMind.html msg=Maximum number of
     potential credit
card numbers seen cn1=1470 cn2=708 cs1=pr_ffc cs2=PPE1
cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

Example of a request side signature violation in CEF format: request is blocked

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request=
```



```
3 http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature
    violation rule ID 807:
4 web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=
    PPE0
5 cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    blocked
6 <!--NeedCopy-->
```

Logging geolocation in the Web App Firewall violation messages

Geolocation, which identifies the geographic location from which requests originate, can help you configure the Web App Firewall for the optimal level of security. To bypass security implementations such as rate limiting, which rely on the IP addresses of the clients, malware or rogue computers can keep changing the source IP address in requests. Identifying the specific region from where requests are coming can help determine whether the requests are from a valid user or a device attempting to launch cyberattacks. For example, if an excessively large number of requests are received from a specific area, it is easy to determine whether they are being sent by users or a rogue machine. Geolocation analysis of the received traffic can be very useful in deflecting attacks such as denial of service (DoS) attacks.

The Web App Firewall offers you the convenience of using the built-in Citrix ADC database for identifying the locations corresponding to the IP addresses from which malicious requests are originating. You can then enforce a higher level of security for requests from those locations. Citrix default syntax (PI) expressions give you the flexibility to configure location based policies that can be used in conjunction with the built-in location database to customize firewall protection, bolstering your defense against coordinated attacks launched from rogue clients in a specific region.

You can use the Citrix ADC built-in database, or you can use any other database. If the database does not have any location information for the particular client IP address, the CEF log shows geolocation as an Unknown geolocation.

Note: Geolocation logging uses the Common Event Format (CEF). By default, CEF logging and GeoLocationLogging are OFF. You must explicitly enable both parameters.

Example of a CEF log message showing geolocation information

```
1 June 8 00:21:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
2 APPFW_STARTURL|6|src=10.217.253.62 geolocation=NorthAmerica.US.Arizona.
    Tucson.*.*
3 spt=18655 method=GET request=http://aaron.stratum8.net/FFC/login.html
4 msg=Disallow Illegal URL. cn1=77 cn2=1547 cs1=test_pr_adv cs2=PPE1
5 cs3=KDynjg1pbFtfhC/nt0rBU1o/Tyg0001 cs4=ALERT cs5=2015 act=not blocked
6 <!--NeedCopy-->
```


Example of a log message showing geolocation= Unknown

```
June 9 23:50:53 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|
APPFW|APPFW_STARTURL|6|src=10.217.30.251 geolocation=Unknown spt=5086
method=GET request=http://aaron.stratum8.net/FFC/login.html msg=
    Disallow Illegal URL.
cn1=74 cn2=1576 cs1=test_pr_adv cs2=PPE2 cs3=
    PyR0eOEM4gf6GJiTyauiHByL88E0002
cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

Using the command line to configure the log action and other log parameters

To configure the log action for a security checks of a profile by using the command line

At the command prompt, type one of the following commands:

- `set appfw profile <name> SecurityCheckAction ([log] | [none])`
- `unset appfw profile <name> SecurityCheckAction`

Examples

```
set appfw profile pr_ffc StartURLAction log
```

```
unset appfw profile pr_ffc StartURLAction
```

To configure CEF logging by using the command line

The CEF logging is disabled by default. At the command prompt, type one of the following commands to change or display the current setting:

- `set appfw settings CEFLogging on`
- `unset appfw settings CEFLogging`
- `sh appfw settings | grep CEFLogging`

To configure the logging of the credit card numbers by using the command line

At the command prompt, type one of the following commands:

- `set appfw profile <name> -doSecureCreditCardLogging ([ON] | [OFF])`
- `unset appfw profile <name> -doSecureCreditCardLogging`

To configure Geolocation logging by using the command line

1. Use the set command to enable GeoLocationLogging. You can enable the CEF logging at the same time. Use the unset command to disable geolocation logging. The show command shows the current settings of all the Web App Firewall parameters, unless you include the grep command to show the setting for a specific parameter.

- `set appfw settings GeoLocationLogging ON [CEFLogging ON]`
- `unset appfw settings GeoLocationLogging`
- `sh appfw settings | grep GeoLocationLogging`

2. Specify the database

```
add locationfile /var/netscaler/inbuilt_db/Citrix_netscaler_InBuilt_GeoIP_DB
.csv
```

or

```
add locationfile <path to database file>
```

Customizing Web App Firewall Logs

Default format (PI) expressions give you the flexibility to customize the information included in the logs. You have the option to include the specific data that you want to capture in the Web App Firewall generated log messages. For example, if you are using AAA-TM authentication along with the Web App Firewall security checks, and would like to know the accessed URL that triggered the security check violation, the name of the user who requested the URL, the source IP address, and the source port from which the user sent the request, you can use the following commands to specify customized log messages that include all the data:

```
> sh version
NetScaler NS12.1: Build 50.0013.nc, Date: Aug 28 2018, 10:51:08   (64-
   bit)
 Done
<!--NeedCopy-->
```

```
> add audit messageaction custom1 ALERT 'HTTP.REQ.URL + " " + HTTP.REQ.
   USER.NAME + " " + CLIENT.IP.SRC + ":" + CLIENT.TCP.SRCPORT'
Warning: HTTP.REQ.USER has been deprecated. Use AAA.USER instead.
 Done
<!--NeedCopy-->
```

```
> add appfw profile test_profile
 Done
<!--NeedCopy-->
```

```
> add appfw policy appfw_pol true test_profile -logAction custom1
 Done
<!--NeedCopy-->
```


Configuring Syslog policy to segregate Web App Firewall logs

The Web App Firewall offers you an option to isolate and redirect the Web App Firewall security log messages to a different log file. This might be desirable if the Web App Firewall is generating a large number of logs, making it difficult to view other Citrix ADC log messages. You can also use this option when you are interested only in viewing the Web App Firewall log messages and do not want to see the other log messages.

To redirect the Web App Firewall logs to a different log file, configure a syslog action to send the Web App Firewall logs to a different log facility. You can use this action when configuring the syslog policy, and bind it globally for use by Web App Firewall.

Example:

1. Switch to the shell and use an editor such as vi to edit the /etc/syslog.conf file. Add a new entry to use local2.* to send logs to a separate file as shown in the following example:

   ```
   local2.\* /var/log/ns.log.appfw
   ```

2. Restart the syslog process. You can use the grep command to identify the syslog process ID (PID), as shown in the following example:

   ```
   root@ns\## **ps -A | grep syslog**

   1063 ?? Ss 0:03.00 /usr/sbin/syslogd -b 127.0.0.1 -n -v -v -8 -C

   root@ns## **kill -HUP** 1063
   ```

3. From the command line interface, configure the syslog action and policy. Bind it as a global Web App Firewall policy.

```
> add audit syslogAction sysact 1.1.1.1 -logLevel ALL -logFacility LOCAL2

> add audit syslogPolicy syspol1 ns_true sysact1

> bind appfw global syspol1 100
```

1. All Web App Firewall security check violations will now be redirected to the /var/log/ns.log.appfw file. You can tail this file to view the Web App Firewall violations that are getting triggered during the processing of the ongoing traffic.

   ```
   root@ns## tail -f ns.log.appfw
   ```

Warning: If you have configured the syslog policy to redirect the logs to a different log facility, the Web App Firewall log messages no longer appear in the /var/log/ns.log file.

Viewing the Web App Firewall Logs

You can view the logs by using the syslog viewer, or by logging onto the Citrix ADC appliance, opening a UNIX shell, and using the UNIX text editor of your choice.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Web App Firewall security check violations:

- `Shell`
- `tail -f /var/log/ns.log`

You can use the vi editor, or any Unix text editor or text search tool, to view and filter the logs for specific entries. For example, you can use grep command to access the log messages pertaining to the Credit Card violations:

- `tail -f /var/log/ns.log | grep SAFECOMMERCE`

To access the log messages by using the GUI

The Citrix GUI includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- To view log messages for a specific security check of a profile, navigate to **Web App Firewall** > **Profiles**, select the target profile, and click Security Checks. Highlight the row for the target security check and click Logs. When you access the logs directly from the selected security check of the profile, it filters out the log messages and displays only the logs pertaining to the violations for the selected security check. Syslog viewer can display Web App Firewall logs in the Native format as well as the CEF format. However, in order for the syslog viewer to filter out the target profile specific log messages, the logs must be in the CEF log format when accessed from the profile.
- You can also access the Syslog Viewer by navigating to **Citrix ADC** > **System** > **Auditing**. In the Audit Messages section, click Syslog messages link to display the Syslog Viewer, which displays all log messages, including all Web App Firewall security check violation logs for all profiles. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Web App Firewall** > **policies** > **Auditing**. In the Audit Messages section, click Syslog messages link to display the Syslog Viewer, which displays all log messages, including all security check violation logs for all profiles.

The HTML based Syslog Viewer provides the following filter options for selecting only the log messages that are of interest to you:

- **File**—The current /var/log/ns.log file is selected by default, and the corresponding messages appear in the Syslog Viewer. A list of other log files in the /var/log directory are available in a compressed .gz format. To download and un-compress an archived log file, just select the log file from the dropdown option. The log messages pertaining to the selected file are then displayed in the syslog viewer. To refresh the display, click the Refresh icon (a circle of two arrows).

- **Module list box**—You can select the Citrix ADC module whose logs you want to view. You can set it to APPFW for Web App Firewall logs.
- **Event Type list box**—This box contains a set of check boxes for selecting the type of event you are interested in. For example, to view the log messages pertaining to the signature violations, you can select the **APPFW_SIGNATURE_MATCH** check box. Similarly, you can select a check box to enable the specific security check that is of interest to you. You can select multiple options.
- **Severity**—You can select a specific severity level to show just the logs for that severity level. Leave all the check boxes blank if you want to see all logs.

  To access the Web App Firewall security check violation log messages for a specific security check, filter by selecting **APPFW** in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the **APPFW_FIELDFORMAT** check box and click the Apply button, only log messages pertaining to the Field Formats security check violations appear in the Syslog Viewer. Similarly, if you select the **APPFW_SQL** and **APPFW_STARTURL** check boxes and click the **Apply** button, only log messages pertaining to these two security check violations will appear in the syslog viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **EventType**, **EventID**, **ClientIP**, **TransactionID**, and so on appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Click to Deploy: This functionality is available only in the GUI. You can use the Syslog Viewer to not only view the logs but also to deploy relaxation rules based on the log messages for the Web App Firewall security check violations. The log messages must be in CEF log format for this operation. If the relaxation rule can be deployed for a log message, a check box appears at the right edge of the Syslog Viewer box in the row. Select the check box, and then select an option from the Action list to deploy the relaxation rule.
Edit & Deploy,
Deploy, and
Deploy All are available as Action options. For example, you can select an individual log message to edit and deploy. You can also select the check boxes for multiple log messages from one or more security checks and use the Deploy or Deploy All option. Click to Deploy functionality is currently supported for the following security checks:

- StartURL
- URL Buffer overflow
- SQL Injection
- cross-site scripting
- Field consistency
- Cookie consistency

To use Click to Deploy functionality in the GUI

1. In the **Syslog Viewer**, select **APPFW** in the **Module** options.
2. Select the security check for which to filter corresponding log messages.
3. Enable the check box to select the rule.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

> **Note:**
>
> SQL Injection and cross-site scripting rules that are deployed by using Click **Deploy** option do not include the fine grain relaxation recommendations.

Highlights

- **CEF Log Format support**—The CEF log format option provides a convenient option to monitor, parse, and analyze the Web App Firewall log messages to identify attacks, fine tune configured settings to decrease false positives, and gather statistics.
- **Click to Deploy**—The Syslog viewer provides an option to filter, evaluate, and deploy relaxation rules for single or multiple security check violations from one convenient location.
- **Option to customize log message**—You can use advanced PI expressions to customize log messages and include the data you want to see in the logs.
- **Segregate Web App Firewall specific logs**—You have an option to filter and redirect application-firewall specific logs to a separate log file.
- **Remote Logging**—You can redirect the log messages to a remote syslog server.
- **Geolocation Logging**—You can configure the Web App Firewall to include the geolocation of the area from where the request is received. A built-in geolocation database is available, but you have the option to use an external geolocation database. The Citrix ADC appliance supports both IPv4 and IPv6 static geolocation databases.
- **Information rich log message**—Following are some examples of the type of information that can be included in the logs, depending on the configuration:
  - An Web App Firewall policy was triggered.
  - A security check violation was triggered.
  - A request was considered to be malformed.
  - A request or the response was blocked or not blocked.
  - Request data (such as SQL or cross-site scripting special characters) or response data (such as Credit card numbers or safe object strings) was transformed.
  - The number of credit cards in the response exceeded the configured limit.
  - The credit card number and type.
  - The log strings configured in the signature rules, and the signature ID.
  - Geolocation information about the source of the request.
  - Masked (X'd out) user input for protected confidential fields.

Mask sensitive data using regex pattern

The REGEX_REPLACE advanced policy (PI) function in a log expression (bound to a Web Application Firewall (WAF) profile) enables you to mask sensitive data in WAF logs. You can use the option to mask data using a regex pattern and provide a character or a string pattern to mask the data. Also, you can configure the PI function to replace the first occurrence or all occurrences of the regex pattern.

By default the Citrix GUI interface provides the following mask:

- SSN
- Credit Card
- Password
- Username

Mask sensitive data in Web Application Firewall logs

You can mask sensitive data in WAF logs by configuring the REGEX_REPLACE advanced policy expression in the log expression bound to a WAF profile.
To mask sensitive data, you must complete the following steps:

1. Add a Web Application Firewall profile
2. Bind a log expression to the WAF profile

Add a Web Application Firewall profile

At the command prompt, type:

```
add appfw profile <name>
```

Example:

```
Add appfw profile testprofile1
```

Bind a log expression with the Web Application Firewall profile

At the command prompt, type:

```
bind appfw profile <name> -logExpression <string> <expression> -comment <
string>
```

Example:

```
bind appfw profile testProfile -logExpression "MaskSSN""HTTP.REQ.BODY
(10000).REGEX_REPLACE(re!\b\d{ 3 } -\d{ 2 } -\d{ 4 } \b!, “xxx” , ALL)"-
comment "SSN Masked"
```


Mask sensitive data in Web Application Firewall logs by using Citrix ADC GUI

1. On the navigation pane, expand **Security** > **Citrix Web App Firewall** > **Profiles**.
2. On the **Profiles** page, click **Edit**.
3. On the **Citrix Web App Firewall Profile** page, navigate to **Advanced Settings** section and click **Extended Logging**.



4. In the **Extended Logging** section, click **Add**.



5. On the **Create Citrix Web App Firewall Extended Log Binding** page, set the following parameters:
   a) Name. Name of the log expression.
   b) Enabled. Select this option to mask sensitive data.
   c) Log mask. Select the data to be masked.
   d) Expression. Enter the advanced policy expression that enables you to mask sensitive data in WAF logs
   e) Comments. Brief description about the masking sensitive data.
6. Click **Create** and **Close**.



Appendices

September 14, 2021

The following supplemental material provides additional detail about complex or peripheral Web App Firewall tasks.

PCRE character encoding format

September 14, 2021

The **Citrix ADC operating system supports direct** entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your Web App Firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

Many character types require encoding using a PCRE regular expression if you include them in your Web App Firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not

included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.

- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.

  ASCII control characters. Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters must never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The Citrix ADC appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, "English US," the Web App Firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The Web App Firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The Web App Firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The Web App Firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The Web App Firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The Web App Firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The Web App Firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- **Unicode (UTF-8).** The Web App Firewall supports certain more characters in the UTF-8 character set, including those used in modern Russian.

When configuring the Web App Firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which is assigned single, two-digit codes in that

character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the Web App Firewall configuration is "\xNN" for ASCII characters; "\xNN\xNN" for non-ASCII characters used in English, Russian, and Turkish; and "\xNN\xNN\xNN" for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in a Web App Firewall regular expression as a UTF-8 character, you would type \x21. If you want to include an á, you would type \xC3\xA1.

> **Note:**
>
> Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character's UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as \x20 to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the Web App Firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters.

  Actual URL: `http://www.josénuñez.com`
  Encoded URL: `^http://www\[.\]jos\\xC3\\xA9nu\\xC3\\xB1ez\[.\]com$`

- Another URL containing extended ASCII characters.

  Actual URL: `http://www.example.de/trömso.html`
  Encoded URL: `^http://www[.]example[.\]de/tr\xC3\xB6mso[.]html$`

  A form field name containing extended ASCII characters.

  Actual Name: nome_do_usuário
  Encoded Name: ^nome_do_usu\xC3\xA1rio$

- A safe object expression containing extended ASCII characters.

  Unencoded Expression [A-Z]{3,6}¥[1-9][0-9]{6,6}
  Encoded Expression: [A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}

You can find several tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful website that contains this information is available in the following table.

`http://www.utf8-chartable.de/unicode-utf8-table.pl`

For the characters in the table on this website to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of webpages are correct.

Whitehat WASC signature types for WAF use

September 14, 2021

The Citrix Web App Firewall accepts and generates blocking rules for all vulnerability types that the Whitehat scanners generate. However, certain vulnerabilities are most applicable to a web App Firewall. Following are lists of those vulnerabilities, categorized by whether they are addressed by WASC 1.0, WASC 2.0, or best practices signature types.

WASC 1.0 signature types

- HTTP Request Smuggling
- HTTP Response Splitting
- HTTP Response Smuggling
- Null Byte Injection
- Remote File Inclusion
- URL Redirector Abuse

WASC 2.0 signature types

- Abuse of Functionality
- Brute Force
- Content Spoofing
- Denial of Service
- Directory Indexing
- Information Leakage
- Insufficient Anti-automation
- Insufficient Authentication
- Insufficient Authorizatio
- Insufficient Session Expiration
- LDAP Injection
- Session Fixation

Best Practices

- Autocomplete Attribute
- Insufficient Cookie Access Control
- Insufficient Password Strength
- Invalid HTTP Method Usage
- Non-HttpOnly Session Cookie
- Persistent Session Cookie
- Personally Identifiable Information
- Secured Cachable HTTP Messages
- Unsecured Session Cookie

Streaming support for request processing

September 14, 2021

Citrix Web App Firewall supports request side streaming to provide a significant performance boost. Instead of buffering a request, the appliance examines the incoming traffic for security violation such as SQL, cross-site scripting, field consistency, field formats. When the appliance completes processing data for a field, the request is forwarded to the back-end server while the appliance continues to evaluate other fields. This data processing significantly improves the processing time in handling forms have many fields.

> **Note:**
>
> Citrix Web App Firewall supports a maximum post size of 20 MB without streaming. For better resource utilization, Citrix recommends you to enable streaming only for payloads greater than 20 MB. Also, the back-end server must accept the chunked requests if streaming is enabled.

Although the streaming process is transparent to the users, minor configuration adjustments are required due to the following changes:

RegEx Pattern Match: RegEx pattern match is now restricted to 4K for contiguous character string match.

Field Name Match: The Web App Firewall learning engine can only distinguish the first 128 bytes of the name. If a form has multiple fields with names that have an identical string match for the first 128 bytes, the learning engine does not distinguish them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Removal of white spaces, percent decoding, Unicode decoding, and charset conversion are done during canonicalization to provide security check inspection. The 128 byte limit is applicable for the canonicalized representation of the field name in UTF-8 character format. The ASCII characters are 1

byte in length but the UTF-8 representation of the characters in some international languages might range from 1 byte to 4 bytes. If each character in a name takes 4 bytes for converting to UTF-8 format, only first 32 characters in the name might be distinguished by the learned rule.

Field Consistency Check: When you enable Field Consistency, all the forms in the session are stored based on the "as_fid" tag inserted by the Web App Firewall without considering the "action_url."

- **Mandatory Form tagging for Form Field consistency:** When the field consistency check is enabled, the form tag must be enabled also. The Field Consistency protection might not work if form tagging is turned off.
- **Sessionless Form Field Consistency:** The Web App Firewall no longer carries out the "GET" to "POST" conversion of forms when the sessionless field consistency parameter is enabled. The form tag is required for sessionless field consistency also.
- **Tampering of as_fid:** If a form is submitted after tampering as_fid, it triggers field consistency violation even if no field was tampered. In non-streaming requests, this was allowed because the forms can be validated using the "action_url" stored in the session.

Signatures: The signatures now have the following specifications:

- **Location:** It is now a mandatory requirement that location must be specified for each pattern. All patterns in the rule **MUST** have a `<Location>` tag.
- **Fast Match:** All signature rules must have a fast match pattern. If there is no fast match pattern, an attempt is made to select one if possible. Fast match is a literal string but PCRE can be used for fast match if they contain a usable literal string.
- **Deprecated Locations:** Following locations are no longer supported in signature rules.
  - HTTP_ANY
  - HTTP_RAW_COOKIE
  - HTTP_RAW_HEADER
  - HTTP_RAW_RESP_HEADER
  - HTTP_RAW_SET_COOKIE

cross-site scripting/SQL Transform: Raw data is used for transformation because the SQL special characters such as single quote(‘), backslash (), and semicolon (;)), and cross-site scripting tags are same and do not require canonicalization of data. Representation of special characters such as HTML entity encoding, percent encoding, or ASCII are evaluated for transform operation.

The Web App Firewall no longer inspects both the attribute name and value for the cross-site scripting transform operation. Now only cross-site scripting attribute names are transformed when streaming is engaged.

Processing cross-site scripting Tags: As part of the streaming changes in NetScaler 10.5.e build and later, the processing of the cross-site scripting tags has changed. In earlier releases, the presence of either open bracket (<), or close bracket (>), or both open and close brackets (<>) was flagged as

cross-site scripting Violation. The behavior has changed in 10.5.e build onwards. Presence of only the open bracket character (<), or only the close bracket character (>) is no longer considered as an attack. This is when an open bracket character (<) is followed by a close bracket character (>), the Cross-site scripting attack gets flagged. Both characters must be present in the right order (< followed by >) to trigger the Cross-site scripting violation.

> **Note:**
>
> **Change in SQL violation log Message:** As part of the streaming changes in 10.5.e build onwards, we now process the input data in blocks. RegEx pattern matching is now restricted to 4K for contiguous character string matching. With this change, the SQL violation log messages might include different information compared to the earlier builds. The keyword and special character in the input are separated by many bytes. The appliance has a track of the SQL keywords and special strings when processing the data, instead of buffering the entire input value. In addition to the field name, the log message includes SQL keyword, SQL special character, or both the SQL keyword and the SQL special character. The rest of the input is no longer included in the log message, as shown in the following example:
>
> **Example:**
>
> In 10.5, when the Web App Firewall detects the SQL violation, the entire input string might be included in the following log message:
>
> SQL Keyword check failed for field **text=”select a name from testbed1\;\(\;\)”.*<blocked>**
>
> In 11.0, we log only the field name, keyword, and special character (if applicable) in the following log message.
>
> SQL Keyword check failed for field `text="select(;)"<blocked>`
>
> This change is applicable to requests that contain **application/x-www-form-urlencoded**, or **multipart/form-data**, or **text/x-gwt-rpc** content-types. Log messages generated during processing of **JSON** or **XML** payloads are not affected in this change.

RAW POST Body: The security check inspections are always done on RAW POST body.

Form ID: The Web App Firewall inserted “as_fid” tag, which is a computed hash of the form is longer unique for the user session. It is an identical value for a specific form irrespective of the user or the session.

Charset: If a request does not have a charset, the default charset specified in the application profile is used when processing the request.

Counters:

Counters with prefix “se” and “appfwreq” are added to track the streaming engine and streaming engine request counters.

```
nsconsmg -d statswt0 -g se_err_
```



```
nsconsmg -d statswt0 -g se_tot_
nsconsmg -d statswt0 -g se_cur_
nsconsmg -d statswt0 -g appfwreq_err_
nsconsmg -d statswt0 -g appfwreq_tot_
nsconsmg -d statswt0 -g appfwreq_cur_
```

`_err counters`: indicate the rare event which must have succeeded but failed due to either memory allocation problem or some other resource crunch.

`_tot counters`: ever increasing counters.

`_cur counters`: counters indicating current values that keep changing based on usage from current transactions.

Tips:

- The Web App Firewall security checks must work the same as before.
- There is no set ordering for the processing of the security checks.
- The response side processing is not affected and remains unchanged.
- Streaming is not engaged if clientless VPN is used.

> **Important:**
>
> **Calculating the Cookie length:** In release 10.5.e in addition to 11.0 release (in builds before 65.x), Web App Firewall way of processing the cookie header was changed. The appliance evaluated the cookie individually, and if the length of a cookie in the cookie header exceeded the configured length, the Buffer Overflow violation was triggered. As a result, requests blocked in NetScaler 10.5 version or earlier releases might be allowed. The length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size might be larger than the accepted value, and the server might respond with “400 Bad Request”.
>
> **Note:**
> The change has been reverted. The behavior in NetScaler version 10.5.e to version 59.13xx.e and its subsequent builds is similar to the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.

Trace HTML requests with security logs

September 14, 2021

> **Note:**
>
> This feature is available in Citrix ADC release 10.5.e.

Troubleshooting requires analysis of data received in the client request and can be challenging. Especially if there is heavy traffic flowing through the appliance. Diagnosing issues might affect the functionality or application security might require a quick response.

The Citrix ADC isolates traffic for a Web App Firewall profile and collects `nstrace` for the HTML requests. The `nstrace` collected in appfw mode includes request details with log messages. You can use “Follow TCP stream” in the trace to view the details of the individual transaction including headers, payload, and the corresponding log message in the same screen.

This gives you a comprehensive overview regarding your traffic. Having a detailed view of the request, payload, and associated log records can be useful to analyze security check violation. You can easily identify the pattern that is triggering the violation. If the pattern must be allowed, you can take a decision to modify the configuration or add a relaxation rule.

Benefits

1. **Isolate traffic for specific profile**: This enhancement is useful when you isolate traffic for only one profile or specific transactions of a profile for troubleshooting. You no longer have to skim through the entire data collected in the trace or need special filters to isolate requests interest you which can be tedious with heavy traffic. You can view the data that you prefer.
2. **Collect data for specific requests**: The trace can be collected for a specified duration. You can collect trace for only a couple of requests to isolate, analyze, and debug specific transactions if needed.
3. **Identify resets or aborts**: Unexpected closing of connections is not easily visible. The trace collected in –appfw mode captures a reset or an abort, triggered by the Web App Firewall. This allows a quicker isolation of an issue when you do not see a security check violation message. Malformed requests or other non-RFC compliant requests terminated by Web App Firewall will now be easier to identify.
4. **View decrypted SSL traffic**: HTTPS traffic is captured in plain text to allow for easier troubleshooting.
5. **Provides comprehensive view**: Allows you to look at the entire request at the packet level, check the payload, look at the logs to check what security check violation is being triggered and identify the match pattern in the payload. If the payload consists of any unexpected data, junk strings, or non-printable characters (null character, \r or \n and so forth), they are easy to discover in the trace.
6. **Modify configuration**: The debugging can provide useful information to decide if the observed behavior is the correct behavior or the configuration must be modified.

7. **Expedite response time**: Faster debugging on target traffic can improve the response time to provide explanations or root cause analysis by the Citrix engineering and support team.

For more information, see Manual Configuration by using the command line interface topic.

To configure debug tracing for a profile by using the command line interface

Step 1. Enable ns trace.

You can use the show command to verify the configured setting.

- `set appfw profile <profile> -trace ON`

Step 2. Collect trace. You can continue to use all the options which are applicable for the `nstrace` command.

- `start nstrace -mode APPFW`

Step 3. Stop trace.

- `stop nstrace`

Location of the trace: The `nstrace` is stored in a time-stamped folder which is created in the /var/nstrace directory and can be viewed using `wireshark`. You can tail the `/var/log/ns.log` to see the log messages providing details regarding the location of the new trace.

Tips:

- When the appfw mode option is used, the `nstrace` will only collect the data for one or more profiles for which the "nstrace" was enabled.
- Enabling the trace on the profile will not automatically start collecting the traces until you explicitly run the "start ns trace" command to collect the trace.
- Although enabling trace on a profile may not have any adverse effect on the performance of the Web App Firewall but you may want to enable this feature only for the duration for which you want to collect the data. It is recommended that you turn the –trace flag off after you have collected the trace. The option prevents the risk of inadvertently getting data from profiles for which you had enabled this flag in the past.
- The block or log action must be enabled for the security check for the transaction record to be included in the `nstrace`.
- Resets and aborts are logged independently of security checks actions when trace is "On" for the profiles.
- The feature is only applicable for troubleshooting the requests received from the client. The traces in –appfw mode do not include the responses received from the server.
- You can continue to use all the options which are applicable for the `nstrace` command. For example,


```
start nstrace -tcpdump enabled -size 0 -mode appFW
```

- If a request triggers multiple violations, the `nstrace` for that record includes all the corresponding log messages.
- CEF log message format is supported for this functionality.
- Signature violations triggering block or log action for request side checks will also be included in the trace.
- Only HTML (non-XML) requests are collected in the trace.

Web App Firewall support for cluster configurations

September 14, 2021

> **Note:**
>
> Citrix Web App Firewall for striped and partially striped configurations was introduced in Citrix ADC 11.0 version.

A cluster is a group of Citrix ADC appliances configured and managed as a single system. Each appliance in the cluster is called a node. Depending on the number of nodes the configurations are active on, cluster configurations are referred to as striped, partially striped, or spotted configurations. The Web App Firewall is fully supported in all configurations.

The two main advantages of striped and partially striped virtual server support in cluster configurations are the following:

1. Session failover support—striped and partially striped virtual server configurations support session failover. The advanced Web App Firewall security features, such as Start URL closure and the Form Field Consistency check, maintain, and use sessions during transaction processing. In a high availability configuration, or in a spotted cluster configuration, when the node that is processing the Web App Firewall traffic fails, all the session information is lost and the user has to re-establish the session. In striped virtual server configurations, user sessions are replicated across multiple nodes. If a node goes down, a node running the replica becomes the owner. Session information is maintained without any visible impact to the user.
2. Scalability—Any node in the cluster can process the traffic. Multiple nodes of the cluster can process the incoming requests served by the striped virtual server. This improves the Web App Firewall's ability to handle multiple simultaneous requests, thereby improving the overall performance.

Security checks and signature protections can be deployed without the need for any additional cluster-specific Web App Firewall configuration. You can do the usual Web App Firewall configuration on the configuration coordinator (CCO) node for propagation to all the nodes.

> **Note:**
>
> The session information is replicated across multiple nodes, but not across all the nodes in the striped configuration. Therefore, failover support accommodates a limited number of simultaneous failures. If multiple nodes fail simultaneously, the Web App Firewall might lose the session information if a failure occurs before the session is replicated on another node.

Highlights

- Web App Firewall offers scalability, high throughput, and session failover support in cluster deployments.
- All Web App Firewall security checks and signature protections are supported in all cluster configurations.
- Character-maps are not yet supported for a cluster. The learning engine recommends FieldTypes in learned rules for the Field Format security check.
- Stats and learned rules are aggregated from all the nodes in a cluster.
- Distributed Hash Table (DHT) provides the caching of the session and offers the ability to replicate session information across multiple nodes. When a request comes to the virtual server, the Citrix ADC appliance creates Web App Firewall sessions in the DHT, and can also retrieve the session information from the DHT.
- Clustering is licensed with the Advanced and Premium licenses. This feature is not available with the Standard license.

Debugging and troubleshooting

September 14, 2021

Refer to the following troubleshooting and debugging information related to each of the Web App Firewall functionality:

- Application Firewall - High CPU
- Memory
- Large File Upload Failure
- Learning
- Signatures
- Trace log
- Miscellaneous

High CPU

September 14, 2021

Following are some of the functionality and high CPU related debugging issues encoutered and the best practices to follow when working with Web App Firewall:

Check Policy hits, Bindings, Network configuration, Web App Firewall configuration:

- Identify misconfiguration
- Identify *vserver* that is serving the affected traffic

Inspect logs in the following log files for security violations and recent configuration changes:

- `/var/log/ns.log`
- `/var/nslog/import.log`
- `/var/nslog/aslearn.log`
- `tail -f /var/log/ns.log | grep APPFW_SIGNATURE_MATCH`

Example:

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW| APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method
    =GET request= http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=
    Signature violation rule ID 807: web-cgi /wwwboard/passwd.txt access
     cn1=140 cn2=841 cs1=pr_ffc cs2=PPE0 cs3=
    OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    not blocked
<!--NeedCopy-->
```

Isolate the traffic that is effected:

- Isolate the profile
- Isolate the security check
- Isolate the URL, virtual server and traffic parameters

Conditional profile level trace helps identify the traffic and violation records:

- `set appfw profile <profile> -trace ON`
- `start nstrace -mode APPFW -size 0`
- `stop nstrace`

Note: Ensure that the trace is collected with -size 0 option.

Check appfw, dht, IP reputation activity counters:

- `nsconmsg -g as_ -g appfwreq_ -g iprep -d current`

Monitor window size for resets in connection:

Appfw sets the window size to 9845 when Citrix ADC resets the connection due to an invalid http message.

Examples:

- Malformed request received - connection reset
- High CPU related issues
- Check data sheets for system limits
- Inspect for cpu usage, appfw, DHT and memory related activity. Monitor appfw sessions
- nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -g mem_AS_OBJ -g mem_AS_COMPONENT -d current

Monitor memory allocated and freed from Web App Firewall components and objects during the target time period. It helps in isolating the protection leading to high CPU usage.

- Profiler output
- Observe logs

Isolate appfw check leading to high CPU:

- startURLClosure
- Formfiledconsistency
- CSRF
- Cookie protections
- Referer header check

Ascertain that autoupdate of signatures is not leading to high CPU (Disable to confirm).

Memory

September 14, 2021

Following are some of the best practices to follow when encountered with Web App Firewall usage memory related issues:

nsconmsg command usage:

- Look for global memory statistics to ascertain that there is enough memory in the system and there are no memory allocation failures by executing the following command:

  ```
  * *- nsconmsg -d memstats
  ```

- Observe current allocated and maximum memory limits for appsecure, IP reputation, cache and compression by executing the following command:


```
nsconmsg -d memstats | egrep -i APPSECURE|IPREP|CACHE|CMP
```

- Check appfw, DHT, IP reputation activity counters by executing the following command:

  ```
  nsconmsg -g as -g appfwreq_ -g iprep -d current
  ```

- Check all Web App Firewall error counters by executing the following command:

  ```
  nsconmsg -g as_ -g appfwreq_ -g iprep_ -d stats | grep err
  ```

- Check all system error counters by executing the following command:

  ```
  nsconmsg -g err -d current
  ```

- Inspect for CPU, APPFWREQ, AS and DHT counters by executing the following command:

  ```
  nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -d current
  ```

- Check the configured Cache memory by executing the following command:
- `show cacheparameter`
- Check the configured memory by executing the following command:

  ```
  nsconmsg -d memstats | egrep -i CACHE
  ```

- Identify distribution of memory in Web App Firewall components and objects:

Display AS_OBJ_ memory:

```
nsconmsg -K newnslog -d stats | grep AS_OBJ | egrep -v AppFW_cpu0|total |
sort -k3
```

Display AS_COMPONENT_ memory:

```
nsconmsg -K newnslog -d stats | grep AS_COMPONENT | egrep -v AppFW_cpu0|
total | sort -k3
```

Check for number of alive sessions by executing the following command:

Monitor/plot active session counts:

```
nsconmsg -g as_alive_sessions -d current
```

Monitor/plot total allocated, free, updated sessions:

- `nsconmsg -g as_tot_alloc_sessions -g as_tot_free_sessions -d current`
- `nsconmsg -g as_tot_update_sessions -d current`

If required, reduce session timeout to ensure that session limits are not used by executing the following command:

```
set appfwsettings -sessionTimeout <300>
```

If required, set maximum lifetime of session by executing the following command:

```
set appfwsettings -sessionLifetime <7200>
```


Checking allocated and used memory

To check the total allocated memory and used memory:

- Use the **nsconmsg –d memstats** command. Observe the **MEM_APPSECURE** field.
- Use the **stat appfw** command to obtain meory consumption information.

Web App Firewall does not automatically delete the logs after certain period of time or size.

- `All AppFw logs are archived in the */var/log/ns.log*` file. The *ns.log* file performs the rollover task.

For more information, refer to the following link: <http://support.citrix.com/article/CTX121898>

Increasing Web App Firewall memory:

- There is no CLI option to increase Web App Firewall memory. Web App Firewall memory is platform-specific.
- You may use the *nsapimgr* option to increase memory but it is not recommended.

The max allowed memory for Web App Firewall is determined by the platform and disabling IC does not impact memory allocation.

Large file upload failures

September 14, 2021

When you encounter large file upload failures, ensure that you check the following:

– Misconfigured application firewall postbody limit
– Enabled file upload scanning leading to increased processing time.
– Hitting system limits.

For payloads greater than 20 MB, Citrix recommends you to enable streaming on the application firewall profile. Also, you must ensure the back-end server supports chunked requests before enabling streaming.

Since release 11.0, the streaming flag can be enabled on per profile basis to avoid buffering by executing the following command:

```
set appfw profile <profile name> -streaming on
```


Learning

September 14, 2021

Following are some of the best practices recommended when encountered with Learning functionality issues:

Aslearn process:

- Verify that the process *aslearn* is running.
- Check top command output
- Check output of ps command by executing the following command:

  ```
  ps -ax | grep aslearn | grep -v "grep"
  ```

  Example:

  ```
  root@ns\# ps -ax | grep aslearn | grep -v "grep"
  1439  ??  Ss     0:03.86 /netscaler/aslearn -start -f /netscaler/
     aslearn.conf
  <!--NeedCopy-->
  ```

- Identify recent configuration commands ran prior to the observed problem by verifying the *ns.log* file:

  ```
  /var/log/ns.log
  ```

- Inspect aslearn logs to check for aslearn messages:

  ```
  /var/log/aslearn.log
  ```

- Isolate the profile and security check that is effected
- Identify the GUI and CLI command which is failing by executing the following command:

  ```
  show appfw learningdata <profileName> <securityCheck>
  ```

  Examples:

  - `show learningdata test_profile starturl`
  - `show learningdata test_profile crosssiteScripting`
  - `show learningdata test_profile sqLInjection`
  - `show learningdata test_profile csRFtag`
  - `show learningdata test_profile fieldformat`
  - `show learningdata test_profile fieldconsistency`

- Perform integrity check of sqlite from bsd shell prompt:


```
nsshell ## sqlite3 /var/nslog/asl/<profile_name_in_lowercase>.db '
pragma integrity_check;
```

Examples:

```
root@ns# sqlite3 /var/nslog/asl/tsk0247284.db 'pragma
   integrity_check;'
ok
<!--NeedCopy-->
```

- Deploy or remove rules to start learning again:
  - If 2000 learn items (per protection) are reached, you cannot start learning any more for that protection
  - If 20 MB size is reached for the database, stop learning for all protections
  - Restart aslearn process

```
*/netscaler/aslearn -start -f/netscaler/aslearn.conf*
```

- Check the space in the /var folder by executing the following:

```
du -h /var
```

- Check the learning threshold limits by executing the following command:

```
show appfwlearningsettings <profile_name> <securityCheck>
```

- Collect learned data by executing the following command:

```
export appfwlearningdata <profile_name> <securityCheck>
```

- Ascertain that learned data is uploaded in the collector.

Signatures

September 14, 2021

Getting started with signatures

To add signature:

1. Select the **Default**-signature and click **add** to make a copy.
2. Give a meaningful name. The new sig object is added as a User-Defined object.
3. Enable the target rules that are pertinent to your specific need.
   - The rules are disabled by default.

    - more rules require more processing

4. Configure the actions:

   Block and Log actions are enabled by default. Stats is another option

5. Set the signature to be used by your profile.

Tips for Using Signatures

- Optimize the processing overhead by enabling only those signatures that are applicable for protecting your application.
- Every pattern in the rule must match to trigger a signature match.
- You can add your own customized rules to inspect incoming requests to detect various types of attacks, such as SQL injection or cross-site scripting attacks.
- You can also add rules to inspect the responses to detect and block leakage of sensitive information such as credit card numbers.
- Add multiple security check conditions to create your own customized check.

Best Practices for Using Signatures

Following are some of the best practices you can follow when encountered with issues related to Signatures:

- Verify that the import command has succeeded on primary as well as secondary.
- Verify that CLI and GUI outputs are consistent.
- Check ns.log to identify any errors during signature import and auto update.
- Check if the DNS name server is configured properly.
- Check schema version incompatibility.
- Check if the device is unable to access the Signature Update URL hosted on AWS for auto-update.
- Check for the version mismatch between Default signature and user-added ones.
- Check for version mismatch between signature objects on the primary and secondary nodes.
- Monitor for High CPU Utilization (disable auto-update to rule out issue with signature update).

Trace Log

September 14, 2021

To record trace logs:

1. Enable tracing for the profile. You can use the show command to verify the configured setting.

```
set appfw profile <profile> -trace ON
```

1. Start collecting trace. You can continue to use all the options which are applicable for the nstrace command.

```
start nstrace -mode APPFW
```

1. Stop collecting the trace

```
stop nstrace
```

Location of the trace: The nstrace is stored in a time-stamped folder which is created in the */var/nstrace* directory and can be viewed using wireshark. You can tail the */var/log/ns.log* file to see the log messages providing details regarding the location of the new trace.

Advantages of trace logs:

- Isolate traffic for specific profile
- Collect data for specific requests
- Identify resets or aborts
- View decrypted SSL traffic: HTTPS traffic is captured in plain text to allow for easier troubleshooting.
- Provides comprehensive view: Allows you to look at the entire request at the packet level, check the payload, view logs to check what security check violation is being triggered and identify the match pattern in the payload. If the payload consists of any unexpected data, junk strings, or non-printable characters (null character, \r or \n etc), they are easy to discover in the trace.
- Expedite response time: Faster debugging on target traffic to do root cause analysis.

Miscellaneous

September 14, 2021

Following are the resolutions for some of the issues that you might encounter when using Web App Firewall.

- Web App Firewall sets window size to 9845 when resetting connection for invalid http messages.

  – Malformed request received - connection reset [Client/Server sending invalid content-length header]
  – Unknown content-type in request headers

- System Limit: the application appears frozen

– Occurs when maximum session limit is reached. (100K)
– Less system memory for operation.

- IP Reputation feature not working
  – The iprep process takes about five minutes to start after you enable the reputation feature. The IP reputation feature might not work for that duration.

- Unexpected Web App Firewall violations being triggered

  – Session timeout has a default value of 900 seconds. If session timeout is set to a low value, browser may trigger false positives for checks which rely on sessionization (e.g CSRF, FFC). Check for session timeout and look at the session ID (cs3 in CEF logs). If the sessionID is different, the session timeout might be the reason.
  – If form is dynamically generated by javascript, it may trigger false FFC violations.

- Empty field name in FFC violation logs (prior to 11.0 release)

  This may be seen in scenarios where we come across a form field which is not in the forms in our session.

  Scenarios where this may occur:

  – The session has timed out from when the form was sent to the client and when it was received.
  – The form was generated on the client side using a java script.

References

September 14, 2021

Refer to the following additional reosurces for more information and useful tips when using Web App Firewall:

- HowCitrix Web App Firewall Modifies Application DataTraffic

  – Conditional headers modified by Web App Firewall.
  – Integrated caching and Web App Firewall interoperability.

- TraceHTML Requests with Web App Firewall Security Violation Logs on NetScalerAppliance

  - Isolating request and debugging the end-to-end transaction.

- Top Level Protection

  - security relaxations

- Information about configuring and deploying.

  - Application

  - Firewall
  - Logs

- Details regarding anatomy of the Web App Firewall log messages.

  <https://regex101.com/>

- Configuring Regular expressions.
- Datasheets

  – Using recommended memory and CPU for system.

  – Ensuring enough memory for Web App Firewall and configuring cache limit appropriately.

Signature alert Articles

September 14, 2021

Citrix Web Application Firewall (WAF) announces signature updates that you can download and apply on your appliance. When you detect a security attack, you will receive an email notification about the new signature update. You can download the signature and apply it on your appliance.

How to receive signature alert notification

September 14, 2021

This article explains how to configure signature alert settings to receive email notifications for new signature updates.

Summary

Network administrators would like to receive an email notification for new Web Application Firewall signature updates and notification.

Problem

A Network administrator who wants to be notified when a new signature is available for the Web Application Firewall can elect to be notified by email. The administrator will receive an email notification when new signatures are available to be downloaded.
Network administrators to receive email notification for new signature updates.

Solution

To receive email notification for new signature updates, follow the steps given below:

1. Sign into Citrix support website, `https://support.citrix.com/user/alerts`.
2. In the **Alert Settings** section, enable the Notify me through email option.
3. Select **Add Products** to view the product catalog.
4. Click **Citrix Web App Firewall** and then select **Citrix Web App Firewall** check box.
5. Click **Save Settings**.



1. In the **Alert Settings** section, enable the Notify me through email option.
2. Select **Add Products** to view the product catalog.
3. Click **Application Firewall** and then click the **Signatures** check box.
4. Click **Save Settings**.

Signature update version 27

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 27. You can download and configure these signature rules to protect your appliance from security vulnerable attacks. The signature update includes the signature ID, signature version, and list of CVEs addressed.

Signature version

Signature version 27 applicable to NetScaler VPX 11.1, NetScaler 12.0, and Citrix ADC 12.1 platforms.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999921	cve-2018-1002000	WEB-MISCWordpress Arigato Autoresponder and Newsletter SQL Injection vulnerability.
999920		WEB-MISCWordPress plug-in Corner Ad 1.0.7 - Stored Cross-Site Scripting
999919	cve-2018-1002009	WEB-MISCWordpress Arigato Autoresponder and Newsletter bft_unsubscribe cross-site scripting vulnerability.
999918	cve-2018-1002002	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002003	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.

Signature rule	CVE ID	Description
999918	cve-2018-1002004	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002005	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002006	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002007	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999917	cve-2018-1002001	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999917	cve-2018-1002008	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999916	cve-2018-8719	WEB-MISCWordPress plug-in WP Security Audit Log - wp-content/uploads/wp-security-audit-log/* unrestricted access
999915	cve-2019-7743	WEB-MISC- Joomla phar:// stream wrapper object injection vulnerability execution of uploaded non-phar files

Signature rule	CVE ID	Description
999914		WEB-MISCWordpress plug-in E-mail Subscribers and Newsletters 3.4.7 information disclosure vulnerability
999913		WEB-MISCWordPress plug-in AD Manager WD v1.0.11 - wd_ads_admin_class.php Arbitrary File Download
999912		WEB-IISMicrosoft IIS - Short File/Folder Name Disclosure

Signature update version 28

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 28. You can download and configure these signature rules to protect your appliance from security vulnerable attacks. The signature update includes the signature ID, signature version, and list of CVEs addressed.

Signature version

Signature version 28 applicable to NetScaler VPX 11.1, NetScaler 12.0, and Citrix ADC 12.1 and Citrix ADM 13.0 platforms.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999898	CVE-2018-12895	WEB-MISC WordPress before 4.9.7-Directory Traversal Vulnerability.

Signature rule	CVE ID	Description
999899	CVE-2019-9618	WEB-MISC-GraceMedia Media Player WordPress plug-in 1.0 Arbitrary Local File Inclusion Vulnerability
999900	CVE-2018-20714	WEB-MISC WordPress plug-in WooCommerce before 3.4.6 - File Deletion Vulnerability.
999901	CVE-2018-11868	WEB-MISC FlowPaper FlexPaper before 2.3.7 can Allow Remote Code Execution-Reset of Config Files.
999902	CVE-2018-11868	WEB-MISC FlowPaper FlexPaper before 2.3.7 can Allow Remote Code Execution.
999903	CVE-2019-9184	WEB-MISC-Joomla! J2Store plug-in 3.x Before 3.3.7 Allows SQL Injection.
999904	CVE-2019-9168	WEB-MISC WordPress plug-in WooCommerce before 3.5.5-cross-site scripting via Photoswipe caption.
999905		WEB-MISC WordPress plug-in Abandoned Cart before 5.1.3 for WooCommerce-Stored Cross-Site Scripting.
999906	CVE-2019-8942	WEB-MISC WordPress before 4.9.9 and 5.x before 5.0.1-remote code execution.
999907	CVE-2019-8942	WEB-MISC WordPress before 4.9.9 and 5.x before 5.0.1-remote code execution.

Signature rule	CVE ID	Description
999908	CVE-2019-8942	WEB-MISC WordPress before 4.9.9 and 5.x before 5.0.1-remote code execution
999909	CVE-2017-16562	WEB-MISC-Deluxe Theme UserPro WordPress plug-in Security Bypass Vulnerability Via up_auto_log=true Parameter
999910	CVE-2018-20782	WEB-MISC WordPress plug-in GloBee before 1.1.2 for WooCommerce-IPN Messages Spoofing
999911	CVE-2019-6340	Drupal-Arbitrary Remote Code Execution in Drupal Core 8 RESTFul WebServices

Signature update version 29

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 29. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 29 applicable to NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, and Citrix ADC 13.0 platforms.

> **Note**:
>
> Enabling Post body and Response body signature rules may affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999896	CVE-2019-2725	Weblogic 10.3.6 Remote Code Execution
999897	CVE-2019-2725	Weblogic 10.3.6 Remote Code Execution

Signature update version 30

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 30. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 30 applicable to NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, and Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules may affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999879	<>	WEB-MISC WordPress plug-in WooCommerce Checkout Manager - Arbitrary File Upload Vulnerability
999880	<>	WEB-MISC WordPress plug-in Advance Contact Form 7 DB Prior To 1.6.1 - SQL Injection Vulnerability

Signature rule	CVE ID	Description
999881	<>	WEB-MISC WordPress plug-in Contact Form Builder Prior To 1.0.67 - Local File Inclusion Vulnerability
999882	<>	SQL HTTP URI Blind Injection Attempt
999883	<>	WEB-MISC Loco Translate WordPress plug-in 2.1.1 and prior - Local File Inclusion Vulnerability
999884	<>	WEB-MISC WordPress plug-in Duplicate-Page Prior To 3.4 - SQL Injection Vulnerability
999885	CVE-2019-0232	WEB-MISC Apache Tomcat RCE Via .CMD CGI Scripts When enableCmdLineArguments=true in MS Windows
999886	CVE-2019-0232	WEB-MISC Apache Tomcat RCE Via .BAT CGI Scripts When enableCmdLineArguments=true in MS Windows
999887	CVE-2019-10692	WWEB-MISC WordPress plug-in wp-google-maps Prior To 7.11.18 - SQL Injection Vulnerability.
999888	CVE-2019-10946	WEB-MISC Joomla! Prior To 3.9.5 - Security Bypass Vulnerability
999889	CVE-2019-10945	WEB-MISC Joomla! Prior To 3.9.5 - Directory Traversal Vulnerability
999890	CVE-2019-9912	WEB-MISC WpGoogleMaps WordPress plug-in prior to 7.10.41 Reflected cross-site scripting Vulnerability

Signature rule	CVE ID	Description
999890	CVE-2019-9912	WEB-MISC WpGoogleMaps WordPress plug-in prior to 7.10.41 Reflected cross-site scripting Vulnerability
999891	CVE-2019-9911	WEB-MISC WordPress plug-in Social Networks Auto-Poster Prior To 4.2.8 - Reflected cross-site scripting Vulnerability
999892	CVE-2019-9908	WEB-MISC WordPress plug-in Font_Organizer 2.1.1 - Reflected cross-site scripting
999893	CVE-2019-9787	WEB-MISC WordPress before 4.9.7 - Remote Code Execution Vulnerability
999894	CVE-2019-9568	WEB-MISC Forminator Contact Form, Poll & Quiz Builder WordPress plug-in prior to 1.6 Blind SQLi Vulnerability
999895	CVE-2019-9567	WEB-MISC Forminator Contact Form, Poll & Quiz Builder WP plug-in prior to 1.6 Persistent cross-site scripting Vulnerability
999877	CVE-2018-20062	WEB-MISC NoneCms V1.3 - ThinkPHP Filter Arbitrary PHP Code Execution Vulnerability
999878	CVE-2019-9082	WEB-MISC Remote Code Execution Vulnerability in ThinkPHP 5.x prior to 5.1.32

Signature update version 32

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 32. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 32 applicable to NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, and Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules may affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999875	CVE-2016-4438, CVE-2016-3087	WEB-STRUTS Apache Struts 2.3.20 Through 2.3.28.1 Remote Execution Vulnerability Via URL
999876	CVE-2019-10867	WEB-MISC Pimcore Prior to 5.7.1 - Deserializing Vulnerability (CVE-2019-10867)

Signature update version 33

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 33. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 33 applicable to NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, and Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules may affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Rule	CVE	Description	Vulnerability Reference
999860		WordPress plug-in Yuzo Related Posts cross-site scripting Vulnerability	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild
999861	CVE-2019-12099		cve,2019-12099
999862		WordPress plug-in Database Backup <= 5.2 - Remote Code Execution	https://www.wordfence.com/blog/2019/05/os-command-injection-vulnerability-patched-in-wp-database-backup-plug-in
999863		WordPress plug-in Slick Popup - Privilege Escalation	https://www.wordfence.com/blog/2019/05/privilege-escalation-flaw-present-in-slick-popup-plug-in

Rule	CVE	Description	Vulnerability Reference
999864	CVE-2019-10866	WordPress plug-in Form Maker 1.13.3 - SQL Injection	cve,2019-10866
999865		WordPress plug-in Give – Stored cross-site scripting for Donors	https://blog.sucuri.net/2019/05/wordpress-plug-in-give-stored-xss-for-donors.html
999866		WordPress plug-in My Calendar <= 3.1.9 - Unauthenticated cross-site scripting Vulnerability	https://wpvulndb.com/vulnerabilities/9267
999867		WordPress plug-in Slimstat <= 4.8 - Unauthenticated Stored cross-site scripting	https://blog.sucuri.net/2019/05/slimstat-stored-xss-from-visitors.html
999868	CVE-2019-2618	WebLogic Arbitrary Upload Vulnerability	cve,2019-2618
999869	CVE-2019-11871	WEB-WORDPRESS WordPress plug-in Custom Field Suite Prior To 2.5.15 - Cross-Site Scripting Vulnerability	cve,2019-11871

Rule	CVE	Description	Vulnerability Reference
999870		WEB-WORDPRESS WordPress Live Chat Support plug-in Persistent cross-site scripting Vulnerability prior 8.0.27 via wplc_custom_js parameter	https://blog.sucuri.net/2019/05/persistent-cross-site-scripting-in-wp-live-chat-support-plug-in.html
999871		WEB-WORDPRESS WordPress plug-in W3 Total Cache Prior To 0.9.7.4 - PHAR Remote Code Execution Vulnerability	https://wpvulndb.com/vulnerabilities/9270
999872		WEB-WORDPRESS WordPress plug-in W3 Total Cache Prior To 0.9.7.4 - PHAR Remote Code Execution Vulnerability	https://wpvulndb.com/vulnerabilities/9269
999873	CVE-2019-0604	WEB-MISC Microsoft Windows Sharepoint Server - Remote Code Execution Vulnerability	cve,2019-0604
999874		WEB-WORDPRESS Yuzo Related Posts Unauthenticated Stored cross-site scripting Vulnerability in 5.12.91	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild

Signature update version 34

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 34. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 34 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, and Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999843		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - Setting Arbitrary File For Read
999844		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - Arbitrary File Read
999845		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - File Removal Via File Replacement
999846		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - File Removal

Signature rule	CVE ID	Description
999847		WEB-WORDPRESS WordPress plug-in Shortlinks Prior To 2.1.10 - CSV Injection Vulnerability
999848		WEB-WORDPRESS WordPress plug-in Shortlinks Prior To 2.1.10 - Unauthenticated Stored cross-site scripting Vulnerability
999849		WEB-WORDPRESS WordPress plug-in FV Flowplayer Video Player Prior To 7.3.13.727 - Unauthenticated Stored cross-site scripting Vulnerability
999850		WEB-WORDPRESS WordPress plug-in Easy Digital Downloads Prior To 2.9.16 - Unauthenticated Stored cross-site scripting Vulnerability
999851		WEB-WORDPRESS WordPress plug-in Crelly Slider Prior to version 1.3.5 - Arbitrary File Upload Vulnerability
999853	CVE-2019-2615	WEB-MISC Oracle WebLogic Server Information Disclosure Vulnerability
999854	CVE-2019-11872	WordPress plug-in Hustle Prior To 6.0.8.1 - CSV Injection Vulnerability
999855	CVE-2019-11231	WEB-MISC GetSimple CMS Version 3.3.15 and Prior - Arbitrary File Upload Vulnerability

Signature rule	CVE ID	Description
999856	CVE-2019-11231	WEB-MISC GetSimple CMS Version 3.3.15 and Prior - API Key Information Disclosure
999857		WEB-WORDPRESS WordPress plug-in WP Database Backup Prior To 5.2 - Command Injection Vulnerability
999858		WEB-WORDPRESS WordPress plug-in Slick Popup Up To 1.7.1 - Privilege Escalation Vulnerability
999859	CVE-2019-12099	WEB-MISC PHP Fusion CMS Remote Code Execution Vulnerability in Version 9.03.00 and Prior

Signature update version 35

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 35. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 35 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, and Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999834	CVE-2019-13024	WEB-MISC Centreon Version 19.04 and Prior - Command Injection Vulnerability
999835	CVE-2019-5420	WEB-MISC Rails Development Mode - Secret Token Disclosure Vulnerability
999836	CVE-2019-5418	WEB-MISC Rails Action View - File Content Disclosure Vulnerability
999837	CVE-2018-12426, CVE-2019-11185	WEB-WORDPRESS WP Live Chat Support Pro plug-in Prior to 8.0.26 - Arbitrary File Upload
999838	CVE-2019-10270	WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.40 - Arbitrary Password Reset
999839	CVE-2019-12826	WEB-WORDPRESS WordPress plug-in Widget Logic Prior To 5.10.2 - CSRF Vulnerability
999840		WEB-WORDPRESS WordPress plug-in All-In-One Event Calendar Prior To 2.5.39 - cross-site scripting Vulnerability
999841	CVE-2019-11565	WEB-WORDPRESS WordPress plug-in Print My Blog Prior To 1.6.7 - Unauthenticated SSRF Vulnerability
999842		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - Multiple `cross-site scripting</LogString`

Signature update version 36

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 36. You can download and configure the signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 36 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note**:
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999817		WEB-WORDPRESS WordPress Ad Inserter plug-in Prior to Version 2.4.22 - Remote Code Execution
999818	CVE-2019-7839	WEB-MISC Adobe ColdFusion Multiple Versions - Remote Code Execution Vulnerability Via HTTP/SOAP DotNet-to-Java (CVE-2019-7839)
999819	CVE-2019-7839	WEB-MISC Adobe ColdFusion Multiple Versions - Remote Code Execution Vulnerability Via HTTP/SOAP Java-to-DotNet (CVE-2019-7839)

Signature rule	CVE ID	Description
999820	CVE-2019-11469	WEB-MISC Zoho ManageEngine Applications Manager Prior to 14 Build 14150 Allows SQLi Via resourceid Parameter (CVE-2019-11469)
999821	CVE-2019-11448	WEB-MISC Zoho ManageEngine Application Manager 11.0 Through 14.0 - Unauthenticated SQL Injection (CVE-2019-11448)
999822	CVE-2019-1003000	WEB-MISC Jenkins Script Security plug-in Up To 1.49 - Sandbox Bypass Vulnerability (CVE-2019-1003000)
999823		WEB-WORDPRESS WordPress Cforms2 plug-in Up To 15.0.1 - Unauthenticated HTML Injection Vulnerability
999824	CVE-2019-0193	WEB-MISC Apache Solr Prior To 8.2 - DIH Remote Code Execution Vulnerability Via dataConfig Parameter (CVE-2019-0193)
999825	CVE-2019-11580	WEB-MISC Atlassian Crowd Pdkinstall Development plug-in Enabled - Unauthenticated RCE (CVE-2019-11580)
999826	CVE-2019-0192	WEB-MISC Apache Solr Up To 5.5.5 / 6.6.5 - Config API Remote Code Execution Vulnerability (CVE-2019-0192)

Signature rule	CVE ID	Description
999827		WEB-WORDPRESS WooCommerce Variation Swatches plug-in Up To 1.0.61 - Reflected cross-site scripting Vulnerability
999828	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy plug-in Up To 2.61 - Sandbox Bypass Vulnerability Via Job Creation (CVE-2019-1003001)
999829	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy plug-in Up To 2.61 - Sandbox Bypass Vulnerability (CVE-2019-1003001)
999830		WEB-WORDPRESS WordPress Bold Page Builder plug-in Prior To 2.3.2 - Security Bypass Vulnerability
999831	CVE-2019-15107	WEB-MISC Webmin Prior To 1.930 - Unauthenticated Remote Code Execution Vulnerability (CVE-2019-15107)
999832	CVE-2019-2767	WEB-MISC Oracle BI Publisher 11.1.1.9.0 and 12.2.1.4 - XXE Vulnerability (CVE-2019-2767)
999833	CVE-2019-15106	WEB-MISC Zoho ManageEngine OpManager Through 12.4x - Authentication Bypass Vulnerability (CVE-2019-15106)

Signature rule	CVE ID	Description
999948	CVE-2014-0114	Apache Struts 1 through 1.3.10 allows ClassLoader manipulation allowing arbitrary code execution via HTTP_FORM_FIELD
999949	CVE-2013-4316	Apache Struts 2 before 2.3.15.2 allows Dynamic Method Invocation by affecting confidentiality, integrity or availability
999950	CVE-2013-4316	Apache Struts 2 before 2.3.15.2 allows Dynamic Method Invocation by affecting confidentiality, integrity or availability

> **Note:**
>
> Signature rule 999947 is deleted because of performance issue.

Signature update version 37

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 37. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 37 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999806	CVE-2019-3394	WEB-MISC Atlassian Confluence or Data Center - Local File Disclosure Vulnerability (CVE-2019-3394)
999807	CVE-2019-13569	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.1.8 - SQLi Via Esfpx_lists Param (CVE-2019-13569)
999808	CVE-2019-13569	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.1.8 - SQLi Via Order Param (CVE-2019-13569)
999809	CVE-2019-2768	WEB-MISC Oracle BI Publisher - Predictable Session Token Vulnerability (CVE-2019-2768)
999810	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy plug-in Up To 2.61 - Sandbox Bypass Vulnerability Via Job Update (CVE-2019-1003001)
999811	CVE-2019-13575	WEB-WORDPRESS WPEverest Everest Forms plug-in Prior to 1.5.0 - SQL Injection (CVE-2019-13575)
999812	CVE-2019-15896	WEB-WORDPRESS LifterLMS plug-in Up To 3.34.5 - Security Bypass Vulnerability (CVE-2019-15896)

Signature rule	CVE ID	Description
999813	CVE-2019-3396	WEB-MISC Atlassian Confluence or Data Center - Remote Code Execution Vulnerability (CVE-2019-3396)
999814	CVE-2019-5475	WEB-MISC Sonatype Nexus Repository Manager Prior to 2.14.14 - Remote Code Execution Via Createrepo Path (CVE-2019-5475)
999815	CVE-2019-5475	WEB-MISC Sonatype Nexus Repository Manager Prior to 2.14.14 - Remote Code Execution Via Mergerepo Path (CVE-2019-5475)
999816	CVE-2019-15104	WEB-MISC Zoho ManageEngine OpManager Version Prior to 12.4 - SQL Injection Vulnerability (CVE-2019-15104)

Signature update version 38

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in version 38. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 38 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999800	CVE-2019-12517	WEB-WORDPRESS SlickQuiz plug-in Version 1.3.7.1 and Prior - cross-site scripting Vulnerability (CVE-2019-12517)
999801	CVE-2019-10392	WEB-MISC Jenkins Git Client plug-in 2.8.4 And Prior - OS Command Injection Vulnerability (CVE-2019-10392)
999802	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Filedata Field (CVE-2019-8371)
999803	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Image Field (CVE-2019-8371)
999804	CVE-2019-12516	WEB-WORDPRESS SlickQuiz plug-in Version 1.3.7.1 and Prior - SQL Injection Vulnerability (CVE-2019-12516)
999805	CVE-2019-1262	WEB-MISC Microsoft Sharepoint Server - Cross Site Scripting Vulnerability (CVE-2019-1262)

Signature update for December 2019

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2019-12-19. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 39 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999760		WEB-MISC FusionPBX Versions Prior to 4.4.7 and 4.5.5 - Remote Code Execution Vulnerability Via /app/exec/exec.php
999761	CVE-2019-12747	WEB-MISC Typo3 Prior to 8.7.27 and 9.5.8 - Deserialization of Untrusted Data (CVE-2019-12747)
999762	CVE-2019-13608	WEB-MISC Citrix StoreFront Server - XML External Entity Injection Vulnerability (CVE-2019-13608)
999763		WEB-WORDPRESS WordPress Prior To 5.2.4 - Unauthenticated View Of Private or Draft Posts/Pages Vulnerability Via FORM

Signature rule	CVE ID	Description
999764		WEB-WORDPRESS WordPress Prior To 5.2.4 - Unauthenticated View Of Private or Draft Posts/Pages Vulnerability Via URL
999765	CVE-2019-15954	WEB-MISC Total.js CMS 12.0.0 - Widget JavaScript Code Injection Vulnerability Via JSON (CVE-2019-15954)
999766	CVE-2019-15954	WEB-MISC Total.js CMS 12.0.0 - Widget JavaScript Code Injection Vulnerability Via FORM (CVE-2019-15954)
999767		WEB-WORDPRESS SyntaxHighlighter Evolved plug-in Prior To 5.3.1 - Stored Cross-Site Scripting Vulnerability Via Comment
999768		WEB-WORDPRESS SyntaxHighlighter Evolved plug-in Prior To 5.3.1 - Stored Cross-Site Scripting Vulnerability Via POST
999769		WEB-WORDPRESS SyntaxHighlighter Evolved plug-in Prior To 5.3.1 - Stored Cross-Site Scripting Vulnerability Via JSON
999770	CVE-2019-16120	WEB-WORDPRESS Event Tickets plug-in Before 4.10.7.2 - CSV Injection Vulnerability (CVE-2019-16120)
999771	CVE-2019-15029	WEB-MISC FusionPBX Prior to 4.4.8 - Remote Code Execution Vulnerability (CVE-2019-15029)

Signature rule	CVE ID	Description
999772		WEB-WORDPRESS Sassy Social Share plug-in Prior To 3.3.4 - Unauthenticated Cross-Site Scripting Vulnerability
999773		WEB-WORDPRESS Email Subscribers & Newsletters plug-in Version 4.3.1 and Prior - Unauthenticated Blind SQLi Vulnerability
999774	CVE-2019-3398	WEB-MISC Atlassian Confluence or Data Center - downloadallattachments Path Traversal Vulnerability (CVE-2019-3398)
999775	CVE-2019-15952	WEB-MISC Total.js CMS 12.0.0 - Page Template Path Traversal Vulnerability (CVE-2019-15952)
999776	CVE-2019-17236	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up To 3.4.0 - Stored cross-site scripting (CVE-2019-17236)
999777	CVE-2019-10475	WEB-MISC Jenkins Build-Metrics plug-in 1.3 - Reflected cross-site scripting Vulnerability (CVE-2019-10475)
999778	CVE-2019-17132	WEB-MISC vBulletin Prior to 5.5.4 Patch Level 2 - UpdateAvatar API Endpoint Remote Code Execution Vulnerability (CVE-2019-17132)

Signature rule	CVE ID	Description
999779	CVE-2019-14994	WEB-MISC Atlassian Jira Service Desk - Path Traversal Vulnerability (CVE-2019-14994)
999780	CVE-2019-19367	WEB-MISC FusionPBX 4.4.1 and Prior - Cross-Site Scripting Vulnerability (CVE-2019-19367)
999781	CVE-2019-18668	WEB-WORDPRESS Currency Switcher plug-in Before 2.11.2 - Currency Setting Bypass Vulnerability Via POST (CVE-2019-18668)
999782	CVE-2019-18668	WEB-WORDPRESS Currency Switcher plug-in Before 2.11.2 - Currency Setting Bypass Vulnerability Via GET (CVE-2019-18668)
999783	CVE-2019-16663	WEB-MISC rConfig 3.9.2 and Prior - Remote Code Execution Vulnerability via Search.crud.php (CVE-2019-16663)
999784		WEB-MISC Apache Solr Up to 8.3.0 - Unauthenticated Remote Code Execution Via VelocityResponseWriter Custom Template
999785	CVE-2019-17235	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up To 3.4.0 - Information Disclosure Via Csv (CVE-2019-17235)

Signature rule	CVE ID	Description
999786	CVE-2019-17235	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up To 3.4.0 - Information Disclosure Via Bcc (CVE-2019-17235)
999787	CVE-2019-12276	WEB-MISC GrandNode 4.40 - LetsEncryptController Path Traversal Vulnerability (CVE-2019-12276)
999788		WEB-WORDPRESS Email Subscribers & Newsletters plug-in Prior to Version 4.2.3 - Unauthenticated Information Disclosure
999789	CVE-2019-4013	WEB-MISC IBM BigFix Platform 9.5 - Authenticated Arbitrary File Upload With Root Privileges (CVE-2019-4013)
999790	CVE-2019-11409	WEB-MISC FusionPBX Version 4.4.3 and Prior - Remote Code Execution Via /app/basic_operator_panel/exec.php (CVE-2019-11409)
999791	CVE-2019-11409	WEB-MISC FusionPBX Version 4.4.3 and Prior - Remote Code Execution Via /app/operator_panel/exec.php (CVE-2019-11409)
999792	CVE-2019-16662	WEB-MISC rConfig 3.9.2 and Prior - Unauthenticated Remote Code Execution Via AjaxServerSettingsChk.php (CVE-2019-16662)

Signature rule	CVE ID	Description
999793	CVE-2019-7609	WEB-MISC Elastic Kibana Prior to 5.6.15 and 6.6.1 - Prototype Pollution Vulnerability Allows Unauthenticated RCE (CVE-2019-7609)
999794	CVE-2019-10092	WEB-MISC Apache HTTP Server Up To 2.4.39 - mod_proxy Limited Cross-Site Scripting (CVE-2019-10092)
999795	CVE-2019-16520	WEB-WORDPRESS All In One SEO Pack plug-in Before 3.2.7 - Stored cross-site scripting Vulnerability (CVE-2019-16520)
999796	CVE-2019-17234	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up to 3.4.0 - Arbitrary File Deletion (CVE-2019-17234)
999797	CVE-2019-16525	WEB-WORDPRESS Checklist plug-in Prior to Version 1.1.9 - cross-site scripting Vulnerability (CVE-2019-16525)
999798		WEB-WORDPRESS Safe SVG plug-in Prior to 1.9.6 - cross-site scripting Vulnerability
999799		WEB-WORDPRESS Email Subscribers & Newsletters plug-in Prior to Version 4.2.3 - Unauthenticated Arbitrary Option Creation

Signature update version 40

September 14, 2021

New signatures rules are generated for the vulnerabilities identified for the week 2020-01-14. You can download and configure these signature rules to protect your appliance from security vulnerable attacks. The signature update includes the signature ID, signature version, and list of CVEs addressed.

Signature version

Signature version 40 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> The signature update version 40 includes a fix for the incorrect signature rule 1861.
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999732	CVE-2019-1620	WEB-MISC Cisco Data Center Network Manager Prior To 11.2(1) - Arbitrary File Upload Vulnerability (CVE-2019-1620)
999733	CVE-2019-16702	WEB-MISC Integard Pro 2.2.0.9026 - NoJs Buffer Overflow Vulnerability (CVE-2019-16702)
999734	CVE-2019-1621	WEB-MISC Cisco Data Center Network Manager Prior To 11.2(1) - Arbitrary File Download Vulnerability (CVE-2019-1621)
999735	CVE-2019-8451	WEB-MISC Atlassian Jira Server Before 8.4.0 - Server Side Request Forgery Vulnerability (CVE-2019-8451)

Signature rule	CVE ID	Description
999736		WEB-WORDPRESS GDPR Cookie Compliance plug-in Prior to 4.0.3 - Authenticated Arbitrary Settings Deletion Vulnerability
999737	CVE-2019-11287	WEB-MISC Pivotal RabbitMQ 3.7.x prior to 3.7.21 and 3.8.x prior to 3.8.1 - Denial of Service Vulnerability (CVE-2019-11287)
999738		WEB-WORDPRESS Ultimate Addons For Elementor Prior To 1.20.1 - Authentication Bypass Via Facebook Login Vulnerability
999739		WEB-WORDPRESS Ultimate Addons For Elementor Prior To 1.20.1 - Authentication Bypass Via Google Login Vulnerability
999740	CVE-2019-19366	WEB-MISC FusionPBX Prior to 4.4.10 - cross-site scripting Vulnerability in xml_cdr_search.php Via Redirect Parameter (CVE-2019-19366)
999741	CVE-2019-16931	WEB-WORDPRESS Visualizer plug-in Prior to Version 3.3.1 - Unauthenticated cross-site scripting Vulnerability (CVE-2019-16931)
999742	CVE-2019-16932	WEB-WORDPRESS Visualizer plug-in Prior to Version 3.3.1 - Unauthenticated SSRF (CVE-2019-16932)

Signature rule	CVE ID	Description
999743	CVE-2019-1619	WEB-MISC Cisco Data Center Network Manager Prior To 11.1(1) - Authentication Bypass Vulnerability (CVE-2019-1619)
999744	CVE-2019-12562	WEB-MISC DotNetNuke Before 9.4.0 - Stored Cross Site Scripting Vulnerability (CVE-2019-12562)
999745	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Filedata Field (CVE-2019-8371)
999746	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Image Field (CVE-2019-8371)
999747		WEB-WORDPRESS Beaver Builder Ultimate Addons Prior To 1.24.1 - Authentication Bypass Via Facebook Login Vulnerability
999748		WEB-WORDPRESS Beaver Builder Ultimate Addons Prior To 1.24.1 - Authentication Bypass Via Google Login Vulnerability
999749	CVE-2019-19650	WEB-MISC Zoho ManageEngine AM Prior to Build 13640 - SQLi Via Agent Servlet (CVE-2019-19650)

Signature rule	CVE ID	Description
999750		WEB-MISC Zoho ManageEngine AM Prior to Build 13620 - API Key Disclosure Via OPMRequestHandlerServlet Servlet
999751	CVE-2019-1622	WEB-MISC Cisco Data Center Network Manager 11.0(1) - Information Disclosure Vulnerability (CVE-2019-1622)
999752	CVE-2019-16759	WEB-MISC vBulletin Prior to 5.5.4 Patch Level 1 - Remote Code Execution Vulnerability (CVE-2019-16759)
999753		WEB-WORDPRESS Featured Image from URL plug-in Prior to 2.7.8 - Missing Access Controls on REST API Vulnerability
999754	CVE-2019-10098	WEB-MISC Apache HTTP Server Up To 2.4.39 - mod_rewrite Self-Referential Redirect Vulnerability (CVE-2019-10098)
999755	CVE-2019-1936	WEB-MISC Cisco UCS Director 6.0 to 6.6.1.0 and 6.7.0.0 to 6.7.1.0 - Command Injection Vulnerability (CVE-2019-1936)
999756	CVE-2019-19649	WEB-MISC Zoho ManageEngine AM Prior to Build 13620 - Unauthenticated SQLi Via EventID Parameter (CVE-2019-19649)

Signature rule	CVE ID	Description
999757	CVE-2019-19649	WEB-MISC Zoho ManageEngine AM Prior to Build 13620 - Unauthenticated SQLi Via Entity Parameter (CVE-2019-19649)
999758	CVE-2019-15036	WEB-MISC JetBrains TeamCity Before 2019.1 - OS Command Injection Vulnerability (CVE-2019-15036)
999759	CVE-2019-17239	WEB-WORDPRESS Download plug-ins and Themes from Dashboard plug-in Up To 1.5 - Stored cross-site scripting Vulnerability (CVE-2019-17239)

Signature update version 41

September 14, 2021

New signatures rules are generated for the vulnerabilities identified for the week 2020-02-04. You can download and configure these signature rules to protect your appliance from security vulnerable attacks. The signature update includes the signature ID, signature version, and list of CVEs addressed.

Signature version

Signature version 41 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> The signature update version 41 includes a fix for the incorrect signature rule 1861. Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999717		WEB-WORDPRESS WordPress Version 5.3.x and Prior - Denial of Service Vulnerability Via xmlrpc.php pingback.ping Method
999718		WEB-WORDPRESS Backup And Staging By WP Time Capsule plug-in Prior To 1.21.16 - Authentication Bypass Vulnerability
999719	CVE-2019-19731	WEB-MISC Roxy Fileman For .NET 1.4.5 - Path Traversal Vulnerability Via RENAMEFILE (CVE-2019-19731)
999720	CVE-2019-19915	WEB-WORDPRESS 301 Redirects – Easy Redirect Manager plug-in Up To 2.4.0 - Multiple Vulnerabilities (CVE-2019-19915)
999721	CVE-2019-17662	WEB-MISC Cybele Software ThinVNC Prior to Version 1.0b1 - Directory Traversal Vulnerability (CVE-2019-17662)
999722	CVE-2020-6168	WEB-WORDPRESS Minimal Coming Soon And Maintenance Mode plug-in Prior To 2.17 - Maintenance Setting Vulnerability (CVE-2020-6168)

Signature rule	CVE ID	Description
999723	CVE-2020-6166	WEB-WORDPRESS Minimal Coming Soon And Maintenance Mode plug-in Prior To 2.17 - Theme Change Vulnerability (CVE-2020-6166)
999724	CVE-2020-6166	WEB-WORDPRESS Minimal Coming Soon And Maintenance Mode plug-in Prior To 2.17 - Export Settings Vulnerability (CVE-2020-6166)
999725		WEB-WORDPRESS InifiniteWP Client plug-in Prior to 1.9.4.5 - Authentication Bypass Vulnerability
999726	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via REST API With JSON Object (CVE-2019-16773)
999727	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via REST API With FORM FIELD (CVE-2019-16773)
999728	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via user-edit.php (CVE-2019-16773)

Signature rule	CVE ID	Description
999729	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via profile.php (CVE-2019-16773)
999730	CVE-2019-16113	WEB-MISC Bludit 3.9.2 - Image Upload Remote Code Execution Vulnerability Via uuid (CVE-2019-16113)
999731	CVE-2019-16113	WEB-MISC Bludit 3.9.2 - Image Upload Remote Code Execution Vulnerability Via filename (CVE-2019-16113)

Signature update for February 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-02-11. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 42 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999707		WEB-WORDPRESS WPCentral plug-in Prior to Version 1.4.8 - Privilege Escalation Vulnerability
999708	CVE-2019-15979	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Command Injection Vulnerability (CVE-2019-15979)
999709	CVE-2019-15978	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Command Injection Vulnerability (CVE-2019-15978)
999710	CVE-2019-15975	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Authentication Bypass Vulnerability (CVE-2019-15975)
999711	CVE-2019-15976	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Authentication Bypass Vulnerability (CVE-2019-15976)
999712	CVE-2019-16405	WEB-MISC Centreon Prior to Version 19.10.2 - Remote Code Execution Vulnerability (CVE-2019-16405)
999713	CVE-2020-7048	WEB-WORDPRESS WP Database Reset plug-in Up To 3.1 - Unauthenticated DataBase Table Reset Vulnerability (CVE-2020-7048)

Signature rule	CVE ID	Description
999714	CVE-2020-7108	WEB-WORDPRESS LearnDash plug-in Prior to Version 3.1.2 - Reflected cross-site scripting Vulnerability (CVE-2020-7108)
999715	CVE-2019-15977	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Authentication Bypass Vulnerability (CVE-2019-15977)
999716	CVE-2020-2096	WEB-MISC Jenkins Gitlab Hook plug-in Version 1.4.2 and Prior - cross-site scripting Vulnerability (CVE-2020-2096)

Signature update for February 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-02-27. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 43 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999696	CVE-2019-15983	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - XML External Entity Vulnerability (CVE-2019-15983) Via CablePlans
999697	CVE-2019-20197	WEB-MISC Nagios XI 5.6.9 - Authenticated Arbitrary Command Execution Vulnerability (CVE-2019-20197)
999698	CVE-2020-8417	WEB-WORDPRESS Code Snippets plug-in Prior to 2.14.0 - CSRF Vulnerability (CVE-2020-8417)
999699		WEB-WORDPRESS WPCentral plug-in Prior to Version 1.4.8 - Privilege Escalation Vulnerability
999700	CVE-2020-8596	WEB-WORDPRESS Participants Database plug-in Prior To 1.9.5.6 - Authenticated SQL Injection Vulnerability (CVE-2020-8596)
999701	CVE-2020-8426	WEB-WORDPRESS Elementor Page Builder plug-in Prior To 2.8.5 - Authenticated Reflected cross-site scripting Vulnerability (CVE-2020-8426)
999702	CVE-2019-19509	WEB-MISC RConfig 3.9.3 - Remote Code Execution Vulnerability Via ajaxArchiveFiles.php (CVE-2019-19509)

Signature rule	CVE ID	Description
999703	CVE-2019-8449	WEB-MISC Atlassian Jira Server Before 8.4.0 - Information Disclosure Vulnerability (CVE-2019-8449)
999704	CVE-2019-9194	WEB-MISC elFinder Prior To 2.1.48 - PHP Connector Command Injection Vulnerability (CVE-2019-9194)
999705	CVE-2019-15985	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - SQL Injection Vulnerability (CVE-2019-15985) Via getVmHostData
999706	CVE-2020-8549	WEB-WORDPRESS Strong Testimonials plug-in Prior To 2.40.1 - Stored Cross Site Scripting Vulnerability (CVE-2020-8549)

Signature update for April 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-04-27. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 44 is applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999683	CVE-2020-9043	WEB-WORDPRESS wpCentral plug-in Prior To 1.5.1 - Connection Key Disclosure Vulnerability (CVE-2020-9043)
999684		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999685		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999686	CVE-2020-0618	WEB-MISC Microsoft SQL Server Reporting Services - Remote Code Execution Vulnerability (CVE-2020-0618)
999687	CVE-2019-16278	WEB-MISC Nostromo Nhttpd Prior to 1.3.7 - Strcutl Function Allows Unauthenticated Remote Code Execution (CVE-2019-16278)
999688	CVE-2019-1937	WEB-MISC Cisco UCS Director 6.6.0.0 to 6.6.1.0 and 6.7.0.0 to 6.7.1.0 - Authentication Bypass Vulnerability (CVE-2019-1937)
999689		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting

Signature rule	CVE ID	Description
999690	CVE-2020-9006	WEB-WORDPRESS Popup Builder plug-in Prior to 3.0 - SQL Injection Via PHP Deserialization Vulnerability (CVE-2020-9006)
999691		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999692		WEB-MISC prevent request smuggling via content-length and transfer-encoding header
999693		WEB-WORDPRESS ThemeGrill Demo Importer plug-in Prior To 1.6.3 - Authentication Bypass And Database Wipe Vulnerability
999694	CVE-2019-17237	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Prior to 3.4.1 - CSRF Vulnerability Via Message (CVE-2019-17237)
999695	CVE-2019-17237	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Prior to 3.4.1 - CSRF Vulnerability Via Subject (CVE-2019-17237)

Signature update for May 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-05-26. You can

download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 45 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU. According to the latest Snort release, the signature rules with ID 1258, 1306, 2520, 2661, 5695, 10996, 11817, 12056, 15471, 17049 and 21634 have been removed.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999666		WEB-WORDPRESS Duplicator plug-in Prior To 1.3.28 - Unauthenticated Arbitrary File Download Vulnerability
999667	CVE-2020-10220	WEB-MISC rConfig Through 3.94 - SQL Injection Vulnerability (CVE-2020-10220)
999668	CVE-2020-5844	WEB-MISC Artica Pandora FMS 7.0 - Execution of Arbitrary Files of Dangerous Type Via /attachment/files_repo/ (CVE-2020-5844)
999669	CVE-2020-8813	WEB-MISC Cacti Prior to 1.2.10 - Remote Code Execution Vulnerability Via graph_realtime.php (CVE-2020-8813)

Signature rule	CVE ID	Description
999670	CVE-2020-8654	WEB-MISC EyesOfNetwork 5.3 - Remote Code Execution Vulnerability (CVE-2020-8654)
999671	CVE-2020-10196	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Unauthenticated cross-site scripting Vulnerability (CVE-2020-10196)
999672	CVE-2019-15949	WEB-MISC Nagios XI Prior To 5.6.6 - Remote Code Execution As Root Vulnerability (CVE-2019-15949)
999673	CVE-2020-10879	WEB-MISC RConfig 3.9.5 and Prior - Remote Code Execution Vulnerability Via search.crud.php (CVE-2020-10879)
999674	CVE-2020-8656	WEB-MISC EyesOfNetwork 5.3 - EyesOfNetwork API 2.4.2 SQL Injection Vulnerability (CVE-2020-8656)
999675	CVE-2020-10195	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Authenticated System Information Disclosure (CVE-2020-10195)
999676	CVE-2020-10195	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Authenticated Subscriber Information Disclosure (CVE-2020-10195)

Signature rule	CVE ID	Description
999677	CVE-2020-10195	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Authenticated Settings Modification (CVE-2020-10195)
999678	CVE-2020-0646	Microsoft SharePoint Server - .NET Framework Workflow Remote Code Execution Vulnerability Via SOAP 1.2 (CVE-2020-0646)
999679	CVE-2020-0646	Microsoft SharePoint Server - .NET Framework Workflow Remote Code Execution Vulnerability Via SOAP 1.1 (CVE-2020-0646)
999680	CVE-2020-10221	WEB-MISC rConfig Through 3.94 - Remote Code Execution Vulnerability (CVE-2020-10221)
999681	CVE-2019-19134	WEB-WORDPRESS Hero Maps Premium Prior to 2.2.3 - Unauthenticated Reflected cross-site scripting Vulnerability (CVE-2019-19134)
999682	CVE-2020-10385	WEB-WORDPRESS WPForms plug-in Prior to 1.5.9 - Stored cross-site scripting Vulnerability (CVE-2020-10385)

Signature update for June 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-06-03. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 46 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999643		WEB-WORDPRESS 10Web Map Builder for Google Maps plug-in Prior to 10.0.64 - Unauthenticated cross-site scripting Vulnerability Via gmwd_setup Page
999644		WEB-WORDPRESS 10Web Map Builder for Google Maps plug-in 10.0.64 and Prior - cross-site scripting Vulnerability Via options_gmwd Page
999645	CVE-2020-5187	WEB-MISC DNN Up To 9.4.4 - Path Traversal Vulnerability Via URL (CVE-2020-5187)
999646	CVE-2020-5187	WEB-MISC DNN Up To 9.4.4 - Path Traversal Vulnerability Via Local (CVE-2020-5187)

Signature rule	CVE ID	Description
999647	CVE-2020-9335	WEB-WORDPRESS Photo Gallery plug-in Prior to 1.5.46 - cross-site scripting Vulnerability Via image_alt_text_ Field (CVE-2020-9335)
999648	CVE-2020-9335	WEB-WORDPRESS Photo Gallery plug-in Prior to 1.5.46 - cross-site scripting Vulnerability Via Name Field (CVE-2020-9335)
999649	CVE-2020-9335	WEB-WORDPRESS Photo Gallery plug-in Prior to 1.5.46 - cross-site scripting Vulnerability Via Description Fields (CVE-2020-9335)
999650	CVE-2020-10189	WEB-MISC Zoho ManageEngine Desktop Central Prior to 10.0.479 - Unauthenticated Remote Code Execution Vuln (CVE-2020-10189)
999651	CVE-2020-10189	WEB-MISC Zoho ManageEngine Desktop Central Prior to 10.0.479 - Unauthenticated Arbitrary File Upload Vuln (CVE-2020-10189)
999652		WEB-WORDPRESS Flexible Checkout Fields for WooCommerce plug-in Prior to 2.3.2 - Unauthenticated Settings Modification Vuln

Signature rule	CVE ID	Description
999653	CVE-2020-0688	WEB-MISC Microsoft Exchange Server - Validation Key Remote Code Execution Vulnerability (CVE-2020-0688)
999654	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via ip_src Parameter (CVE-2020-8947, CVE-2019-20224)
999655	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via dst_port Parameter (CVE-2020-8947, CVE-2019-20224)
999656	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via src_port Parameter (CVE-2020-8947, CVE-2019-20224)
999657	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via ip_dst Parameter (CVE-2020-8947, CVE-2019-20224)
999658	CVE-2020-5186	WEB-MISC DNN Up To 9.5.0 - Cross Site Scripting Vulnerability Via Journal XML Upload (CVE-2020-5186)

Signature rule	CVE ID	Description
999659		WEB-WORDPRESS WP Sitemap Page plug-in 1.6.2 and Prior - cross-site scripting Vulnerability Via wsp_exclude_pages
999660	CVE-2020-5188	WEB-MISC DNN Up To 9.5.0 - Insecure Permissions Vulnerability Via UploadFromUrl (CVE-2020-5188)
999661	CVE-2020-5188	WEB-MISC DNN Up To 9.5.0 - Insecure Permissions Vulnerability Via UploadFromLocal (CVE-2020-5188)
999662	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via API Theme (CVE-2020-7799)
999663	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via API Email Template (CVE-2020-7799)
999664	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via GUI Theme (CVE-2020-7799)
999665	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via GUI Email Template (CVE-2020-7799)

Signature update for June 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-06-12. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 47 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999580	CVE-2020-6010	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - SQL Injection Vulnerability (CVE-2020-6010)
999581		WEB-MISC Nagios XI Up To 5.6.13 - Service Command_Test Arbitrary Command Execution Vulnerability
999582	CVE-2020-0932	Microsoft SharePoint Server - WebPart Source Markup Remote Code Execution Vulnerability Via SOAP 1.2 (CVE-2020-0932)

Signature rule	CVE ID	Description
999583	CVE-2020-0932	Microsoft SharePoint Server - WebPart Source Markup Remote Code Execution Vulnerability Via SOAP 1.1 (CVE-2020-0932)
999584	CVE-2020-12642	WEB-WORDPRESS Ninja Forms plug-in Prior to 3.4.24.2 - Cross-Site Request Forgery Vulnerability via Import Fields (CVE-2020-12642)
999585	CVE-2020-12642	WEB-WORDPRESS Ninja Forms plug-in Prior to 3.4.24.2 - Cross-Site Request Forgery Vulnerability via Import Form (CVE-2020-12642)
999586	CVE-2020-11450	WEB-MISC Microstrategy Web 10.4 - Information Disclosure Vulnerability (CVE-2020-11450)
999587	CVE-2020-7935	WEB-MISC Artica Pandora FMS 7.0 - Unrestricted Upload of File With Dangerous Type Vulnerability Allows RCE (CVE-2020-7935)
999588	CVE-2020-12116	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125125 - Information Disclosure Vulnerability (CVE-2020-12116)
999589		WEB-WORDPRESS Elementor Page Builder Prior to 2.9.6 - Privilege Escalation Vulnerability

Signature rule	CVE ID	Description
999590	CVE-2020-11738	WEB-WORDPRESS - Snap Creek Duplicator plug-in Prior to 1.3.28 - Path Traversal Vulnerability (CVE-2020-11738)
999591	CVE-2020-10389	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Remote Code Execution vulnerability (CVE-2020-10389)
999592	CVE-2020-11516	WEB-WORDPRESS Contact Form 7 Datepicker plug-in Up To 2.6.0 - Stored cross-site scripting Vulnerability (CVE-2020-11516)
999593		WEB-MISC Nagios XI Up To 5.6.13 - Export-RRD Arbitrary Command Execution Vulnerability Via Step
999594		WEB-MISC Nagios XI Up To 5.6.13 - Export-RRD Arbitrary Command Execution Vulnerability Via End
999595		WEB-MISC Nagios XI Up To 5.6.13 - Export-RRD Arbitrary Command Execution Vulnerability Via Start
999596	CVE-2019-19799	Zoho ManageEngine Applications Manager Previous To 14600 - Information Disclosure Vulnerability (CVE-2019-19799)

Signature rule	CVE ID	Description
999597	CVE-2020-10458	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Arbitrary Folder Deletion Vulnerability (CVE-2020-10458)
999598	CVE-2017-9822	WEB-MISC DNN Before 9.1.1 - Remote Code Execution Vulnerability Via DNNPersonalization Cookie (CVE-2017-9822)
999599	CVE-2020-7953	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Information Disclosure Vulnerability Via nmap_options Param (CVE-2020-7953)
999600	CVE-2020-7953	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Information Disclosure Vulnerability Via host Param (CVE-2020-7953)
999601		WEB-MISC Bolt CMS 3.7.0 - File Rename to a Dangerous Type Vulnerability Via newname Parameter
999602		WEB-MISC Bolt CMS 3.7.0 - Path Traversal Vulnerability Via newname Parameter
999603		WEB-MISC Bolt CMS 3.7.0 - Path Traversal Vulnerability Via oldname Parameter
999604		WEB-MISC Bolt CMS 3.7.0 - Path Traversal Vulnerability Via parent Parameter

Signature rule	CVE ID	Description
999605		WEB-MISC Bolt CMS 3.7.0 - Improper Field Validation Vulnerability in displayname Parameter
999606	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in View Logs (CVE-2020-9004)
999607	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in Media Cache Settings (CVE-2020-9004)
999608	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in Applications Settings (CVE-2020-9004)
999609	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in Server Settings (CVE-2020-9004)
999610		WEB-MISC PrestaShop 1.7.6.5 - CSRF Vulnerability via Filemanager
999611	CVE-2020-10238	WEB-MISC Joomla! Previous To 3.9.16 - Security Bypass Vulnerability via com_templates (CVE-2020-10238)
999612	CVE-2020-11510	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - Privilege Escalation Via learnpress_create_page (CVE-2020-11510)

Signature rule	CVE ID	Description
999613	CVE-2020-11510	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - Privilege Escalation Via learn-press_update_order_status (CVE-2020-11510)
999614	CVE-2020-8636	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Remote Code Execution Vulnerability Via nmap_options Parameter (CVE-2020-8636)
999615	CVE-2020-8636	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Remote Code Execution Vulnerability Via host Parameter (CVE-2020-8636)
999616	CVE-2020-11511	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - Privilege Escalation Via accept-to-be-teacher (CVE-2020-11511)
999617	CVE-2020-11451	WEB-MISC Microstrategy Web - Unsecure File Type Upload Vulnerability Via JSP (CVE-2020-11451)
999618	CVE-2020-11451	WEB-MISC Microstrategy Web - Unsecure File Type Upload Vulnerability Via ASP (CVE-2020-11451)
999619	CVE-2020-11515	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Redirection Vulnerability Via REST API Through URL (CVE-2020-11515)

Signature rule	CVE ID	Description
999620	CVE-2020-11515	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Redirection Vulnerability Via REST API rest_route Param (CVE-2020-11515)
999621	CVE-2020-10457	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Arbitrary File Renaming Vulnerability Via imgName (CVE-2020-10457)
999622	CVE-2020-10457	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Arbitrary File Renaming Vulnerability Via imgUrl (CVE-2020-10457)
999623	CVE-2019-1821	WEB-MISC Cisco Prime Infrastructure - Remote Code Execution Vulnerability (CVE-2019-1821)
999624		WEB-WORDPRESS Page Builder plug-in Prior to 2.10.16 - CSRF Vulnerability Via Ajax action_builder_content
999625		WEB-WORDPRESS Page Builder plug-in Prior to 2.10.16 - CSRF Vulnerability Via Live Editor
999626	CVE-2020-11514	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Privilege Escalation Via REST API Through URL (CVE-2020-11514)

Signature rule	CVE ID	Description
999627	CVE-2020-11514	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Privilege Escalation Via REST API rest_route Param (CVE-2020-11514)
999628	CVE-2019-6713	WEB-MISC ThinkCMF Prior to 5.0.190312 - Code Injection Vulnerability Via /route/editpost.html (CVE-2019-6713)
999629	CVE-2019-6713	WEB-MISC ThinkCMF Prior to 5.0.190312 - Code Injection Vulnerability Via /route/addpost.html (CVE-2019-6713)
999630		WEB-WORDPRESS Google Site Kit plug-in Prior to 1.8.0 - Unprotected Verification Vulnerability
999631	CVE-2020-9315	WEB-MISC Oracle iPlanet Web Server 7.0.x - Incorrect Access Control Vulnerability (CVE-2020-9315)
999632	CVE-2020-1947	WEB-MISC Apache ShardingSphere 4.0.0-RC3 and 4.0.0 - SnakeYAML Remote Code Execution Vulnerability (CVE-2020-1947)
999633	CVE-2020-7961	Liferay Portal Prior To 7.2.1 CE GA2 - JSONWS Deserialization RCE Vulnerability Via JSON-RPC (CVE-2020-7961)
999634	CVE-2020-7961	Liferay Portal Prior To 7.2.1 CE GA2 - JSONWS Deserialization RCE Vulnerability Via URL Path (CVE-2020-7961)

Signature rule	CVE ID	Description
999635	CVE-2020-7961	Liferay Portal Prior To 7.2.1 CE GA2 - JSONWS Deserialization RCE Vulnerability Via Form And URI Query (CVE-2020-7961)
999636	CVE-2020-8518	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Remote Code Execution Vulnerability (CVE-2020-8518)
999637	CVE-2020-7351	WEB-MISC Fonality Trixbox CE 2.8.0.4 and Prior - Remote Code Execution Vulnerability (CVE-2020-7351)
999638	CVE-2020-12720	WEB-MISC vBulletin Prior to 5.6.1 Patch Level 1 - Unauthenticated SQL Injection Vulnerability (CVE-2020-12720)
999639	CVE-2019-19800	Zoho ManageEngine Applications Manager Previous To 14520 - Path Traversal Vulnerability (CVE-2019-19800)
999640	CVE-2020-10386	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Remote Code Execution (CVE-2020-10386)
999641	CVE-2020-8497	WEB-MISC Artica Pandora FMS 7.0 - Unauthenticated Information Disclosure Vulnerability (CVE-2020-8497)

Signature rule	CVE ID	Description
999642	CVE-2020-6009	WEB-WORDPRESS LearnDash LMS plug-in Prior to 3.1.6 - Unauthenticated SQL Injection Vulnerability (CVE-2020-6009)

Signature update for July 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-07-01. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 48 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999563		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - cross-site scripting Vulnerability Via pagelayer_cf_to_email

Signature rule	CVE ID	Description
999564		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - cross-site scripting Vulnerability Via pagelayer-phone
999565		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - cross-site scripting Vulnerability Via pagelayer-address
999566	CVE-2020-1961	WEB-MISC Apache Syncope - Server-Side Template Injection Vulnerability (CVE-2020-1961)
999567	CVE-2019-18935	WEB-MISC Progress Telerik UI For ASP.NET AJAX - RadAsyncUpload .NET Deserialization Vulnerability (CVE-2019-18935)
999568	CVE-2020-9463	WEB-MISC Centreon 19.10 - OS Command Injection Vulnerability (CVE-2020-9463)
999569		WEB-WORDPRESS Support Review plug-in Prior to 3.7.6 - Unauthenticated Stored Cross Site Scripting Vulnerability
999570		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - Improper Access Control Vuln Via pagelayer_save_template
999571		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - Improper Access Control Vuln Via pagelayer_update_site_title

Signature rule	CVE ID	Description
999572		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - Improper Access Control Vuln Via pagelayer_save_content
999573		WEB-WORDPRESS Drag And Drop Upload For Contact Form 7 Prior To 1.3.3.3 - Arbitrary File Extension Upload Vulnerability
999574	CVE-2020-9314	WEB-MISC Oracle iPlanet Web Server 7.0.x - Image Injection Vulnerability (CVE-2020-9314)
999575	CVE-2020-9484	WEB-MISC Apache Tomcat Multiple Versions - Deserialization of Untrusted Data (CVE-2020-9484)
999576	CVE-2020-13252	WEB-MISC Centreon Prior to 19.04.15 - Remote Code Execution Vulnerability (CVE-2020-13252)
999577	CVE-2020-11453	WEB-MISC Microstrategy Web - CSRF Vulnerability Via SOAP (CVE-2020-11453)
999578	CVE-2020-11453	WEB-MISC Microstrategy Web - CSRF Vulnerability (CVE-2020-11453)
999579	CVE-2020-7237	WEB-MISC Cacti Prior to 1.2.8 - Remote Code Execution Vulnerability (CVE-2020-7237)

Signature update for August 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-08-26. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 49 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999556	CVE-2020-13241	WEB-MISC Microweber 1.1.18 - Unrestricted Upload of File with Dangerous Type Vulnerability (CVE-2020-13241)
999557	CVE-2020-3250	WEB-MISC Cisco UCS Director - REST API Path Traversal Vulnerability Via userAPIDownloadFile (CVE-2020-3250)
999558		WEB-WORDPRESS PageBuilder KingComposer plug-in Prior to 2.9.4 - Arbitrary Deletion of Directories Via action=bulk-delete
999559		WEB-WORDPRESS PageBuilder KingComposer plug-in Prior to 2.9.4 - Remote Code Execution Vulnerability Via action=upload

Signature rule	CVE ID	Description
999560	CVE-2018-1999024	WEB-MISC Moodle - MathJax Unicode cross-site scripting Vulnerability (CVE-2018-1999024)
999561	CVE-2020-13693	WEB-WORDPRESS bbPress plug-in Prior To 2.6.5 - Unauthenticated Privilege Escalation Vulnerability (CVE-2020-13693)
999562	CVE-2020-12847	WEB-MISC Pydio Cells Prior to 2.0.7 - Remote Code Execution Vulnerability (CVE-2020-12847)

Signature update for September 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-09-26. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 50 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999532	CVE-2020-1956	WEB-MISC Apache Kylin - Cube Migrate Remote Code Execution Via dest-config (CVE-2020-1956)
999533	CVE-2020-1956	WEB-MISC Apache Kylin - Cube Migrate Remote Code Execution Via src-config (CVE-2020-1956)
999534	CVE-2020-1956	WEB-MISC Apache Kylin - Cube Migrate Remote Code Execution Via projectName (CVE-2020-1956)
999535	CVE-2020-3247	WEB-MISC Cisco UCS Director - CopyFileRunnable Arbitrary Symlink Creation Vulnerability (CVE-2020-3247)
999536	CVE-2019-16872	WEB-MISC Portainer Prior To 1.22.1 - Incorrect Access Control Vulnerability Via Update Stacks (CVE-2019-16872)
999537	CVE-2019-16872	WEB-MISC Portainer Prior To 1.22.1 - Incorrect Access Control Vulnerability Via Create Stacks (CVE-2019-16872)
999538	CVE-2020-13855	WEB-MISC Artica Pandora FMS 7.44 - Arbitrary File Upload Vulnerability Via File Repository Manager (CVE-2020-13855)
999539	CVE-2020-5902	WEB-MISC F5 BIG-IP - Traffic Management User Interface RCE Vulnerability Via /hsqldb (CVE-2020-5902)

Signature rule	CVE ID	Description
999540	CVE-2020-5902	WEB-MISC F5 BIG-IP - Traffic Management User Interface RCE Vulnerability Via /tmui (CVE-2020-5902)
999541		WEB-MISC WebERP 4.15.1 and Prior - Unauthenticated Information Disclosure Vulnerability
999542	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via timeline.php and timestamp Param (CVE-2020-7209)
999543	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and ts Param (CVE-2020-7209)
999544	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and end Param (CVE-2020-7209)
999545	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and start Param (CVE-2020-7209)
999546	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and pid Param (CVE-2020-7209)
999547	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kidsk_trace_view.php and end Param (CVE-2020-7209)

Signature rule	CVE ID	Description
999548	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kidsk_trace_view.php and start Param (CVE-2020-7209)
999549		WEB-MISC PHP-Fusion Prior to 9.03.70 - PHP Object Injection Vulnerability
999550	CVE-2020-1181	WEB-MISC Microsoft SharePoint Server - Remote Code Execution via Web Parts (CVE-2020-1181)
999551	CVE-2020-10547	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Policy Elements Via searchColumn (CVE-2020-10547)
999552	CVE-2020-10547	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Policy Elements Via searchField (CVE-2020-10547)
999553	CVE-2020-8605	WEB-MISC Trend Micro InterScan Web Security Virtual Appliance Prior to 6.5 SP2 Patch 4 - RCE Vulnerability (CVE-2020-8605)
999554	CVE-2019-10068	WEB-MISC Kentico CMS Multiple Versions - Unauthenticated Remote Code Execution Vulnerability (CVE-2019-10068)

Signature rule	CVE ID	Description
999555	CVE-2020-11108	WEB-MISC Pi-hole Up To 4.4 - Authenticated RCE Vulnerability (CVE-2020-11108)

Signature update for Oct 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-10-13. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 51 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999505		WEB-WORDPRESS WordPress plug-in wpDiscuz 7.0.0 Up To 7.0.4 - Unauthenticated Arbitrary File Upload Vulnerability
999506		WEB-WORDPRESS Quiz & Survey Master - cross-site scripting Vulnerability in Questions Feature

Signature rule	CVE ID	Description
999507	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /log_search and cf Param (CVE-2020-8604)
999508	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /collection and cf Param (CVE-2020-8604)
999509	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /log_search and File Param (CVE-2020-8604)
999510	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /collection and File Param (CVE-2020-8604)
999511	CVE-2020-7361	WEB-MISC ZenTao Enterprise 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Edit (CVE-2020-7361)
999512	CVE-2020-7361	WEB-MISC ZenTao Pro 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Edit (CVE-2020-7361)
999513	CVE-2020-7361	WEB-MISC ZenTao Enterprise 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Create (CVE-2020-7361)
999514	CVE-2020-7361	WEB-MISC ZenTao Pro 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Create (CVE-2020-7361)

Signature rule	CVE ID	Description
999515	CVE-2020-5768	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.5.1 - SQL Injection Vulnerability (CVE-2020-5768)
999516	CVE-2020-5767	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.5.1 - CSRF Vulnerability (CVE-2020-5767)
999517	CVE-2020-15299	WEB-WORDPRESS KingComposer plug-in Prior To 2.9.5 - cross-site scripting Vulnerability (CVE-2020-15299)
999518	CVE-2020-13854	WEB-MISC Artica Pandora FMS - Privilege Escalation Vulnerability (CVE-2020-13854)
999519	CVE-2020-13852	WEB-MISC Artica Pandora FMS - Arbitrary File Upload Vulnerability Via File Manager (CVE-2020-13852)
999520	CVE-2020-13700	WEB-WORDPRESS WordPress plug-in acf-to-rest-api Before 3.3.0 - Information Disclosure Vulnerability Via URI (CVE-2020-13700)
999521	CVE-2020-13700	WEB-WORDPRESS WordPress plug-in acf-to-rest-api Before 3.3.0 - Information Disclosure Vulnerability Via URL (CVE-2020-13700)

Signature rule	CVE ID	Description
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 Through 7.0.1 - CSRF Bypass Leading To DOS Vulnerability (CVE-2020-13379)
999523	CVE-2020-12851	WEB-MISC Pydio Cells Prior to 2.0.7 - Arbitrary File Write Vulnerability (CVE-2020-12851)
999524	CVE-2020-12848	WEB-MISC Pydio Cells Prior to 2.0.7 - Login as Temporary Shared User Vulnerability (CVE-2020-12848)
999525	CVE-2020-11749	WEB-MISC Artica Pandora FMS Prior To 7.47 - cross-site scripting Vulnerability Via SNMP Browser (CVE-2020-11749)
999526	CVE-2020-11579	WEB-MISC PHPKBV9 - File Exfiltration Vulnerability (CVE-2020-11579)
999527	CVE-2020-10546	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Compliance Policies Via searchColumn (CVE-2020-10546)
999528	CVE-2020-10546	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Compliance Policies Via searchField (CVE-2020-10546)
999529	CVE-2019-16876	WEB-MISC Portainer Prior To 1.22.1 - Directory Traversal Vulnerability (CVE-2019-16876)

Signature rule	CVE ID	Description
999530		WEB-WORDPRESS - ADning plug-in Prior to 1.5.6 - Unauthenticated Arbitrary File Deletion Vulnerability
999531		WEB-WORDPRESS - ADning plug-in Prior to 1.5.6 - Unauthenticated Arbitrary File Upload Vulnerability

Signature update for October 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-10-29. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 52 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU. Also, vulnerable versions are mentioned in some of the signature rule log string. You must enable it accordingly.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999500	CVE-2018-14667	WEB-MISC RichFaces Framework 3.X Through 3.3.4 - EL Injection Via UserResource (CVE-2018-14667)
999501	CVE-2018-12533	WEB-MISC RichFaces Framework 3.1.0 Through 3.3.4 - EL Injection Via Paint2DResource (CVE-2018-12533)
999502	CVE-2015-0279, CVE-2018-12532	WEB-MISC RichFaces Framework 4.X Through 4.5.17 - EL Injection Via MediaOutputResource (CVE-2015-0279,CVE-2018-12532)
999503	CVE-2013-2165	WEB-MISC RichFaces v4 Prior to 4.3.3 - Java Object Deserialization Vulnerability (CVE-2013-2165)
999504	CVE-2013-2165	WEB-MISC RichFaces v3 Prior to 3.3.4 - Java Object Deserialization Vulnerability (CVE-2013-2165)

Signature update for November 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-11-10. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 53 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999411		WEB-WORDPRESS WordPress plug-in wpDiscuz 7.0.0 Up To 7.0.4 - Unauthenticated Arbitrary File Upload Vulnerability
999412		WEB-WORDPRESS Quiz & Survey Master - cross-site scripting Vulnerability in Questions Feature
999413		WEB-WORDPRESS WordPress plug-in File Manager Prior To 6.9 - Unauthenticated elFinder Commands Execution Vulnerability
999414	CVE-2020-11700	WEB-MISC Titan SpamTitan Prior To 7.08 - Information Disclosure Vulnerability (CVE-2020-11700)
999415	CVE-2020-9446	WEB-MISC Apache OFBiz 17.12.03 - XML-RPC Unsafe Deserialization Vulnerability (CVE-2020-9446)
999416	CVE-2020-9446	WEB-MISC Apache OFBiz 17.12.03 - XML-RPC Cross-Site Scripting Vulnerability (CVE-2020-9446)

Signature rule	CVE ID	Description
999417	CVE-2020-9047	WEB-MISC exacqVision Web Service Up To 20.06.3.0 - OS Command Injection Vulnerability (CVE-2020-9047)
999418	CVE-2020-8866	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Unrestricted Upload of File Vulnerability Via edit.php (CVE-2020-8866)
999419	CVE-2020-8866	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Unrestricted Upload of File Vulnerability Via add.php (CVE-2020-8866)
999420	CVE-2020-8865	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Arbitrary File Inclusion Vulnerability Via edit.php (CVE-2020-8865)
999421	CVE-2020-8816	WEB-MISC Pi-hole Prior To 4.3.2 - Remote Code Execution Vulnerability Via removestatic (CVE-2020-8816)
999422	CVE-2020-8816	WEB-MISC Pi-hole Prior To 4.3.2 - Remote Code Execution Vulnerability Via AddMAC (CVE-2020-8816)
999423	CVE-2020-8243	WEB-MISC Pulse Connect Secure Prior To 9.1R8.2 - Remote Code Execution Vulnerability (CVE-2020-8243)
999424	CVE-2020-8218	WEB-MISC Pulse Connect Secure Prior To 9.1R8 - Remote Code Execution Vulnerability (CVE-2020-8218)

Signature rule	CVE ID	Description
999425	CVE-2020-6143, CVE-2020-6144	WEB-MISC OS4Ed OpenSIS - Code Injection Vulnerability Via /install/Ins1.php (CVE-2020-6143, CVE-2020-6144)
999426	CVE-2020-6142	WEB-MISC OS4Ed OpenSIS - Path Traversal Vulnerability Via modname (CVE-2020-6142)
999427	CVE-2020-6141	WEB-MISC OS4Ed OpenSIS Prior to 7.4 - Unauthenticated SQLi Vulnerability Via USERNAME (CVE-2020-6141)
999428	CVE-2020-6140	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via username_stn_id (CVE-2020-6140)
999429	CVE-2020-6139	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via username_stf_email (CVE-2020-6139)
999430	CVE-2020-6138	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via uname (CVE-2020-6138)
999431	CVE-2020-6137	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via password_stf_email (CVE-2020-6137)

Signature rule	CVE ID	Description
999432	CVE-2020-6125	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via GetSchool.php and u Parameter (CVE-2020-6125)
999433	CVE-2020-6124	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via EmailCheckOthers.php (CVE-2020-6124)
999434	CVE-2020-6123	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via EmailCheck.php and p_id Parameter (CVE-2020-6123)
999435	CVE-2020-6123	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via EmailCheck.php and email Parameter (CVE-2020-6123)
999436	CVE-2020-6122	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and mn Parameter (CVE-2020-6122)
999437	CVE-2020-6121	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and ln Parameter (CVE-2020-6121)

Signature rule	CVE ID	Description
999438	CVE-2020-6120	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and fn Parameter (CVE-2020-6120)
999439	CVE-2020-6119	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and byear Parameter (CVE-2020-6119)
999440	CVE-2020-6118	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and bmonth Parameter (CVE-2020-6118)
999441	CVE-2020-6117	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and bday Parameter (CVE-2020-6117)
999442	CVE-2020-5780	WEB-WORDPRESS WordPress plug-in Email Subscribers And Newsletters Prior To 4.5.6 - Email Forgery Vulnerability (CVE-2020-5780)
999443	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via JSON-RPC (CVE-2020-4280)

Signature rule	CVE ID	Description
999444	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteMethod (CVE-2020-4280)
999445	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteJavaScript (CVE-2020-4280)
999446	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via JSON-RPC (CVE-2020-4280)
999447	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteMethod (CVE-2020-4280)
999448	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteJavaScript (CVE-2020-4280)
999449	CVE-2020-24786	WEB-MISC Zoho ManageEngine ADManager Plus 7.0 Prior to Build 55 - Improper Authentication Vulnerability (CVE-2020-24786)
999450	CVE-2020-24389	WEB-WORDPRESS Drag and Drop Multiple File Uploader plug-in Prior To 1.3.5.5 - Security Bypass Vulnerability (CVE-2020-24389)

Signature rule	CVE ID	Description
999451	CVE-2020-24046	WEB-MISC TitanHQ SpamTitan Gateway 7.08 - Privilege Escalation Vulnerability (CVE-2020-24046)
999452	CVE-2020-17506	WEB-MISC Artica Web Proxy 4.30.000000 - PreAuth SQL Injection Vulnerability Via Apikey Parameter (CVE-2020-17506)
999453	CVE-2020-17505	WEB-MISC Artica Web Proxy 4.30.000000 - OS Command Injection Vulnerability Via Service-cmds-peform Parameter (CVE-2020-17505)
999454	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/users/items (CVE-2020-17463)
999455	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/sitevariables/items (CVE-2020-17463)
999456	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/permissions/items (CVE-2020-17463)
999457	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/pages/items (CVE-2020-17463)
999458	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/navigation/items (CVE-2020-17463)

Signature rule	CVE ID	Description
999459	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/logs/items (CVE-2020-17463)
999460	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/blocks/items (CVE-2020-17463)
999461	CVE-2020-16875	WEB-MISC Microsoft Exchange Server - DLP Policy Remote Code Execution Vulnerability (CVE-2020-16875)
999462	CVE-2020-16171	WEB-MISC Acronis Cyber Backup Prior To 12.5 Build 16342 - SSRF Via Shard Header Vulnerability (CVE-2020-16171)
999463	CVE-2020-14947	WEB-MISC OCS Inventory Prior to 2.8 - OS Command Injection Vulnerability Via SNMP_MIB_DIRECTORY (CVE-2020-14947)
999464	CVE-2020-14947	WEB-MISC OCS Inventory Prior to 2.8 - OS Command Injection Vulnerability Via mib_file (CVE-2020-14947)
999465	CVE-2020-14008	WEB-MISC Zoho ManageEngine Applications Manager Up To 14710 - Remote Code Execution Vulnerability (CVE-2020-14008)
999466	CVE-2020-13925	WEB-MISC Apache Kylin Prior To 3.1.0 - Remote Code Execution Vulnerability Via Job (CVE-2020-13925)

Signature rule	CVE ID	Description
999467	CVE-2020-13925	WEB-MISC Apache Kylin Prior To 3.1.0 - Remote Code Execution Vulnerability Via Project (CVE-2020-13925)
999468	CVE-2020-13854	WEB-MISC Artica Pandora FMS - Privilege Escalation Vulnerability (CVE-2020-13854)
999469	CVE-2020-13405	WEB-MISC Microweber Prior to 1.1.20 - Unauthenticated Information Disclosure Vulnerability (CVE-2020-13405)
999470	CVE-2020-13376	WEB-MISC SecurEnvoy SecurMail 9.3.503 - SecurEnvoyReply Cookie Path Traversal Vulnerability (CVE-2020-13376)
999471	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via domain (CVE-2020-13159)
999472	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via netbiosname (CVE-2020-13159)
999473	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via alias (CVE-2020-13159)

Signature rule	CVE ID	Description
999474	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via hostname (CVE-2020-13159)
999475	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via dhclient_server (CVE-2020-13159)
999476	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via dhclient_interface (CVE-2020-13159)
999477	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via dhclient_mac (CVE-2020-13159)
999478	CVE-2020-13158	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - Path Traversal Vulnerability Via popup (CVE-2020-13158)
999479	CVE-2020-12851	WEB-MISC Pydio Cells Prior to 2.0.7 - Arbitrary File Write Vulnerability (CVE-2020-12851)
999480	CVE-2020-12848	WEB-MISC Pydio Cells Prior to 2.0.7 - Login as Temporary Shared User Vulnerability (CVE-2020-12848)

Signature rule	CVE ID	Description
999481	CVE-2020-11699	WEB-MISC Titan SpamTitan Prior To 7.08 - Remote Code Execution Vulnerability (CVE-2020-11699)
999482	CVE-2020-11579	WEB-MISC PHPKBV9 - File Exfiltration Vulnerability (CVE-2020-11579)
999483	CVE-2020-10818	WEB-MISC Artica Web Proxy 4.26 - OS Command Injection Vulnerability Via fw.system.info.php (CVE-2020-10818)
999484	CVE-2020-10228	WEB-MISC Vtenext CE Prior to Version 20 - Unrestricted Upload of File with Dangerous Type Vulnerability (CVE-2020-10228)
999485	CVE-2020-10204	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via coreui_User roles (CVE-2020-10204)
999486	CVE-2020-10204	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via coreui_Role privileges (CVE-2020-10204)
999487	CVE-2020-10204	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via coreui_Role roles (CVE-2020-10204)

Signature rule	CVE ID	Description
999488	CVE-2020-10199	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via REST Endpoint /bower/group (CVE-2020-10199)
999489	CVE-2020-10199	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via REST Endpoint /go/group (CVE-2020-10199)
999490	CVE-2020-10199	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via REST Endpoint /docker/group (CVE-2020-10199)
999491	CVE-2019-19699	WEB-MISC Centreon Up To 19.10 - Remote Code Execution Vulnerability (CVE-2019-19699)
999492	CVE-2019-19499	WEB-MISC Apache Grafana Up To 6.4.3 - Arbitrary File Read Vulnerability (CVE-2019-19499)
999493	CVE-2019-18394	WEB-MISC Ignite Realtime Openfire Up To 4.4.2 - FaviconServlet Server Side Request Forgery Vulnerability (CVE-2019-18394)
999494	CVE-2019-18393	WEB-MISC Ignite Realtime Openfire Up To 4.4.2 - plug-inServlet Directory Traversal Vulnerability (CVE-2019-18393)

Signature rule	CVE ID	Description
999495	CVE-2019-16759	WEB-MISC vBulletin Prior to 5.6.2 - Remote Code Execution Vulnerability Via Nested Template (CVE-2019-16759)
999496	CVE-2019-15715	WEB-MISC MantisBT Prior to 1.3.20 and 2.22.1 - Remote Code Execution Vulnerability Via neato_tool (CVE-2019-15715)
999497	CVE-2019-15715	WEB-MISC MantisBT Prior to 1.3.20 and 2.22.1 - Remote Code Execution Vulnerability Via dot_tool (CVE-2019-15715)
999498	CVE-2019-11043	WEB-MISC PHP-FPM Multiple Versions - Out-Of-Bounds Write Vulnerability Allows Arbitrary Code Execution (CVE-2019-11043)
999499		WEB-WORDPRESS WordPress plug-in Autoptimize Up To 2.7.6 - Authenticated Arbitrary File Upload Vulnerability

Signature update for December 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-12-02. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 54 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU. As part of signature update version 54, log string for signature 999720 is changed to ensure that it includes only ASCI characters.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999394	CVE-2020-8255	WEB-MISC Pulse Connect Secure Prior To 9.1R9 - Information Disclosure Vulnerability (CVE-2020-8255)
999395	CVE-2020-6128	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CoursePeriodModal.php (CVE-2020-6128)
999396	CVE-2020-6126, CVE-2020-6127	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CoursePeriodModal.php (CVE-2020-6126, CVE-2020-6127)
999397	CVE-2020-28328	WEB-MISC SuiteCRM Prior to 7.11.16 - Remote Code Execution Vulnerability (CVE-2020-28328)

Signature rule	CVE ID	Description
999398	CVE-2020-27995	WEB-MISC Zoho ManageEngine Applications Manager 14 Prior to Build 14560 - SQL Injection Vulnerability (CVE-2020-27995)
999399	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /service/ (CVE-2020-26879)
999400	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /reboot (CVE-2020-26879)
999401	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /patch/ (CVE-2020-26879)
999402	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /upgrade/ (CVE-2020-26879)
999403	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /module/ (CVE-2020-26879)
999404	CVE-2020-26878	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Arbitrary OS Command Injection Vulnerability (CVE-2020-26878)

Signature rule	CVE ID	Description
999405	CVE-2020-25790	WEB-MISC Typesetter CMS 5.x Through 5.1 - Unsecure File Upload Vulnerability (CVE-2020-25790)
999406	CVE-2020-25540	WEB-MISC ThinkAdmin v6 - Directory Traversal Vulnerability (CVE-2020-25540)
999407	CVE-2020-14883	WEB-MISC Oracle WebLogic Server - Authenticated Remote Code Execution Vulnerability (CVE-2020-14883)
999408	CVE-2020-14882, CVE-2020-14750	WEB-MISC Oracle WebLogic Server - Authentication Bypass Vulnerability (CVE-2020-14882, CVE-2020-14750)
999409	CVE-2020-11975, CVE-2020-13942	WEB-MISC Apache Unomi Prior to 1.5.2 - Remote Code Execution Vulnerability (CVE-2020-11975, CVE-2020-13942)
999410	CVE-2020-11803	WEB-MISC Titan SpamTitan Prior To 7.08 - Remote Code Execution Vulnerability (CVE-2020-11803)

Signature update for December 2020

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2020-12-17. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 55 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999377		WEB-WORDPRESS TI WooCommerce Wishlist Plugin Prior To 1.21.11 - Information Disclosure Vulnerability Via tinvwl_export_settings
999378		WEB-WORDPRESS TI WooCommerce Wishlist Plugin Prior To 1.21.11 - WP Options Change Vulnerability Via tinvwl_import_settings
999379	CVE-2020-6134	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via MassDropModal.php (CVE-2020-6134)
999380	CVE-2020-6133	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CourseMoreInfo.php (CVE-2020-6133)
999381	CVE-2020-6132	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via ChooseCP.php (CVE-2020-6132)

Signature rule	CVE ID	Description
999382	CVE-2020-6131	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via MassScheduleSessionSet.php (CVE-2020-6131)
999383	CVE-2020-6130	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via MassDropSessionSet.php (CVE-2020-6130)
999384	CVE-2020-6129	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CpSessionSet.php (CVE-2020-6129)
999385	CVE-2020-35234	WEB-WORDPRES Easy WP SMTP Plugin Prior to 1.4.4 - Information Disclosure Vulnerability (CVE-2020-35234)
999386	CVE-2020-25042	WEB-MISC Mara CMS 7.5 - Arbitrary File Upload Vulnerability (CVE-2020-25042)
999387	CVE-2020-13526	WEB-MISC ProcessMaker - SQL Injection Vulnerability Via clientSetupAjax (CVE-2020-13526)
999388	CVE-2020-13525	WEB-MISC ProcessMaker - SQL Injection Vulnerability Via reportTables_Ajax (CVE-2020-13525)

Signature rule	CVE ID	Description
999389	CVE-2020-12147	WEB-MISC Silver Peak Unity Orchestrator - Arbitrary MySQL Queries Vulnerability Via sqlExecution REST API (CVE-2020-12147)
999390	CVE-2020-12146	WEB-MISC Silver Peak Unity Orchestrator - Path Traversal Vulnerability Via debugFiles REST API (CVE-2020-12146)
999391	CVE-2020-12145	WEB-MISC Silver Peak Unity Orchestrator - Authentication Bypass Vulnerability (CVE-2020-12145)
999392	CVE-2019-8394	WEB-MISC Zoho ManageEngine ServiceDesk Plus Prior to 10.0 Build 10012 - Arbitrary File Upload Vulnerability (CVE-2019-8394)
999393	CVE-2019-11447	WEB-MISC CutePHP CuteNews 2.1.2 - Remote Code Execution Vulnerability (CVE-2019-11447)

Signature update for January 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-01-18. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 56 applicable for NetScaler VPX 11.1, NetScaler 12.0,Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999366	CVE-2020-8466	WEB-MISC Trend Micro IWSSVA 6.5 SP2 Prior to Build 1919 - Unauthenticated OS Command Injection Vulnerability (CVE-2020-8466)
999367	CVE-2020-6135	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via Validator.php (CVE-2020-6135)
999368	CVE-2020-4001	WEB-MISC VMWare SD-WAN Orchestrator - Pass-the-Hash Vulnerability (CVE-2020-4001)
999369	CVE-2020-4000	WEB-MISC VMWare SD-WAN Orchestrator - Path Traversal Vulnerability (CVE-2020-4000)
999370	CVE-2020-3984	WEB-MISC VMWare SD-WAN Orchestrator - SQL Injection Vulnerability Via Modulus (CVE-2020-3984)
999371	CVE-2020-35606	WEB-MISC Webmin Up to 1.962 - Remote Code Execution Vulnerability (CVE-2020-35606)
999372	CVE-2020-17143	WEB-MISC Microsoft Exchange Server - Information Disclosure Vulnerability (CVE-2020-17143)

Signature rule	CVE ID	Description
999373	CVE-2020-17141	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability Via RouteComplaint (CVE-2020-17141)
999374	CVE-2020-10816	WEB-MISC Zoho ManageEngine Applications Manager 14 Prior to Build 14790 - Improper Authentication Vulnerability (CVE-2020-10816)
999375	CVE-2019-5533	WEB-MISC VMWare SD-WAN Orchestrator - Information Disclosure Vulnerability (CVE-2019-5533)
999376	CVE-2018-15961	WEB-MISC Adobe ColdFusion 12 Prior to Update 6 or 14 - Arbitrary File Upload Vulnerability (CVE-2018-15961)

Signature update for February 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-02-03. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 57 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999339		WEB-MISC Zoom Meeting Connector 4.6.348.20201217 - Remote Code Execution Vulnerability Via proxyPasswd
999340		WEB-MISC Zoom Meeting Connector 4.6.348.20201217 - Remote Code Execution Vulnerability Via proxyName
999341	CVE-2021-3129	WEB-MISC Ignition Prior to 2.5.2 - Unauthenticated Remote Code Execution Vulnerability (CVE-2021-3129)
999342	CVE-2021-3025	WEB-MISC Invision Community IPS Community Suite Prior to 4.5.4.2 - SQL Injection Vulnerability Via sortDir (CVE-2021-3025)
999343	CVE-2021-2109	WEB-MISC Oracle WebLogic Server - Remote Code Execution Vulnerability Via JNDI Injection (CVE-2021-2109)
999344	CVE-2020-7200	WEB-MISC HPE Systems Insight Manager 7.6.x - AMF Unsecure Deserialization Vulnerability (CVE-2020-7200)

Signature rule	CVE ID	Description
999345	CVE-2020-7199	WEB-MISC HPE EIM Prior to 1.21 - Improper Authentication Vulnerability in /private/EIMApplianceIP (CVE-2020-7199)
999346	CVE-2020-7199	WEB-MISC HPE EIM Prior to 1.21 - Improper Authentication Vulnerability in /private/AdminPassReset (CVE-2020-7199)
999347	CVE-2020-7199	WEB-MISC HPE EIM Prior to 1.21 - Improper Authentication Vulnerability in /private/ResetAppliance (CVE-2020-7199)
999348	CVE-2020-6136	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via DownloadWindow.php (CVE-2020-6136)
999349	CVE-2020-35729	WEB-MISC KLog Server 2.4.1 and Prior - OS Command Injection Vulnerability (CVE-2020-35729)
999350	CVE-2020-35701	WEB-MISC Cacti 1.2.16 and Prior - SQL Injection Vulnerability Via site_id (CVE-2020-35701)
999351	CVE-2020-35489	WEB-WORDPRESS Contact Form 7 Prior to 5.3.2 - Unrestricted File Upload Vulnerability (CVE-2020-35489)

Signature rule	CVE ID	Description
999352	CVE-2020-27615	WEB-WORDPRESS Loginizer Plugin Prior to 1.6.4 - SQL Injection Vulnerability (CVE-2020-27615)
999353	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/sitevariables/create (CVE-2020-26046)
999354	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/sitevariables/edit (CVE-2020-26046)
999355	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/navigation/create (CVE-2020-26046)
999356	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/navigation/edit (CVE-2020-26046)
999357	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/blocks/create (CVE-2020-26046)
999358	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/blocks/edit (CVE-2020-26046)
999359	CVE-2020-26045	WEB-MISC Fuel CMS 1.4.11 - SQLi Vulnerability Via /fuel/permissions/create (CVE-2020-26045)
999360	CVE-2020-17519	WEB-MISC Apache Flink Prior to 1.11.3 - Arbitrary File Disclosure Vulnerability (CVE-2020-17519)

Signature rule	CVE ID	Description
999361	CVE-2020-17518	WEB-MISC Apache Flink 1.5.1 to 1.11.2 - Arbitrary Location File Upload Vulnerability (CVE-2020-17518)
999362	CVE-2019-16010	WEB-MISC Cisco SD-WAN vManage Prior to 19.2.2 - Stored XSS Vulnerability (CVE-2019-16010)
999363	CVE-2019-15000	WEB-MISC VMWare Bitbucket Server and Data Center - Git Command Injection Vulnerability Via at (CVE-2019-15000)
999364	CVE-2019-15000	WEB-MISC VMWare Bitbucket Server and Data Center - Git Command Injection Vulnerability Via until/untilID (CVE-2019-15000)
999365	CVE-2019-15000	WEB-MISC VMWare Bitbucket Server and Data Center - Git Command Injection Vulnerability Via since/sinceID (CVE-2019-15000)

Signature update for February 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-02-17. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 58 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999328	CVE-2021-3317	WEB-MISC KLog Server 2.4.1 and Prior - OS Command Injection Vulnerability (CVE-2021-3317)
999329	CVE-2021-3110	WEB-MISC PrestaShop Prior to 1.7.7.1 - SQL Injection Vulnerability Via id_products (CVE-2021-3110)
999330	CVE-2021-3110	WEB-MISC PrestaShop Prior to 1.7.7.1 - SQL Injection Vulnerability Via /module/productcomments/-CommentGrade (CVE-2021-3110)
999331	CVE-2021-25646	WEB-MISC Apache Druid Prior to 0.20.1 - Remote Code Execution Vulnerability (CVE-2021-25646)
999332	CVE-2020-36171	WEB-WORDPRESS Elementor Page Builder Plugin Prior to 3.0.14 - XSS Vulnerability (CVE-2020-36171)

Signature rule	CVE ID	Description
999333	CVE-2020-35765	WEB-MISC Zoho ManageEngine Applications Manager Prior to Build 15000 - SQL Injection Vulnerability (CVE-2020-35765)
999334	CVE-2020-35589	WEB-WORDPRESS Limit Login Attempts Reloaded Prior to 2.15.2 - Reflected Cross-Site Scripting Vulnerability (CVE-2020-35589)
999335	CVE-2020-26282	WEB-MISC BrowserUp Proxy Prior to 2.1.2 - Template Injection Leading To RCE Vulnerability Via mostRecentEntry (CVE-2020-26282)
999336	CVE-2020-26282	WEB-MISC BrowserUp Proxy Prior to 2.1.2 - Template Injection Leading To RCE Vulnerability Via entries (CVE-2020-26282)
999337	CVE-2020-14815	WEB-MISC Oracle Business Intelligence Enterprise Edition - Reflected Cross-Site Scripting Vulnerability (CVE-2020-14815)
999338		WEB-WORDPRESS Contact Form 7 Database Addon Prior to 1.2.5.4 - SQLi Vulnerability Via Delete Bulk Action

Signature update for March 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-03-08. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 59 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999313	CVE-2021-25299	WEB-MISC NagiosXI Up to 5.7.5 - XSS Vulnerability via url (CVE-2021-25299)
999314	CVE-2021-25298	WEB-MISC NagiosXI Up to 5.7.5 - Remote Code Execution Vulnerability via DigitalOcean Wizard (CVE-2021-25298)
999315	CVE-2021-25297	WEB-MISC NagiosXI Up to 5.7.5 - Remote Code Execution Vulnerability via Switch Wizard (CVE-2021-25297)
999316	CVE-2021-25296	WEB-MISC NagiosXI Up to 5.7.5 - Remote Code Execution Vulnerability via WindowsWMI Wizard (CVE-2021-25296)

Signature rule	CVE ID	Description
999317	CVE-2021-24164	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34.1 - Information Disclosure Vulnerability (CVE-2021-24164)
999318	CVE-2021-24163	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34 - Authorization Bypass Vulnerability (CVE-2021-24163)
999319	CVE-2021-21972	WEB-MISC VMWare vCenter Server Plugin - Remote Code Execution Vulnerability (CVE-2021-21972)
999320	CVE-2020-35129	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via New Social Monitoring Form (CVE-2020-35129)
999321	CVE-2020-35129	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via Edit Social Monitoring Form (CVE-2020-35129)
999322	CVE-2020-35128	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via New Companies Form (CVE-2020-35128)
999323	CVE-2020-35128	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via Edit Companies Form (CVE-2020-35128)
999324	CVE-2020-35125	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via Referer Header (CVE-2020-35125)

Signature rule	CVE ID	Description
999325	CVE-2020-35125	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via mauticform[return] (CVE-2020-35125)
999326	CVE-2020-13933	WEB-MISC Apache Shiro Prior to 1.6.0 - Authentication Bypass Vulnerability Via Semicolon (CVE-2020-13933)
999327	CVE-2020-13921, CVE-2020-9483	WEB-MISC Apache SkyWalking Prior to 8.4.0 - SQL Injection Vulnerability Via queryLogs Feature (CVE-2020-13921, CVE-2020-9483)

Signature update for March 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-03-09. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 60 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999311	CVE-2021-26855	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability Via X-AnonResource-Backend (CVE-2021-26855)
999312	CVE-2021-26855	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability Via X-BEResource (CVE-2021-26855)

Signature update for March 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-03-11. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 61 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999308	CVE-2021-21302	WEB-MISC PrestaShop Prior to 1.7.7.2 - CSV Injection Vulnerability (CVE-2021-21302)
999309	CVE-2020-35749	WEB-WORDPRESS Simple Job Board Prior to 2.9.4 - Arbitrary File Disclosure Vulnerability (CVE-2020-35749)
999310	CVE-2019-16012	WEB-MISC Cisco SD-WAN vManage Prior to 19.2.2 - SQL Injection Vulnerability (CVE-2019-16012)

Signature update for March 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-03-11. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 62 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999307	CVE-2021-27065	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability (CVE-2021-27065)

Signature update for April 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-04-08. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 63 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999294	CVE-2021-3273	WEB-MISC NagiosXI Prior to 5.7 - Code Injection Vulnerability (CVE-2021-3273)
999295	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ssh_priv (CVE-2021-3197)

Signature rule	CVE ID	Description
999296	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ssh_port (CVE-2021-3197)
999297	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ssh_options (CVE-2021-3197)
999298	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ProxyCommand in JSON Object (CVE-2021-3197)
999299	CVE-2021-25282	WEB-MISC SaltStack Prior to 3002.3 - Path Traversal Vulnerability Via pillar_roots.write (CVE-2021-25282)
999300	CVE-2021-24166	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34 - CSRF Vulnerability (CVE-2021-24166)
999301	CVE-2021-24085	WEB-MISC Microsoft Exchange Server - Spoofing Vulnerability (CVE-2021-24085)
999302	CVE-2021-22986	WEB-MISC F5 iControl REST API - Remote Code Execution Vulnerability (CVE-2021-22986)
999303	CVE-2021-21978	WEB-MISC VMWare View Planner Harness 4.x prior to 4.6 Security Patch 1 - Remote Code Execution Vulnerability (CVE-2021-21978)

Signature rule	CVE ID	Description
999304	CVE-2020-23132	WEB-MISC Joomla! Prior to 3.9.25 - Unsafe com_media Upload Path Vulnerability Via file_path (CVE-2020-23132)
999305	CVE-2020-23132	WEB-MISC Joomla! Prior to 3.9.25 - Unsafe com_media Upload Path Vulnerability Via image_path (CVE-2020-23132)
999306	CVE-2020-22425	WEB-MISC Centreon Prior to 20.10.4 - SQL Injection Vulnerability (CVE-2020-22425)

Signature update for April 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-04-22. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 64 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999275	CVE-2021-3378	WEB-MISC FortiLogger 4.4.2.2 - Unauthenticated Arbitrary File Upload Vulnerability (CVE-2021-3378)
999276	CVE-2021-28925	WEB-MISC Nagios Network Analyzer Prior to 2.4.3 - SQL Injection Vulnerability (CVE-2021-28925)
999277	CVE-2021-28924	WEB-MISC Nagios Network Analyzer Prior to 2.4.3 - XSS Vulnerability (CVE-2021-28924)
999278	CVE-2021-27927	WEB-MISC Zabbix - CSRF Vulnerability Via action=authentication.update (CVE-2021-27927)
999279	CVE-2021-26295	WEB-MISC Apache OFBiz 17.12.06 - Unauthenticated Arbitrary Deserialization Vulnerability (CVE-2021-26295)
999280	CVE-2021-25770	WEB-MISC JetBrains YouTrack Prior to 2020.5.3123 - Server-Side Template Injection Vulnerability (CVE-2021-25770)
999281	CVE-2021-25283	WEB-MISC SaltStack Prior to 3002.5 - Remote Code Execution Vulnerability (CVE-2021-25283)
999282	CVE-2021-25283	WEB-MISC SaltStack Prior to 3002.5 - Remote Code Execution Vulnerability Via JSON Object (CVE-2021-25283)

Signature rule	CVE ID	Description
999283	CVE-2021-24218	WEB-WORDPRESS Facebook for WordPress Plugin Prior to 3.0.4 - Stored Cross-Site Scripting Vulnerability (CVE-2021-24218)
999284	CVE-2021-24217	WEB-WORDPRESS Facebook for WordPress Plugin Prior to 3.0.2 - PHP Object Injection Vulnerability (CVE-2021-24217)
999285	CVE-2021-24209	WEB-WORDPRESS WP Super Cache Plugin Prior to 1.7.2 - Remote Code Execution Vulnerability in wp-cache-config.php (CVE-2021-24209)
999286	CVE-2021-24209	WEB-WORDPRESS WP Super Cache Plugin Prior to 1.7.2 - Arbitrary Code Injection Vulnerability (CVE-2021-24209)
999287	CVE-2021-24165	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34 - Open Redirect Vulnerability (CVE-2021-24165)
999288	CVE-2021-21975	WEB-MISC vRealize Operations Manager - Unauthenticated Server Side Request Forgery Vulnerability (CVE-2021-21975)
999289	CVE-2020-35578	WEB-MISC Nagios XI Prior to 5.8.0 - Remote Code Execution Vulnerability (CVE-2020-35578)

Signature rule	CVE ID	Description
999290	CVE-2020-2766	WEB-MISC Oracle WebLogic Server - Unauthenticated SSRF Vulnerability (CVE-2020-2766)
999291	CVE-2020-17523	WEB-MISC Apache Shiro Prior to 1.7.1 - Authentication Bypass Vulnerability Via Space (CVE-2020-17523)
999292	CVE-2020-17523	WEB-MISC Apache Shiro Prior to 1.7.1 - Authentication Bypass Vulnerability Via Dot (CVE-2020-17523)
999293	CVE-2020-15160	WEB-MISC PrestaShop Prior to 1.7.6.8 - SQL Injection Vulnerability (CVE-2020-15160)

Signature update for June 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-06-02. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 65 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999243	CVE-2021-31761	WEB-MISC Webmin Prior to 1.974 - XSS Vulnerability Via /servers/link.cgi/ (CVE-2021-31761)
999244	CVE-2021-31761	WEB-MISC Webmin Prior to 1.974 - XSS Vulnerability Via /tunnel/link.cgi/ (CVE-2021-31761)
999245	CVE-2021-31166	WEB-IIS Microsoft HTTP Protocol Stack - Remote Code Execution Vulnerability (CVE-2021-31166)
999246	CVE-2021-29447	WEB-WORDPRESS WordPress Prior to 5.7.1 - Media Library XXE Vulnerability (CVE-2021-29447)
999247	CVE-2021-28157	WEB-MISC Devolutions Server Prior to 2021.1 and 2020.3.18 - SQL Injection Vulnerability Via User Delete (CVE-2021-28157)
999248	CVE-2021-27905	WEB-MISC Apache Solr Prior to 8.2.2 - ReplicationHandler SSRF Vulnerability via leaderUrl (CVE-2021-27905)
999249	CVE-2021-27905	WEB-MISC Apache Solr Prior to 8.2.2 - ReplicationHandler SSRF Vulnerability via masterUrl (CVE-2021-27905)
999250	CVE-2021-27890	WEB-MISC MyBB Prior to 1.8.26 - Theme Properties SQL Injection Vulnerability (CVE-2021-27890)

Signature rule	CVE ID	Description
999251	CVE-2021-27850, CVE-2019-0195	WEB-MISC Apache Tapestry - Unauthenticated Information Disclosure Vulnerability (CVE-2021-27850 and CVE-2019-0195)
999252	CVE-2021-27183	WEB-MISC MDaemon Prior to 20.0.4 - Arbitrary File Write Vulnerability (CVE-2021-27183)
999253	CVE-2021-27181	WEB-MISC MDaemon Prior to 20.0.4 - Anti-CSRF Token Fixation Vulnerability (CVE-2021-27181)
999254	CVE-2021-27180	WEB-MISC MDaemon Prior to 20.0.4 - Reflected XSS Vulnerability (CVE-2021-27180)
999255	CVE-2021-24340	WEB-WORDPRESS WP Statistics Prior to 13.0.8 - Unauthenticated SQL Injection Vulnerability (CVE-2021-24340)
999256	CVE-2021-24171	WEB-WORDPRESS WooCommerce Upload Files Plugin Prior to 59.4 - Path Traversal Vulnerability (CVE-2021-24171)
999257	CVE-2021-24171	WEB-WORDPRESS WooCommerce Upload Files Plugin Prior to 59.4 - Arbitrary File Upload Vulnerability (CVE-2021-24171)

Signature rule	CVE ID	Description
999258	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via UserServlet and user_password (CVE-2021-22658)
999259	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via UserServlet and user_name (CVE-2021-22658)
999260	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via CommandServlet and user_password (CVE-2021-22658)
999261	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via CommandServlet and user_name (CVE-2021-22658)
999262	CVE-2021-21983	WEB-MISC VMWare vRealize Operations Manager Prior to 8.4 - Arbitrary File Write Vulnerability (CVE-2021-21983)
999263	CVE-2020-6754	WEB-MISC dotCMS Prior to 5.2.4 - Directory Traversal Vulnerability Via assets (CVE-2020-6754)
999264	CVE-2020-27128	WEB-MISC Cisco SD-WAN vManage Prior to 20.3.1 - Arbitrary File Write Vulnerability Via remoteprocessing (CVE-2020-27128)

Signature rule	CVE ID	Description
999265	CVE-2020-27128	WEB-MISC Cisco SD-WAN vManage Prior to 20.3.1 - Arbitrary File Write Vulnerability Via dr (CVE-2020-27128)
999266	CVE-2020-15714	WEB-MISC rConfig 3.9.5 and Prior - SQL Injection Vulnerability (CVE-2020-15714)
999267	CVE-2020-15713	WEB-MISC rConfig Prior to 3.9.6 - SQL Injection Vulnerability (CVE-2020-15713)
999268	CVE-2020-14295	WEB-MISC Cacti Prior to 1.2.13 - SQL Injection Vulnerability (CVE-2020-14295)
999269	CVE-2020-13778	WEB-MISC rConfig Prior to 3.9.5 - Remote Code Execution Vulnerability Via ajaxEditTemplate.php (CVE-2020-13778)
999270	CVE-2020-13778	WEB-MISC rConfig Prior to 3.9.5 - Remote Code Execution Vulnerability Via ajaxAddTemplate.php (CVE-2020-13778)
999271	CVE-2020-13592	WEB-MISC Rukovoditel Project Management App - SQL Injection Vulnerability Via selected_fields (CVE-2020-13592)
999272	CVE-2020-13592	WEB-MISC Rukovoditel Project Management App - SQL Injection Vulnerability Via lists_id (CVE-2020-13592)

Signature rule	CVE ID	Description
999273	CVE-2020-13591	WEB-MISC Rukovoditel Project Management App - SQL Injection Vulnerability (CVE-2020-13591)
999274	CVE-2020-13550	WEB-MISC Advantech WebAccess/SCADA - Path Traversal Vulnerability Via fileName (CVE-2020-13550)

Signature update for July 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-07-08. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 66 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999231	CVE-2021-34074	WEB-MISC Artica Pandora FMS Up to 7.54 - Arbitrary File Upload Vulnerability Via Relative Path (CVE-2021-34074)

Signature rule	CVE ID	Description
999232	CVE-2021-32633	WEB-MISC Plone CMS - Zope Page Templates Remote Code Execution Vulnerability Via Upload (CVE-2021-32633)
999233	CVE-2021-32633	WEB-MISC Plone CMS - Zope Page Templates Remote Code Execution Vulnerability Via New (CVE-2021-32633)
999234	CVE-2021-31181	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability (CVE-2021-31181)
999235	CVE-2021-24370	WEB-WORDPRESS Fancy Product Designer Plugin Prior to 5.6.9 - RCE Vulnerability Via fpd_custom_uplod_file (CVE-2021-24370)
999236	CVE-2021-24370	WEB-WORDPRESS Fancy Product Designer Plugin Prior to 5.6.9 - RCE Vulnerability Via custom-image-handler.php (CVE-2021-24370)
999237	CVE-2021-24354	WEB-WORDPRESS Simple 301 Redirects Plugin Prior to 2.0.4 - Arbitrary Plugin Installation Vulnerability (CVE-2021-24354)
999238	CVE-2021-24352	WEB-WORDPRESS Simple 301 Redirects Plugin Prior to 2.0.4 - Redirect Export Vulnerability (CVE-2021-24352)
999239	CVE-2021-1497, CVE-2021-1498	WEB-MISC Cisco HyperFlex HX Prior to 4.0(2e) - Remote Code Execution Vulnerability (CVE-2021-1497, CVE-2021-1498)

Signature rule	CVE ID	Description
999240	CVE-2020-21057	WEB-MISC FusionPBX 4.5.7 - Path Traversal Vulnerability Via folderdelete Feature (CVE-2020-21057)
999241	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via backupDatabase (CVE-2020-16245)
999242	CVE-2020-10148	WEB-MISC SolarWinds Orion Multiple Versions - Authentication Bypass Vulnerability (CVE-2020-10148)

Signature update for August 2021

September 14, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-08-29. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 67 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999183	CVE-2021-37557	WEB-MISC Centreon Multiple Versions - SQL Injection Vulnerability (CVE-2021-37557)
999184	CVE-2021-35501	WEB-MISC Artica Pandora FMS Up to 7.54 - Visual Console Stored XSS Vulnerability (CVE-2021-35501)
999185	CVE-2021-35464	WEB-MISC ForgeRock Access Management and OpenAM - Remote Code Execution Vulnerability (CVE-2021-35464)
999186	CVE-2021-34523	WEB-MISC Microsoft Exchange Server - Elevation of Privilege Vulnerability (CVE-2021-34523)
999187	CVE-2021-34473	WEB-MISC Microsoft Exchange Server - Server Side Request Forgery Authentication Bypass Vulnerability Via Query (CVE-2021-34473)
999188	CVE-2021-34473	WEB-MISC Microsoft Exchange Server - Server Side Request Forgery Authentication Bypass Vulnerability Via Cookie (CVE-2021-34473)
999189	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability via Absolute Path (CVE-2021-33203)

Signature rule	CVE ID	Description
999190	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability via Path Traversal (CVE-2021-33203)
999191	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability via backslash (CVE-2021-33203)
999192	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability Via Slash (CVE-2021-33203)
999193	CVE-2021-3287, CVE-2020-28653	WEB-MISC Zoho ManageEngine OpManager Prior to 12.5.329 - Unauthenticated RCE Vulnerability (CVE-2021-3287, CVE-2020-28653)
999194	CVE-2021-32789	WEB-WORDPRESS WooCommerce Plugin Up to 5.5.0 - SQL Injection Vulnerability Via taxonomy and rest_route (CVE-2021-32789)
999195	CVE-2021-32789	WEB-WORDPRESS WooCommerce Plugin Up to 5.5.0 - SQL Injection Vulnerability Via taxonomy (CVE-2021-32789)

Signature rule	CVE ID	Description
999196	CVE-2021-32604	WEB-MISC SolarWinds Serv-U Prior to 15.2.3 - Cross-Site Scripting Vulnerability Via SenderEmail Parameter (CVE-2021-32604)
999197	CVE-2021-32093	WEB-MISC National Security Agency Emissary 5.9.0 - Arbitrary File Read Vulnerability (CVE-2021-32093)
999198	CVE-2021-31760	WEB-MISC Webmin Prior to 1.974 - CSRF Vulnerability Lead to RCE Via run.cgi (CVE-2021-31760)
999199	CVE-2021-31207	WEB-MISC Microsoft Exchange Server - Security Feature Bypass Vulnerability (CVE-2021-31207)
999200	CVE-2021-31195	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vunerability (CVE-2021-31195)
999201	CVE-2021-28474	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability (CVE-2021-28474)
999202	CVE-2021-24385	WEB-WORDPRESS FileBird Plugin 4.7.3 - SQL Injection Vulnerability Via selectedFolder Parameter and rest_route (CVE-2021-24385)
999203	CVE-2021-24385	WEB-WORDPRESS FileBird Plugin 4.7.3 - SQL Injection Vulnerability Via selectedFolder Parameter (CVE-2021-24385)

Signature rule	CVE ID	Description
999204	CVE-2021-24385	WEB-WORDPRESS FileBird Plugin 4.7.3 - SQL Injection Vulnerability Via JSON-Encoded Body (CVE-2021-24385)
999205	CVE-2021-24356	WEB-WORDPRESS Simple 301 Redirects Plugin Prior to 2.0.4 - Arbitrary Plugin Activation Vulnerability (CVE-2021-24356)
999206	CVE-2021-23024	WEB-MISC F5 BIG-IQ Multiple Versions - Remote Code Execution Vulnerability (CVE-2021-23024)
999207	CVE-2021-22911	WEB-MISC Rocket.Chat Server 3.11, 3.12 and 3.13 - Blind NOSQL Injection Vulnerability (CVE-2021-22911)
999208	CVE-2021-22900	WEB-MISC Pulse Connect Secure Prior To 9.1R11.4 - Remote Code Execution Vulnerability Via smimeCert.cgi (CVE-2021-22900)
999209	CVE-2021-22900	WEB-MISC Pulse Connect Secure Prior To 9.1R11.4 - Remote Code Execution Vulnerability Via admincert.cgi (CVE-2021-22900)
999210	CVE-2021-22900	WEB-MISC Pulse Connect Secure Prior To 9.1R11.4 - Remote Code Execution Vulnerability Via clientauthcert.cgi (CVE-2021-22900)

Signature rule	CVE ID	Description
999211	CVE-2021-22160	WEB-MISC Apache Pulsar - JSON Web Tokens Authentication Bypass Vulnerability (CVE-2021-22160)
999212	CVE-2021-21809	WEB-MISC Moodle - Remote Code Execution Vulnerability Via Spellchecker Plugin and getSuggestions Method (CVE-2021-21809)
999213	CVE-2021-21809	WEB-MISC Moodle - Remote Code Execution Vulnerability Via Spellchecker Plugin and checkWords Method (CVE-2021-21809)
999214	CVE-2021-21809	WEB-MISC Moodle - Remote Code Execution Vulnerability Via s__aspellpath (CVE-2021-21809)
999215	CVE-2021-21805	WEB-MISC Advantech R-SeeNet - Unauthenticated Remote Code Execution Vulnerability (CVE-2021-21805)
999216	CVE-2021-21804	WEB-MISC Advantech R-SeeNet - Local File Inclusion Vulnerability Via sub_opt (CVE-2021-21804)
999217	CVE-2021-21587	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/os/listfiles (CVE-2021-21587)

Signature rule	CVE ID	Description
999218	CVE-2021-21587	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/app/rsp/listfiles (CVE-2021-21587)
999219	CVE-2021-21586	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/app and fileName (CVE-2021-21586)
999220	CVE-2021-21586	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/os and fileName (CVE-2021-21586)
999221	CVE-2021-21586	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/os and filePath (CVE-2021-21586)
999222	CVE-2020-25223	WEB-MISC Sophos SG UTM - Remote Code Execution Via SID and /var (CVE-2020-25223)
999223	CVE-2020-25223	WEB-MISC Sophos SG UTM - Remote Code Execution Via SID and /webadmin.plx (CVE-2020-25223)
999224	CVE-2020-21056	WEB-MISC FusionPBX 4.5.7 - Path Traversal Vulnerability Via foldernew (CVE-2020-21056)

Signature rule	CVE ID	Description
999225	CVE-2020-21055	WEB-MISC FusionPBX 4.5.7 - Path Traversal Vulnerability Via File Rename Feature (CVE-2020-21055)
999226	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability in findSummaryUpdateDeviceListExpo (CVE-2020-16245)
999227	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via findCfgDeviceListExport (CVE-2020-16245)
999228	CVE-2020-14181	WEB-MISC Atlassian Jira Server - Information Disclosure Vulnerability Via ViewUserHover.jspa (CVE-2020-14181)
999229	CVE-2020-14005	WEB-MISC SolarWinds Orion Prior to 2020.2.1 HF 2 - Remote Code Execution Via ExecuteVBScript Action Type (CVE-2020-14005)
999230	CVE-2020-14005	WEB-MISC SolarWinds Orion Prior to 2020.2.1 HF 2 - Remote Code Execution Via ExecuteExternalProgram Action Type (CVE-2020-14005)

Signature update for September 2021

September 20, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-09-11. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 68 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 and Citrix ADC 13.1 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999163	CVE-2021-37556	WEB-MISC Centreon Multiple Versions - SQL Injection Vulnerability Via End Parameter (CVE-2021-37556)
999164	CVE-2021-37556	WEB-MISC Centreon Multiple Versions - SQL Injection Vulnerability Via Start Parameter (CVE-2021-37556)
999165	CVE-2021-37353	WEB-MISC Nagios XI Docker Wizard Prior to 1.1.3 - SSRF Vulnerability Via host Parameter Without URI Scheme (CVE-2021-37353)
999166	CVE-2021-37353	WEB-MISC Nagios XI Docker Wizard Prior to 1.1.3 - SSRF Vulnerability Via host Parameter With URI Scheme (CVE-2021-37353)

Signature rule	CVE ID	Description
999167	CVE-2021-34638	WEB-WORDPRESS Download Manager Plugin Prior to 3.1.25 - Directory Traversal Vulnerability (CVE-2021-34638)
999168	CVE-2021-33766	WEB-MISC Microsoft Exchange Server - Information Disclosure Vulnerability (CVE-2021-33766)
999169	CVE-2021-32682	WEB-MISC elFinder Prior To 2.1.59 - Command Injection Vulnerability Via Archive (CVE-2021-32682)
999170	CVE-2021-26084	WEB-MISC Confluence Server and Data Center - OGNL Injection Vulnerability Via doenterpagevariables (CVE-2021-26084)
999171	CVE-2021-26084	WEB-MISC Confluence Server and Data Center - OGNL Injection Vulnerability Via createpage-entervariables (CVE-2021-26084)
999172	CVE-2021-23394	WEB-MISC elFinder Prior To 2.1.59 - Remote Code Execution Vulnerability Via Phar Makefile (CVE-2021-23394)
999173	CVE-2021-23394	WEB-MISC elFinder Prior To 2.1.59 - Remote Code Execution Vulnerability Via Phar Rename (CVE-2021-23394)

Signature rule	CVE ID	Description
999174	CVE-2021-23394	WEB-MISC elFinder Prior To 2.1.59 - Remote Code Execution Vulnerability Via Phar Upload (CVE-2021-23394)
999175	CVE-2020-36289	WEB-MISC Atlassian Jira Server - Information Disclosure Vulnerability Via QueryComponentRenderer-Value (CVE-2020-36289)
999176	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via findSummaryCfgDeviceListExport (CVE-2020-16245)
999177	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via findUpdateDeviceListExport (CVE-2020-16245)
999178	CVE-2020-13774	WEB-MISC Ivanti Endpoint Manager Multiple Versions - RCE Vulnerability Via EditLaunchPadDialog.aspx (CVE-2020-13774)
999179	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability Via Custom Page (CVE-2020-1147)
999180	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability Via quicklinksdialogform.aspx (CVE-2020-1147)

Signature rule	CVE ID	Description
999181	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability Via quicklinks.aspx (CVE-2020-1147)
999182	CVE-2020-11110	WEB-MISC Apache Grafana Up to 6.7.1 - XSS Vulnerability (CVE-2020-11110)
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 Through 7.0.1 - CSRF Bypass Leading To DOS Vulnerability (CVE-2020-13379)

Signature update for October 2021

October 22, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-10-09. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 69 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 and Citrix ADC 13.1 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999149	CVE-2021-38312	WEB-WORDPRESS Gutenberg Template Library and Redux Framework Plugin Prior to 4.2.12 - REST_ROUTE Vulnerability (CVE-2021-38312)
999150	CVE-2021-38312	WEB-WORDPRESS Gutenberg Template Library and Redux Framework Plugin Prior to 4.2.12 - REST API Vulnerability (CVE-2021-38312)
999151	CVE-2021-34639	WEB-WORDPRESS Download Manager Plugin Prior to 3.1.25 - Double Extension Upload Vulnerability (CVE-2021-34639)
999152	CVE-2021-34621	WEB-WORDPRESS ProfilePress Plugin Prior to 3.1.3 - Elevation of Privilege Vulnerability Via wp_capabilities (CVE-2021-34621)
999153	CVE-2021-32682	WEB-MISC elFinder Prior To 2.1.59 - Path Traversal Vulnerability Via Rename Command (CVE-2021-32682)
999154	CVE-2021-32682	WEB-MISC elFinder Prior To 2.1.59 - Path Traversal Vulnerability Via Abort Command (CVE-2021-32682)
999155	CVE-2021-26086	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via WEB-INF (CVE-2021-26086)

Signature rule	CVE ID	Description
999156	CVE-2021-26086	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via META-INF (CVE-2021-26086)
999157	CVE-2021-22005	WEB-MISC VMWare vCenter - File Upload Vulnerability Via Data App (CVE-2021-22005)
999158	CVE-2021-22005	WEB-MISC VMWare vCenter - File Upload Vulnerability Via Telemetry Stage Log (CVE-2021-22005)
999159	CVE-2021-22005	WEB-MISC VMWare vCenter - File Upload Vulnerability Via Telemetry Prod Log (CVE-2021-22005)
999160	CVE-2021-20081	WEB-MISC Zoho ManageEngine Service Desk Prior to 11.2.0.5 - Remote Code Execution Vulnerability (CVE-2021-20081)
999161	CVE-2020-29453	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via WEB-INF (CVE-2020-29453)
999162	CVE-2020-29453	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via META-INF (CVE-2020-29453)

Signature update for October 2021

November 19, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-10-26. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 70 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999127	CVE-2021-42013	WEB-MISC Apache HTTP Server 2.4.49 and 2.4.50 - Path Traversal Vulnerability Via %%32 (CVE-2021-42013)
999128	CVE-2021-42013	WEB-MISC Apache HTTP Server 2.4.49 and 2.4.50 - Path Traversal Vulnerability Via %2% (CVE-2021-42013)
999129	CVE-2021-41773	WEB-MISC Apache HTTP Server 2.4.49 - Path Traversal Vulnerability Via %2e%2e (CVE-2021-41773)
999130	CVE-2021-41773	WEB-MISC Apache HTTP Server 2.4.49 - Path Traversal Vulnerability Via .%2e (CVE-2021-41773)

Signature rule	CVE ID	Description
999131	CVE-2021-40539	WEB-MISC Zoho ManageEngine ADSelfService Plus 6.1 Prior to Build 6114 - Authentication Bypass Vulnerability (CVE-2021-40539)
999132	CVE-2021-34648	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST_ROUTE Vulnerability via submissions email-action (CVE-2021-34648)
999133	CVE-2021-34648	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST API Vulnerability via submissions email-action (CVE-2021-34648)
999134	CVE-2021-34647	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST_ROUTE Vulnerability via Submissions Export (CVE-2021-34647)
999135	CVE-2021-34647	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST API Vulnerability via Submissions Export (CVE-2021-34647)
999136	CVE-2021-34623	WEB-WORDPRESS ProfilePress Plugin Prior to 3.1.4 - Arbitrary File Upload Vulnerability Via eup_cover_image (CVE-2021-34623)

Signature rule	CVE ID	Description
999137	CVE-2021-34623	WEB-WORDPRESS ProfilePress Plugin Prior to 3.1.4 - Arbitrary File Upload Vulnerability Via eup_avatar (CVE-2021-34623)
999138	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via mobile X ReportTemplateService(CVE-2021-2400)
999139	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via mobile ReportTemplateService(CVE-2021-2400)
999140	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via xmlpservice X ReportTemplateService (CVE-2021-2400)
999141	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via xmlpservice ReportTemplateService (CVE-2021-2400)
999142	CVE-2021-21985	WEB-MISC VMWare vCenter - Virtual SAN Health Check Plugin Remote Code Execution Vulnerability (CVE-2021-21985)
999143	CVE-2021-20078	WEB-MISC Zoho ManageEngine OpManager 12.5 Prior to Build 125362 - Path Traversal Vulnerability (CVE-2021-20078)

Signature rule	CVE ID	Description
999144	CVE-2020-29448	WEB-MISC Atlassian Confluence Server and Data Center - Information Disclosure Vulnerability Via WEB-INF (CVE-2020-29448)
999145	CVE-2020-29448	WEB-MISC Atlassian Confluence Server and Data Center - Information Disclosure Vulnerability Via META-INF (CVE-2020-29448)
999146	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3 - Unauthenticated SQL Injection Vulnerability Via osupdate Endpoint (CVE-2020-12442)
999147	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3 - Unauthenticated SQL Injection Vulnerability Via wapl Endpoint (CVE-2020-12442)
999148		WEB-WORDPRESS BuddyPress Plugin Prior to 9.1.1 - SQL Injection Vulnerability Via bp-members-invitations Feature

Signature update for November 2021

November 19, 2021

New signatures rules are generated for the vulnerabilities identified in the week 2021-11-18. You can download and configure these signature rules to protect your appliance from security vulnerable attacks.

Signature version

Signature version 71 applicable for NetScaler VPX 11.1, NetScaler 12.0, Citrix ADC 12.1, Citrix ADC 13.0 platforms.

> **Note:**
>
> Enabling Post body and Response body signature rules might affect Citrix ADC CPU.

Common Vulnerability Entry (CVE) insight

Following is a list of signature rules, CVE IDs, and its description.

Signature rule	CVE ID	Description
999098	CVE-2021-41765	WEB-MISC ResourceSpace 9.5 and 9.6 prior to rev 18274 - SQL Injection Vulnerability (CVE-2021-41765)
999099	CVE-2021-41288	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125467 - SQL Injection Vulnerability Via getReportData API (CVE-2021-41288)
999100	CVE-2021-40493	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125437 - SQL Injection Vulnerability Via deviceName (CVE-2021-40493)
999101	CVE-2021-40493	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125437 - SQL Injection Vulnerability Via pollingObject (CVE-2021-40493)
999102	CVE-2021-40438	WEB-MISC Apache HTTP Server - mod_proxy Request Forward Vulnerability (CVE-2021-40438)

Signature rule	CVE ID	Description
999103	CVE-2021-39341	WEB-WORDPRESS OptinMonster Plugin Up to 2.6.4 - REST_ROUTE Permission Bypass Vulnerability (CVE-2021-39341)
999104	CVE-2021-39341	WEB-WORDPRESS OptinMonster Plugin Up to 2.6.4 - REST API Permission Bypass Vulnerability (CVE-2021-39341)
999105	CVE-2021-37344	WEB-MISC Nagios XI Switch Wizard Prior to 2.5.7 - Remote Code Execution Vulnerability Via ip_address Parameter (CVE-2021-37344)
999106	CVE-2021-35218	WEB-MISC SolarWinds Orion Prior to 2020.2.6 - Deserialization Vulnerability Via Chart.ashx (CVE-2021-35218)
999107	CVE-2021-35215	WEB-MISC SolarWinds Orion Platform Prior to 2020.2.6 - Remote Code Execution Vulnerability Via Reporting (CVE-2021-35215)
999108	CVE-2021-35215	WEB-MISC SolarWinds Orion Platform Prior to 2020.2.6 - Remote Code Execution Vulnerability Via Alerting (CVE-2021-35215)
999109	CVE-2021-24889	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.6.4 - SQL Injection Vulnerability (CVE-2021-24889)

Signature rule	CVE ID	Description
999110	CVE-2021-24381	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.5.8.2 - Custom Class Name Stored Cross-Site Scripting Vulnerability (CVE-2021-24381)
999111	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via mobile X ReportTemplateService (CVE-2021-2401)
999112	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via mobile ReportTemplateService (CVE-2021-2401)
999113	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via xmlpservice X ReportTemplateService (CVE-2021-2401)
999114	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via xmlpservice ReportTemplateService (CVE-2021-2401)
999115	CVE-2021-2392	WEB-MISC Oracle BI Publisher - Arbitrary Files Upload Vulnerability (CVE-2021-2392)
999116	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase Analytic Provider Services - Remote Code Execution Vulnerability Via Essbase (CVE-2021-2244)

Signature rule	CVE ID	Description
999117	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase Analytic Provider Services - Remote Code Execution Vulnerability Via admin (CVE-2021-2244)
999118	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase Analytic Provider Services - Remote Code Execution Vulnerability Via JAPI (CVE-2021-2244)
999119	CVE-2021-22205	WEB-MISC GitLab CE/EE - Remote Code Execution Vulnerability Via Maliciously Crafted JPEG/TIFF Files (CVE-2021-22205)
999120	CVE-2021-22017	WEB-MISC VMWare vCenter - Path Traversal Vulnerability Via rhhtproxy (CVE-2021-22017)
999121	CVE-2021-20837	WEB-MISC Movable Type Prior to r.5003 - Remote Code Execution Via mt.handler_to_coderef (CVE-2021-20837)
999122	CVE-2021-20131	WEB-MISC Zoho ManageEngine ADManager Prior to Build 7115 - Remote Code Execution Vulnerability Via File Upload (CVE-2021-20131)
999123	CVE-2021-20130	WEB-MISC Zoho ManageEngine ADManager Prior to Build 7115 - Remote Code Execution Vulnerability Via File Upload (CVE-2021-20130)

Signature rule	CVE ID	Description
999124	CVE-2021-20034	WEB-MISC SonicWall Secure Mobile Access - Path Traversal Vulnerability (CVE-2021-20034)
999125		WEB-WORDPRESS BuddyPress Plugin Prior to 9.1.1 - Information Disclosure Vulnerability Via signup REST API and rest_route
999126		WEB-WORDPRESS BuddyPress Plugin Prior to 9.1.1 - Information Disclosure Vulnerability Via signup REST API

Bot Management

September 14, 2021

Sometimes the incoming web traffic is comprised of bots and most organizations suffer from bot attacks. Web and mobile applications are significant revenue drivers for business and most companies are under the threat of advanced cyberattacks, such as bots.

A bot is a software program that automatically performs certain actions repeatedly at a much faster rate than a human. Bots can interact with webpages, submit forms, run actions, scan texts, or download content. They can access videos, post comments, and tweet on social media platforms. Some bots, known as chatbots, can hold basic conversations with human users.

A bot that performs a helpful service, such as customer service, automated chat, and search engine crawlers are good bots. At the same time, a bot that can scrape or download content from a website, steal user credentials, spam content, and perform other kinds of cyberattacks are bad bots.

With a good number of bad bots performing malicious tasks, it is essential to manage bot traffic and protect your web applications from bot attacks. By using Citrix bot management, you can detect the incoming bot traffic and mitigate bot attacks to protect your web applications.

Citrix bot management helps identify bad bots and protect your appliance from advanced security attacks. It detects good and bad bots and identifies if incoming traffic is a bot attack. By using bot management, you can mitigate attacks and protect your web applications.

Citrix ADC bot management provides the following benefits:

- **Defend against bots, scripts, and toolkits**. Provides real-time threat mitigation using static signature based defense and device fingerprinting.
- **Neutralize automated basic and advanced attacks**. Prevents attacks, such as App layer DDoS, password spraying, password stuffing, price scrapers, and content scrapers.
- **Protect your APIs and investments**. Protects your APIs from unwarranted misuse and protects infrastructure investments from automated traffic.

Some use cases where you can benefit by using the Citrix bot management system are:

- Brute force login. A government web portal is constantly under attack by bots attempting to brute force user logins. The organization discovered the attack by looking through web logs and seeing specific users being select over and over again with rapid login attempts and passwords incrementing using a dictionary attack approach. By law, they must protect themselves and their users. By deploying the Citrix bot management, they can stop brute force login using device fingerprinting and rate limiting techniques.
- Block bad bots and device fingerprint unknown bots. A web entity gets 100,000 visitors each day. They have to upgrade the underlying footprint and they are spending a fortune. In a recent audit, the team discovered that 40 percent of the traffic came from bots, scraping content, picking news, checking user profiles, and more. They want to block this traffic to protect their users and reduce their hosting costs. Using bot management, they can block known bad bots, and fingerprint unknown bots that are hammering their site. By blocking these bots, they can reduce bot traffic by 90 percent.

What does Citrix bot management do

The Citrix bot management helps organizations protect their web applications and public assets from advanced security attacks. When an incoming traffic is a bot, the bot management system detects the bot type, assigns an action, and generates bot insights, as shown in the following diagram.



How does Citrix ADC bot management work

The following diagram shows how the Citrix ADC bot management works. The process involves eight detection techniques that help in detecting the incoming traffic as a good or a bad bot. By default good bots detected by signatures are allowed and bad bots detected by signatures are dropped.



1. The process starts by enabling the bot management feature on the appliance.
2. When a client sends a request, the appliance evaluates the traffic using bot policy rules. If the incoming request is identified as a bot, the appliance applies a bot detection profile.
3. You must bind the default or custom bot signature file to the bot detection profile. The bot signature file has a list of bot signature rules for identifying the incoming bot type.
4. The bot detection rules are available under eight detection categories in the signature file. The categories are allow list, block list, static signature, IP reputation, device fingerprint, and rate limiting. Based on the bot traffic, the system applies a detection rule to the traffic.
5. If the incoming bot traffic matches an entry in the bot allow list, the system bypasses other detection techniques and the associated action logs the data.
6. For detection techniques other than bot allow list, if an incoming request matches a configured rule, the corresponding action is applied. The possible actions are drop, redirect, reset, mitigation, and log. CAPTCHA is a mitigation action which is supported for IP reputation, device fingerprinting, and TPS detection techniques.

Bot Detection

November 23, 2021

The Citrix ADC bot management system uses six different techniques to detect the incoming bot traffic. The techniques are used as detection rules to detect the bot type. The techniques are bot allow list, bot block list, IP reputation, device fingerprinting, rate limiting, bot trap, TPS, and CAPTCHA.

> **Note:**
>
> Bot management supports a maximum of 32 configuration entities for block list, allow list, and rate limiting techniques.

Bot white list. A customized list of IP addresses, subnets, and policy expressions that can be bypassed as an allowed list.

Bot black list. A customized list of IP addresses, subnets, and policy expressions that has to be blocked from accessing your web applications.

IP reputation. This rule detects if the incoming bot traffic is from a malicious IP address.

Device fingerprint. This rule detects if the incoming bot traffic has the device fingerprint ID in the incoming request header and browser attributes of an incoming client bot traffic.

> **Limitation:**
>
> 1. Java Script must be enabled in the client browser.
> 2. Does not work for XML responses.

Bot log expression. The detection technique enables you to capture additional information as log messages. The data can be the name of the user who requested the URL, the source IP address, and the source port from which the user sent the request or data generated from an expression.

Rate limit. This rule rate limits multiple requests coming from the same client.

Bot trap. Detects and blocks automated bots by advertising a trap URL in the client response. The URL appears invisible and not accessible if the client is a human user. The detection technique is effective in blocking attacks from automated bots.

TPS. Detects incoming traffic as bots if the maximum number of requests and percentage increase in requests exceeds the configured time interval.

CAPTCHA. This rule uses a CAPTCHA for mitigating bot attacks. A CAPTCHA is a challenge-response validation to determine if the incoming traffic is from a human user or an automated bot. The validation helps block automated bots that cause security violations to web applications. You can configure CAPTCHA as a bot action in IP reputation and device fingerprint detection techniques.

Now, let us see how you can configure each technique to detect and manage your bot traffic.

How to upgrade your appliance to Citrix ADC CLI-based bot management configuration

If you are upgrading your appliance from an older version (Citrix ADC release 13.0 build 58.32 or earlier), you must first manually convert the existing bot management configuration to the Citrix ADC CLI based bot management configuration only once. Complete the following steps to manually convert your bot management configuration.

1. After upgrading to the latest version connect to the upgrade tool “upgrade_bot_config.py” by using the following command

   At the command prompt, type:

   ```
   shell "/var/python/bin/python /netscaler/upgrade_bot_config.py > /var/bot_upgrade_commands.txt"
   ```

2. Run the configuration using the following command.

   At the command prompt, type:

   ```
   batch -f /var/bot_upgrade_commands.txt
   ```

3. Save the upgraded configuration.

```
save ns config
```

Configure Citrix ADC CLI-based bot management

The bot management configuration enables you to bind one or more bot detection techniques to a specific bot profile. You begin the process by enabling the bot management feature on your appliance. Once you enable, you import the bot signature file into the appliance. After import, you must create a bot profile. You then create a bot policy with the bot profile bound to it for evaluating the incoming traffic as bot and bind the policy globally or to a virtual server.

> **Note:**
>
> If you are upgrading your appliance from an older version, you must first manually convert the existing bot management configuration. For more information, see How to upgrade to Citrix ADC CLI-based bot management configuration section.

You must complete the following steps to configure Citrix ADC-based bot management:

1. Enable bot management
2. Import bot signature
3. Add bot profile
4. Bind bot profile
5. Add bot policy
6. Bind bot policy
7. Configure bot settings

Enable bot management

Before you can begin, ensure that the Bot Management feature is enabled on the appliance. If you have a new Citrix ADC or VPX, you must enable the feature before you configure it. If you are upgrading a Citrix ADC or VPX appliance from an earlier version of the Citrix ADC software version to the current version, you must need to enable the feature before you configure it. At the command prompt, type:

```
enable ns feature Bot
```

Import bot signature

You can import the default signature bot file and bind it to the bot profile. At the command prompt, type:

```
import bot signature [<src>] <name> [-comment <string>] [-overwrite]
```

Where,
src. Local path to and name of, or URL (protocol, host, path, and file name) for, the file in which to store the imported signature file. Note: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Maximum Length: 2047
name. Name to assign to the bot signature file object on the Citrix ADC. This is a mandatory argument. Maximum Length: 31
comment. Any comments to preserve information about the signature file object. Maximum Length: 255.
Overwrite. Overwrites the existing file.
Note: Use the `overwrite` option to update the content in the signature file. Alternately, use the `update bot signature <name>` command to update the signature file on the Citrix ADC appliance

Example

```
import bot signature http://www.example.com/signature.json signaturefile -comment commentsforbot -overwrite
```

> **Note:**
>
> You can use the overwrite option to update the content in the signature file. Also, you can use the `update bot signature <name>` command to update the signature file in the Citrix ADC appliance.

Add bot profile

A bot profile is a collection of profile settings to configure bot management on the appliance. You can configure the settings to perform bot detection.

At the command prompt, type:

```
add bot profile <name> [-signature <string>] [-errorURL <string>] [-trapURL
 <string>] [-comment <string>] [-whiteList ( ON | OFF )] [-blackList ( ON
 | OFF )] [-rateLimit ( ON | OFF )] [-deviceFingerprint ( ON | OFF )] [-
deviceFingerprintAction ( none | log | drop | redirect | reset | mitigation
 )] [-ipReputation ( ON | OFF )] [-trap ( ON | OFF )] [-trapAction ( none |
 log | drop | redirect | reset )] [-tps ( ON | OFF )]
```

Example:

```
add bot profile profile1 -signature signature -errorURL http://www.example
.com/error.html -trapURL /trap.html -whitelist ON -blacklist ON -ratelimit
 ON -deviceFingerprint ON -deviceFingerprintAction drop -ipReputation ON -
trap ON
```

Bind bot profile

After you create a bot profile, you must bind the bot detection mechanism to the profile.

At the command prompt, type:

```
bind bot profile <name> ((-blackList [-type ( IPv4 | Subnet | Expression
 )] [-enabled ( ON | OFF )] [-value <string>] [-action ( log | drop |
reset )] [-logMessage <string>] [-comment <string>])| (-whiteList [-type
 ( IPv4 | Subnet | Expression )] [-enabled ( ON | OFF )] [-value <string
>] [-log ( ON | OFF )] [-logMessage <string>] [-comment <string>]))|
(-rateLimit [-type ( session |SOURCE_IP | url )] [-enabled ( ON | OFF
)] [-url <string>] [-cookieName <string>] [-rate <positive_integer>] [-
timeslice <positive_integer>] [-action ( none | log | drop | redirect |
reset )] [-logMessage <string>] [-comment <string>])| (-ipReputation [-
category <ipReputationCategory>] [-enabled ( ON | OFF )] [-action ( none
 | log | drop | redirect | reset | mitigation )] [-logMessage <string>]
[-comment <string>])| (-captchaResource [-url <string>] [-enabled ( ON |
 OFF )] [-waitTime <positive_integer>] [-gracePeriod <positive_integer>]
 [-mutePeriod <positive_integer>] [-requestLengthLimit <positive_integer
>] [-retryAttempts <positive_integer>] [-action ( none | log | drop |
redirect | reset )] [-logMessage <string>] [-comment <string>])| (-tps
 [-type ( SOURCE_IP | GeoLocation | REQUEST_URL | Host )] [-threshold <
positive_integer>] [-percentage <positive_integer>] [-action ( none | log |
 drop | redirect | reset | mitigation )] [-logMessage <string>] [-comment <
string>])
```

Example:

The following example is for binding the IP reputation detection technique to a specific bot profile.

```
bind bot profile profile5 -ipReputation -category BOTNET -enabled ON -
action drop -logMessage message
```

Add bot policy

You must add the bot policy for evaluating bot traffic.

At the command prompt, type:

```
add bot policy <name> -rule <expression> -profileName <string> [-undefAction
 <string>] [-comment <string>] [-logAction <string>]
```

Where,

Name. Name for the bot policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the bot policy is added.

Rule. Expression that the policy uses to determine whether to apply the bot profile on the specified request. This is a mandatory argument. Maximum Length: 1499

profileName. Name of the bot profile to apply if the request matches this bot policy. This is a mandatory argument. Maximum Length: 127

undefAction. Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Maximum Length: 127

Comment. Any type of information about this bot policy. Maximum Length: 255

logAction. Name of the log action to use for requests that match this policy. Maximum Length: 127

Example:

```
add bot policy pol1 -rule "HTTP.REQ.HEADER(\"header\").CONTAINS(\"custom
\")"- profileName profile1 -undefAction drop -comment commentforbotpolicy -
logAction log1
```

Bind bot policy global

At the command prompt, type:

```
bind bot global -policyName <string> -priority <positive_integer> [-gotoPriorityExpres
 <expression>][-type ( REQ_OVERRIDE | REQ_DEFAULT )] [-invoke (-labelType (
 vserver | policylabel )-labelName <string>)]
```

Example:


```
bind bot global -policyName pol1 -priority 100 -gotoPriorityExpression NEXT
-type REQ_OVERRIDE
```

Bind bot policy to a virtual server

At the command prompt, type:

```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] )| <
serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>]
 [-gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )] [-
invoke (<labelType> <labelName>)] )| -analyticsProfile <string>@)
```

Example:

```
bind lb vserver lb-server1 -policyName pol1 -priority 100 -gotoPriorityExpression
 NEXT -type REQ_OVERRIDE
```

Configure bot settings

You can customize the default settings if necessary.
At the command prompt, type:

```
set bot settings [-defaultProfile <string>] [-javaScriptName <string>]
 [-sessionTimeout <positive_integer>] [-sessionCookieName <string>] [-
dfpRequestLimit <positive_integer>] [-signatureAutoUpdate ( ON | OFF )]
[-signatureUrl <URL>] [-proxyServer <ip_addr|ipv6_addr|*>] [-proxyPort <
port|*>]
```

Where,

defaultProfile. Profile to use when a connection does not match any policy. Default setting is “, which sends unmatched connections back to the Citrix ADC without attempting to filter them further. Maximum Length: 31

javaScriptName. Name of the JavaScript that the BotNet feature uses in response. Must begin with a letter or number, and can consist of from 1 to 31 letters, numbers, and the hyphen (-) and underscore (_) symbols. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my cookie name” or ‘my cookie name’). Maximum Length: 31

sessionTimeout. Session times out, in seconds, after which a user session is terminated.

Minimum value: 1, Maximum value: 65535

sessionCookieName. Name of the SessionCookie that the BotNet feature uses it for tracking. Must begin with a letter or number, and can consist of from 1 to 31 letters, numbers, and the hyphen (-)

and underscore (_) symbols. The following requirement applies only to the Citrix ADC CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my cookie name” or ‘my cookie name’). Maximum Length: 31

dfpRequestLimit. Number of requests to allow without bot session cookie if device fingerprint is enabled.

Minimum value: 1, Maximum Value: 4294967295

signatureAutoUpdate. Flag used to enable/disable bot auto update signatures.

Possible values: ON, OFF

Default value: OFF

signatureUrl. URL to download the bot signature mapping file from the server.

Default value: `https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`.

Maximum Length: 2047

ProxyServer. Proxy Server IP to get updated signatures from AWS.

proxyPort. Proxy Server Port to get updated signatures from AWS. Default value: 8080

Example:

```
set bot settings -defaultProfile profile1 -javaScriptName json.js -sessionTimeout
 1000 -sessionCookieName session
```

Configuring bot management by using Citrix ADC GUI

You can configure Citrix ADC bot management by first enabling the feature on the appliance. Once you enable, you can create a bot policy to evaluate the incoming traffic as bot and send the traffic to the bot profile. Then, you create a bot profile and then bind the profile to a bot signature. As an alternative, you can also clone the default bot signature file and use the signature file to configure the detection techniques. After creating the signature file, you can import it into the bot profile.



1. Enable bot management feature
2. Configure bot management settings
3. Clone Citrix bot default signature
4. Import Citrix bot signature
5. Configure bot signature settings
6. Create bot profile
7. Create bot policy

Enable bot management feature

Complete the following steps to enable bot management:

1. On the navigation pane, expand **System** and then click **Settings**.
2. On the **Configure Advanced Features** page, select the **Bot Management** check box.
3. Click **OK**, and then click **Close**.



Configure bot management settings for device fingerprint technique

Complete the following step to configure the device fingerprint technique:

1. Navigate to **Security** > **Citrix Bot Management**.
2. In the details pane, under **Settings** click **Change Citrix Bot Management Settings**.
3. In the **Configure Citrix Bot Management Settings**, set the following parameters.
   a) Default Profile. Select a bot profile.
   b) JavaScript Name. Name of the JavaScript file that bot management uses in its response to the client.
   c) Session Timeout. Timeout in seconds after which the user session is terminated.

d) Session Cookie. Name of the session cookie that the bot management system uses for tracking.

e) Device Fingerprint Request Limit. Number of requests to allow without a bot session cookie, if device fingerprint is enabled



4. Click **OK**.

Clone bot signature file

Complete the following step to clone the bot signature file:

1. Navigate to **Security** > **Citrix Bot Management** and **Signatures**.
2. In **Citrix Bot Management Signatures** page, select the default bot signatures record and click **Clone**.
3. In the **Clone Bot Signature** page, enter a name and edit the signature data.
4. Click **Create**.



Import bot signature file

If you have your own signature file, then you can import it as a file, text, or URL. Perform the following the steps to import the bot signature file:

1. Navigate to **Security** > **Citrix Bot Management** and **Signatures**.
2. On the **Citrix Bot Management Signatures** page, import the file as URL, File, or text.
3. Click **Continue**.



4. On the Import Citrix Bot Management Signature page, set the following parameters.
   a) Name. Name of the bot signature file.
   b) Comment. Brief description about the imported file.
   c) Overwrite. Select the check box to allow overwriting of data during file update.
   d) Signature Data. Modify signature parameters
5. Click **Done**.



Configure bot allow list by using Citrix ADC GUI

This detection technique enables you to bypass URLs that you configure an allowed listed one. Complete the following step to configure an allow list URL:

1. Navigate to **Security** > **Citrix Bot Management** and **Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a file and click **Edit**.
3. On the **Citrix Bot Management Profile** page, go to the **Signature Settings** section and click **White List**.
4. In the **White List** section, set the following parameters:
    a) Enabled. Select the check box to validate the allow list URLs as part of the detection process.
    b) Configure Types. Configure an allow list URL. The URL is bypassed during bot detection. Click Add to add a URL to the bot allow list.
    c) In the **Configure Citrix Bot Management Profile Whitelist Binding** page, set the following parameters:
        i. Type. URL type can be an IPv4 address, subnet IP address, or an IP address matching a policy expression.
        ii. Enabled. Select the check box to validate the URL.
        iii. Value. URL address.
        iv. Log. Select the check box to store log entries.
        v. Log Message. Brief description of the log.

vi. Comments. Brief description about the allow list URL.

vii. Click **OK**.



5. Click **Update**.
6. Click **Done**.



Configure bot block list by using Citrix ADC GUI

This detection technique enables you to drop the URLs that you configure as block listed one. Complete the following step to configure a block list URL.

1. Navigate to **Security** > **Citrix Bot Management** and **Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a signature file and click **Edit**.
3. On the **Citrix Bot Management Profile** page, go to **Signature Settings** section and click **Black List**.
4. In the **Black List** section, set the following parameters:

a) Enabled. Select the check box to validate block list URLs as part of the detection process.

b) Configure Types. Configure a URL to be part of the bot block list detection process. These URLs are dropped during bot detection. Click Add to add a URL to the bot block list

c) In the **Configure Citrix Bot Management Profile Blacklist Binding** page, set the following parameters.

i. Type. URL type can be an IPv4 address, subnet IP address, or IP address.
ii. Enabled. Select the check box to validate the URL.
iii. Value. URL address.
iv. Log. Select the check box to store log entries.
v. Log Message. Brief description of the login.
vi. Comments. Brief description about the block list URL.
vii. Click **OK**.



5. Click **Update**.
6. Click **Done**.



Configure IP reputation by using Citrix ADC GUI

This configuration is a pre-requisite for the bot IP reputation feature. The detection technique enables you to identify if there is any malicious activity from an incoming IP address. As part of the configuration, we set different malicious bot categories and associate a bot action to each of it. Complete the following step to configure the IP reputation technique.

1. Navigate to **Security** > **Citrix Bot Management** and **Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a signature file and click **Edit**.
3. On the **Citrix Bot Management Profile** page, go to **Signature Settings** section and click **IP Reputation**.
4. On the **IP Reputation** section, set the following parameters:
    a) Enabled. Select the check box to validate incoming bot traffic as part of the detection process.
    b) Configure Categories. You can use the IP reputation technique for incoming bot traffic under different categories. Based on the configured category, you can drop or redirect the bot traffic. Click **Add** to configure a malicious bot category.
    c) In the **Configure Citrix Bot Management Profile IP Reputation Binding** page, set the following parameters:
        i. Category. Select a malicious bot category from the list. Associate a bot action based on category.
        ii. Enabled. Select the check box to validate the IP reputation signature detection.
        iii. Bot action. Based on the configured category, you can assign no action, drop, redirect, mitigation, or CAPTCHA action.
        iv. Log. Select the check box to store log entries.
        v. Log Message. Brief description of the log.
        vi. Comments. Brief description about the bot category.
5. Click **OK**.
6. Click **Update**.
7. Click **Done**.



Configure bot rate limit by using Citrix ADC GUI

This detection technique enables you to block bots based on the number of requests received within a predefined time from a client IP address, a session, or a configured resource (for example, from a URL). Complete the following step to configure the rate limit technique.

1. Navigate to **Security** > **Citrix Bot Management** and **Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a signature file and click **Edit**.
3. On the **Citrix Bot Management Profile** page, go to **Signature Settings** section and click **Rate Limit**.
4. On the **Rate Limit** section, set the following parameters:
    a) Enabled. Select the check box to validate the incoming bot traffic as part of the detection process.
    b) Session. Rate limit requests based on a session. Click Add to configure rate limit requests based on a session.
    c) In the **Configure Citrix Bot Management Signature Rate Limit** page, set the following parameters.
        i. Category. Select a malicious bot category from the list. Associate an action based on the category.
        ii. Enabled. Select the check box to validate the incoming bot traffic.
        iii. Bot action. Choose a bot action for the selected category.
        iv. Log. Select the check box to store log entries.
        v. Log Message. Brief description of the log.
        vi. Comments. Brief description about the bot category.
        vii. Click **OK**.



5. Click **Update**.
6. Click **Done**.



Configure device fingerprint technique by using Citrix ADC GUI

This detection technique sends a java script challenge to the client and extracts the device information. Based on device information, the technique drops or bypasses the bot traffic. Follow the steps to configure the detection technique.

1. Navigate to **Security** > **Citrix Bot Management** and **Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a signature file and click **Edit**.
3. On the **Citrix Bot Management Profile** page, go to the **Signature Settings** section and click **Device Fingerprint**.
4. In the **Device Fingerprint** section, set the following parameters:
   a) Enabled. Set this option to enable the rule.
   b) Configuration. For the given device fingerprint, assign no action, drop, or redirect, mitigation, or CAPTCHA action.
   c) Log. Select the check box to store log entries.
5. Click **Update**.
6. Click **Done**.



Configure device fingerprint technique for mobile (Android) applications

Device fingerprint technique detects an incoming traffic as a bot by inserting a JavaScript script in the HTML response to the client. The JavaScript script when invoked by the browser, it collects browser and client attributes and sends a request to the appliance. The attributes are examined to determine whether the traffic is a Bot or a human.

The detection technique is further extended to detect bots on a mobile (Android) platform. Unlike web applications, in mobile (Android) traffic, bot detection based on JavaScript script do not apply. To detect bots in a mobile network, the technique uses a bot mobile SDK which is integrated with mobile applications on the client-side. The SDK intercepts the mobile traffic, collects device details, and sends the data to the appliance. On the appliance side, the detection technique examines the data and determines whether the connection is from a Bot or a human.

How the device fingerprint technique for mobile application works

The following steps explain the bot detection workflow to detect if a request from a mobile device is from a human or a bot.

1. When a user interacts with a mobile application, the device behavior is recorded by the bot mobile SDK.
2. Client sends a request to Citrix ADC appliance.
3. When sending the response, the appliance inserts a bot session cookie with session details, and parameters to collect client parameters.
4. When the mobile application receives the response, the Citrix bot SDK which is integrated with the mobile application validates the response, retrieves the recorded device fingerprint parameters, and sends it to the appliance.
5. The device fingerprint detection technique on the appliance side validates the device details and updates the bot session cookie if it is a suspected bot or not.
6. When the cookie is expired or device fingerprint protection prefers to validate and collect device parameters periodically, the whole procedure or challenge is repeated.

Pre-requisite

To get started with the Citrix ADC device fingerprint detection technique for mobile applications, you must download and install the bot mobile SDK in your mobile application.

Configure fingerprint detection technique for mobile (Android) applications by using the CLI

At the command prompt, type:

```
set bot profile <profile name> -deviceFingerprintMobile ( NONE | Android )
```

Example:

```
set bot profile profile 1 -deviceFingerprintMobile Android
```

Configure device fingerprint detection technique for mobile (Android) applications by using the GUI

1. Navigate to **Security** > **Citrix Bot Management** and **Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a file and click **Edit**.
3. On the **Citrix Bot Management Profile** page, click **Device Fingerprint** under **Profile Settings**.
4. In the **Configure Bot Mobile SDK** section, select the mobile client type.
5. Click **Update** and **Done**.



Configure bot log expression

If the client is identified as a bot, the Citrix bot management enables you to capture additional information as log messages. The data can be the name of the user who requested the URL, the source IP address, and the source port from which the user sent the request or data generated from an expression. To perform custom logging, you must configure a log expression in the bot management profile.

Bind the log expression in the bot profile by using the CLI

At the command prompt, type:

```
1 bind bot profile <name> (-logExpression -name <string> -expression <
     expression> [-enabled ( ON | OFF )]) -comment <string>
2 <!--NeedCopy-->
```

Example:

```
bind bot profile profile1 -logExpression exp1 -expression HTTP.REQ.URL -
enabled ON -comment "testing log expression"
```

Bind log expression to bot profile by using the GUI

1. Navigate to **Security > Citrix Bot Management > Profiles**.
2. On the **Citrix Bot Management Profiles** page, select **Bot Log Expressions** from **Profile Settings** section.
3. In the **Bot Log Expression Settings* section, click **Add**.
4. In the **Configure Citrix Bot Management Profile Bot Log Expression Binding** page, set the following parameters.
    a) Log Expression Name. Name of the log expression.

b) Expression. Enter the log expression.
c) Enabled. Enable or disable the log expression binding.
d) Comments. A brief description about the bot log expression binding.

5. Click**OK** and **Done**.



Configure bot trap technique

The Citrix bot trap technique randomly or periodically inserts a trap URL in the client response. You can also create a trap URL list and add URLs for that The URL appears invisible and not accessible if the client is a human user. However, if the client is an automated bot, the URL is accessible and when accessed, the attacker is categorized as bot and any subsequent request from the bot is blocked. The trap technique is effective in blocking attacks from bots.

The trap URL is an alpha-numeric URL of configurable length and it is auto-generated at configurable interval. Also the technique allows you to configure a trap injection URL for top visited websites or frequently visited websites. By doing this, you can mandate the purpose of injecting the bot trap URL for requests matching the trap injection URL.

> **Note:**
>
> Although the bot trap URL is auto-generated, the Citrix ADC bot management still allows you to configure a customized trap URL in the bot profile. This is done to strengthen the bot detection technique and make it harder for attackers to access the trap URL.

To complete the bot trap configuration, you must complete the following steps.

1. Enable bot trap URL
2. Configure bot trap URL in bot profile
3. Bind bot trap injection URL to bot profile
4. Configure bot trap URL length and interval in bot settings

Enable bot trap URL protection

Before you can begin, you must ensure the Bot trap URL protection is enabled on the appliance. At the command prompt, type:

```
enable ns feature Bot
```

Configure bot trap URL in bot profile

You can configure the bot trap URL and specify a trap action in the bot profile.
At the command prompt, type:

```
add bot profile <name> -trapURL <string> -trap ( ON | OFF )-trapAction <
trapAction>
```

Where,

`trapURL`. URL that Bot protection uses as the Trap URL. Maximum Length: 127

`trap`. Enable bot trap detection. Possible values: ON, OFF, Default value: OFF

`trapAction`. Action to be taken based bot detection. Possible values: NONE, LOG, DROP, REDIRECT, RESET, MITIGATION. Default value: NONE

Example:

```
add bot profile profile1 -trapURL www.bottrap1.com trap ON -trapAction
RESET
```

Bind bot trap injection URL to bot profile

You can configure the bot trap injection URL and bind it to the bot profile.
At the command prompt, type:


```
bind bot profile <profile_name> trapInjectionURL -url <url> -enabled ON|OFF
-comment <comment>
```

Where,

URL. Request URL regex pattern for which the bot trap URL is inserted. Maximum Length: 127

Example:

```
bind bot profile profile1 trapInjectionURL -url www.example.com -enabled ON
-comment insert a trap URL randomly
```

Configure bot trap URL length and interval in bot settings

You can configure the bot trap URL length and also set the interval to auto generate the bot trap URL. At the command prompt, type:

```
set bot settings -trapURLAutoGenerate ( ON | OFF )-trapURLInterval <positive_integer
> -trapURLLength <positive_integer>
```

Where,

trapURLInterval. Time in seconds after which the bot trap URL is updated. Default value: 3600, Minimum value: 300, Maximum value: 86400

trapURLLength. Length of the auto-generated bot trap URL. Default value: 32, Minimum value: 10, Maximum value: 255

Example:

```
set bot settings -trapURLAutoGenerate ON -trapURLInterval 300 -trapURLLength
 60
```

Configure bot trap URL by using the GUI

1. Navigate to **Security > Citrix Bot Management > Profiles**.
2. In the **Citrix Bot Management Profiles** page, click **Edit** to configure the bot trap URL technique.
3. In the **Create Citrix Bot Management Profile** page, enter the bot trap URL in the general section.



4. In the **Create Citrix Bot Management Profile** page, click **Bot Trap** from **Profile Settings**.
5. In the **Bot Trap** section, set the following parameters.

   a. Enabled. Select the check box to enable bot trap detection
   b. Description. Brief description about the URL.
   c. Configure Actions. Action to be taken for bot detected by bot trap access.



6. In the **Configure Trap Insertion URLs** section, click **Add**.
7. In the **Configure Citrix Bot Management Profile Bot Trap Binding** page, set the following parameters.

a) Trap URL. Type the URL that you want to confirm as the bot trap Injection URL.
b) Enabled. Enable or disable bot trap injection URL.
c) Comment. A brief description about the trap injection URL.



8. In the **Signature Settings** section, click **Bot Trap**.
9. In the **Bot Trap** section, set the following parameters:
   a) Enabled. Select the check box to enable bot trap detection.
   b) In the Configure section, set the following parameters.
      i. Action. Action to be taken for bot detected by bot trap access.
      ii. Log. Enable or disable logging for bot trap binding.
10. Click **Update** and **Done**.

Configure bot trap URL settings

Complete the following steps to configure bot trap URL settings:

1. Navigate to **Security > Citrix Bot Management**.
2. In the details pane, under **Settings** click **Change Citrix Bot Management Settings**.
3. In the **Configure Citrix Bot Management Settings**, set the following parameters.
   a) Trap URL Interval. Time in seconds after which the bot trap URL is updated.
   b) Trap URL Length. Length of the auto-generated bot trap URL.
4. Click **OK** and **Done**.



Client IP policy expression for bot detection

The Citrix bot management now enables you to configure an advanced policy expression to extract the client IP address from an HTTP request header, HTTP request body, HTTP request URL, or using an advanced policy expression. The extracted value can be used by a bot detection mechanism (such as TPS, bot trap, or rate limit) to detect if the incoming request is a bot.

> **Note:**
>
> If you have not configured a client IP expression, the default or existing source client IP address is used for bot detection. If an expression is configured, then the evaluation result provides the client IP address that can be used for bot detection.

You can configure and use the client IP expression to extract the actual client IP address if the incoming request is coming through a proxy server and if the client IP address is present in the header. By adding this configuration, the appliance can use the bot detection mechanism in providing more security to software clients and servers.

Configure client IP policy expression in bot profile by using the CLI

At the command prompt, type:


```
add bot profile <name> [-clientIPExpression <expression>]
<!--NeedCopy-->
```

Example:

```
add bot profile profile1 -clientIPExpression 'HTTP.REQ.HEADER("X-Forwarded-For")ALT CLIENT.IP.SRC.TYPECAST_TEXT_T'
```

Configure client IP policy expression in bot profile by using the GUI

1. Navigate to **Security > Citrix Bot Management > Profiles**.
2. In the details pane, click **Add**.
3. In the **Create Citrix Bot Management Profile** page, set the Client IP Expression.
4. Click **Create** and **Close**.



Configure CAPTCHA for IP reputation and device fingerprint detection

CAPTCHA is an acronym that stands for “Completely Automated Public Turing test to tell Computers and Humans Apart”. CAPTCHA is designed to test if an incoming traffic is from a human user or an automated bot. CAPTCHA helps to block automated bots that cause security violations to web applications. In the ADC appliance, CAPTCHA uses the challenge-response module to identify if the incoming traffic is from a human user and not an automated bot.

Configure bot static signatures

This detection technique enables you to identify the user agent info from the browser details. Based on user agent information, the bot is identified as a bad or a good bot and then you assign a bot action to it. Follow the steps below to configure the static signature technique:

1. On the navigation pane, expand **Security > Citrix Bot Management > Signatures**.
2. On the **Citrix Bot Management Signatures** page, select a signature file and click **Edit**.
3. On the **Citrix Bot Management Signature** page, go to the **Signature Settings** section and click the **Bot Signatures**.
4. In the **Bot Signatures** section, set the following parameters:
   a) Configure Static Signatures. This section has a list of bot static signature records. You can select a record and click **Edit** to assign a bot action to it.
   b) Click **OK**.
5. Click **Update Signature**.
6. Click **Done**.



Bot static signature delineation

Citrix ADC bot management protects your web application against bots. Bot static signatures help in identifying good and bad bots based on request parameters such as user-agent in the incoming request.

The list of signatures in the file is huge and also new rules get added and stale ones are removed periodically. As an administrator, you might want to search for a specific signature or list of signatures under a category. To filter signatures easily, the bot signature page provides an enhanced search capability. The search function enables you to find signature rules and configure its property based on one or more signature parameters like action, signature ID, developer and signature name.

Action. Select a bot action that you prefer to configure for a specific category of signature rules. Following are the available action types:

- Enable Selected. Enable all the selected signature rules.
- Disable Selected. Disable all the selected signatures rules.
- Drop Selected. Apply the action as “Drop” to all the selected signature rules.
- Redirect Selected. Apply action as “Redirect” to all the selected signature rules.
- Reset Selected. Apply action as “Reset” to all the selected signature rules.
- Log Selected. Apply action as “Log” to all the selected signature rules.
- Remove Drop Selected. Unset the drop action to all the selected signature rules.
- Remove Redirect Selected. Unset the redirect action to all the selected signature rules.
- Remove Reset Selected. Unset the reset action to all the selected signature rules.
- Remove Log Selected. Unset the log action to all the selected signature rules.

Category. Select a category to filter signature rules accordingly. Following is the list of categories that are available for sorting signature rules.

- Action. Sort based on bot action.
- Category. Sort based on bot category.
- Developer. Sort based on the host company publisher.
- Enabled. Sort based on signature rules that are enabled.
- Id. Sort based on signature rule ID.
- Log. Sort based on signature rules that have logging enabled.
- Name. Sort based on signature rule name.
- Type. Sort based on signature type.
- Version. Sort based on signature rule version.

Search bot static signature rules based on action and category types by using the Citrix ADC GUI

1. Navigate to **Security > Citrix Bot Management > Signature**.
2. In the details page, click **Add**.
3. In the **Citrix Bot Management Signatures** page, click edit in the **Static Signature** section.
4. In the **Configure Static Signature** section, select a signature action from the drop-down list.
5. Use the search function to select a category and filter the rules accordingly.
6. Click **Update**.



Edit bot static signature rule property by using the Citrix ADC GUI

1. Navigate to **Security > Citrix Bot Management > Signature**.
2. In the details page, click **Add**.
3. In the **Citrix Bot Management Signatures** page, click edit in the **Static Signature** section.
4. In the **Configure Static Signature** section, select an action from the drop-down list.
5. Use the search function to select a category and filter rules accordingly.
6. From the static signature list, select a signature to modify its property.



7. Click **OK** to confirm.

How CAPTCHA works in Citrix ADC bot management

In Citrix ADC bot management, CAPTCHA validation is configured as a policy action to be run after bot policy is evaluated. The CAPTCHA action is available only for IP reputation and device fingerprint detection techniques. Following are the steps to understand how CAPTCHA works:

1. If a security violation is observed during IP reputation or device fingerprint bot detection, the ADC appliance sends a CAPTCHA challenge.
2. The client sends the CAPTCHA response.
3. The appliance validates the CAPTCHA response and if the CAPTCHA is valid, the request is allowed and it is forwarded to the back-end server.
4. If the CATCHA response is invalid, the appliance sends a new CAPTCHA challenge until the maximum number of attempts is reached.
5. If the CAPTCHA response is invalid even after the maximum number of attempts, the appliance drops or redirects the request to the configured error URL.
6. If you have configured log action, then the appliance stores the request details in the ns.log file.

Configure CAPTCHA settings by using the Citrix ADC GUI

The bot management CAPTCHA action is supported only for IP reputation and device fingerprint detection techniques. Complete the following steps to configure the **CAPTCHA** settings.

1. Navigate to **Security > Citrix Bot Management and Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a profile and click **Edit**.
3. On the **Citrix Bot Management Profile** page, go to the **Signature Settings** section and click **CAPTCHA**.
4. In the **CAPTCHA Settings** section, click **Add to configure CAPTCHA** settings to the profile:
5. In the **Configure Citrix Bot Management CAPTCHA** page, set the following parameters.
    a) URL. Bot URL for which the CAPTCHA action is applied during IP reputation and device fingerprint detection techniques.
    b) Enabled. Set this option to enable CAPTCHA support.
    c) Grace time. Duration until when no new CAPTCHA challenge is sent after the current valid CAPTCHA response is received.
    d) Wait time. Duration taken for the ADC appliance to wait until the client sends the CAPTCHA response.
    e) Mute Period. Duration for which the client which sent an incorrect CAPTCHA response must wait until allowed to try next. During this mute period, the ADC appliance does not allow any requests. Range: 60–900 seconds, Recommended: 300 seconds

f) Request Length limit. Length of the request for which the CAPTCHA challenge is sent to the client. If the length is greater than the threshold value, the request is dropped. Default value is 10–3000 bytes.

g) Retry Attempts. Number of attempts the client is allowed to retry to solve the CAPTCHA challenge. Range: 1–10, Recommended: 5.

h) No Action/Drop/Redirect action to be taken if the client fails the CAPTCHA validation.

i) Log. Set this option to store request information from the client when response CAPTCHA fails. The data is store in `ns.log` file.

j) Comment. A brief description about the CAPTCHA configuration.

6. Click **OK** and **Done**.



7. Navigate to **Security > Citrix Bot Management > Signatures**.
8. On the **Citrix Bot Management Signatures** page, select a signature file and click **Edit**.
9. On the **Citrix Bot Management Signature** page, go to **Signature Settings** section and click **Bot Signatures**.
10. In the **Bot Signatures** section, set the following parameters:

11. Configure **Static Signatures**. Select a bot static signature record and click Edit to assign a bot action to it.
12. Click **OK**.
13. Click **Update Signature**.
14. Click **Done**.



Auto update for bot signatures

The bot static signature technique uses a signature lookup table with a list of good bots and bad bots. The bots are categorized based on user-agent string and domain names. If the user-agent string and domain name in incoming bot traffic matches a value in the lookup table, a configured bot action is applied.
The bot signature updates are hosted on the AWS cloud and the signature lookup table communicates with the AWS database for signature updates. The auto signature update scheduler runs every 1-hour to check the **AWS database** and updates the signature table in the Citrix ADC appliance.

The signature auto update URL to configure is, `https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

> **Note:**
>
> You can also configure a proxy server and periodically update signatures from the AWS cloud to the appliance through the proxy. For proxy configuration, you must set the proxy IP address and port address in the bot settings.

How bot signature auto update works

The following diagram shows how the bot signatures are retrieved from the AWS cloud, updated on Citrix ADC, and viewed on Citrix ADM for signature update summary.

The bot signature auto-update scheduler does the following:

1. Retrieves the mapping file from the AWS URI.
2. Checks the latest signatures in the mapping file with the existing signatures in the ADC appliance.
3. Downloads the new signatures from AWS and verifies the signature integrity.
4. Updates the existing bot signatures with the new signatures in the bot signature file.
5. Generates an SNMP alert and sends the signature update summary to Citrix ADM.

Configure bot signature auto update

For configuring bot signature auto update, complete the following steps:

Enable bot signature auto update

You must enable the auto update option in the bot settings on the ADC appliance.
At the command prompt, type:

```
set bot settings -signatureAutoUpdate ON
```

Configure proxy server settings (optional)

If you are accessing the AWS signature database through a proxy server, you must configure the proxy server and port.

```
set bot settings -proxyserver -proxyport
```

Example:

```
set bot settings -proxy server 1.1.1.1 -proxyport 1356
```


Configure bot signature auto update using the Citrix ADC GUI

Complete the following steps to configure bot signature auto update:

1. Navigate to **Security > Citrix Bot Management**.
2. In the details pane, under **Settings** click **Change Citrix Bot Management Settings**.
3. In the **Configure Citrix Bot Management Settings**, select the **Auto Update Signature** check box.



4. Click **OK** and **Close**.

Create bot management profile

A bot profile is a collection of bot management settings that are used for detecting the bot type. In a profile, you determine how the Web App Firewall applies each of its filters (or checks) to bot traffic to your websites, and responses from them.

Complete the following steps to configure the bot profile:

1. Navigate to **Security** > **Citrix Bot Management** > **Profiles**.
2. In the details pane, click **Add**.
3. In the **Create Citrix Bot Management Profile** page, set the following parameters.

a) Name. Bot profile name.
b) Signature. Name of the bot signature file.
c) Error URL. URL for redirects.
d) Comment. Brief description about the profile.

4. Click **Create** and **Close**.



Create bot policy

The bot policy controls the traffic going to the bot management system and also to control the bot logs sent to the auditlog server. Follow the procedure to configure the bot policy.

1. Navigate to **Security** > **Citrix Bot Management** > **Bot Policies**.
2. In the details pane, click **Add**.
3. In the **Create Citrix Bot Management Policy** page, set the following parameters.
   a) Name. Name of the Bot policy.
   b) Expression. Type the policy expression or rule directly in the text area.
   c) Bot Profile. Bot profile to apply the bot policy.
   d) Undefined Action. Select an action that you prefer to assign.
   e) Comment. Brief description about the policy.
   f) Log Action. Audit log message action for logging bot traffic. For more information about audit log action, see Audit logging topic.

4. Click **Create** and **Close**.



Bot Transactions Per second (TPS)

The Transactions Per Second (TPS) bot technique detects incoming traffic as a bot if the number of requests per second (RPS) and percentage increase in RPS exceeds the configured threshold value. The detection technique protects your web applications from automated bots that can cause web scraping activities, brute forcing login, and other malicious attacks.

> **Note:**
>
> The bot technique detects an incoming traffic as bot only if both the parameters are configured

> and if both values increase beyond the threshold limit.
> Let us consider a scenario, where the appliance receives many requests coming from a specific URL and you want the Citrix ADC bot management to detect if there is a bot attack. The TPS detection technique examines the number of requests (configured value) coming from the URL within 1 second and the percentage increase (configured value) in the number of requests received within 30 minutes. If the values exceed the threshold limit, the traffic is considered as bot and the appliance runs the configured action.

Configure bot transactions per second (TPS) technique

To configure TPS, you must complete the following steps:

1. Enable bot TPS
2. Bind TPS settings to bot management profile

Bind TPS settings to bot management profile

Once you enable the bot TPS feature, you must bind the TPS settings to the bot management profile.

At the command prompt, type:

```
bind bot profile <name>… (-tps [-type ( SourceIP | GeoLocation | RequestURL
 | Host )] [-threshold <positive_integer>] [-percentage <positive_integer
>] [-action ( none | log | drop | redirect | reset | mitigation )] [-
logMessage <string>])
```

Example:

```
bind bot profile profile1 -tps -type RequestURL -threshold 1 -percentage
100000 -action drop -logMessage log
```

Enable bot transaction per second (TPS)

Before you can begin, you must ensure that the Bot TPS feature is enabled on the appliance. At the command prompt, type:

```
set bot profile profile1 -enableTPS ON
```

Configure bot transactions per second (TPS) by using the Citrix ADC GUI

Complete the following steps to configure bot transactions per second:

1. Navigate to **Security > Citrix Bot Management > Profiles**.
2. In the **Citrix Bot Management Profiles** page, select a profile and click **Edit**.

3. In the **Create Citrix Bot Management Profile** page, click **TPS** under **Signature Settings** section.
4. In the **TPS** section, enable the feature and click **Add**.



5. In **Configure Citrix Bot Management Profile TPS Binding** page, set the following parameters.
   a) Type. Input types allowed by the detection technique. Possible values: SOURCE IP, GEOLOCATION, HOST, URL.

      SOURCE_IP – TPS based on client IP address.

      GEOLOCATION – TPS based on the client's geographic location.

      HOST - TPS based on client requests forwarded to a specific back-end server IP address.

      URL – TPS based on client requests coming from a specific URL.

   b) Fixed Threshold. Maximum number of requests allowed from a TPS input type within 1 second time interval.

   c) Percentage Threshold. Maximum percentage increase in requests from a TPS input type within 30 minute time interval.

   d) Action. Action to be taken for bot detected by TPS binding.

   e) Log. Enable or disable logging for TPS binding.

   f) Log Message. Message to log for bot detected by TPS binding. Maximum Length: 255.

   g) Comments. A brief description about the TPS configuration. Maximum Length: 255
6. Click **OK** and then **Close**.



Bot detection based on mouse and keyboard dynamics

To detect bots and mitigate web scraping anomalies, the Citrix ADC bot management uses an enhanced bot detection technique based on mouse and keyboard behavior. Unlike conventional bot techniques that require direct human interaction (for example, CAPTCHA validation), the enhanced technique passively monitors the mouse and the keyboard dynamics. The Citrix ADC appliance then collects the real-time user data and analyses the behavioral between a human and a bot.

The passive bot detection using mouse and keyboard dynamics has the following benefits over existing bot detection mechanisms:

- Provides continuous monitoring throughout the user session, and eliminates single checkpoint.
- Requires no human interaction and it is completely transparent to users.

How bot detection using mouse and keyboard dynamics works

The bot detection technique using keyboard and mouse dynamics consists of two components, a webpage logger and bot detector. The webpage logger is a JavaScript that records keyboard and mouse movements when a user is performing a task on the webpage (for example, filling a registration form). The logger then sends the data in batches to the Citrix ADC appliance. The appliance then stores the data as a KM record and sends it to the bot detector on the Citrix ADM server, which analyses if the user is a human or bot.

The following steps explain how the components interact with each other:

1. The Citrix ADC admin configures policy expression through the ADM StyleBook, CLI, or NITRO or any other method.
2. The URL is set in the bot profile when the admin enables the feature on the appliance.
3. When a client sends a request, the Citrix ADC appliance tracks the session and all requests in the session.
4. The appliance inserts a JavaScript (webpage logger) in the response if the request matches the configured expression on the bot profile.
5. The JavaScript then collects all the keyboard, mouse activity and sends the KM data in a POST URL (transient).
6. The Citrix ADC appliance stores the data and sends it to the Citrix ADM server at the end of the session. Once the appliance receives the complete data of a POST request, the data is sent it to ADM server.
7. The Citrix ADM Service analyses the data and based on the analysis, the result is available on the Citrix ADM service GUI.

The JavaScript logger records the following mouse and keyboard movements:

- Keyboard events – all events
- Mouse events - mouse move, mouse up, mouse down
- Clipboard events - paste
- Custom events - autofill, autofillcancel
- timestamp of each event

Configure bot detection using mouse and keyboard dynamics

The Citrix ADC bot management configuration includes enabling or disabling keyboard and mouse-based detection feature, and configures the JavaScript URL in the bot profile.
Complete the following steps to configure bot detection using mouse and keyboard dynamics:

1. Enable keyboard and mouse-based detection
2. Configure expression to decide when the JavaScript can be injected in the HTTP response

Enable keyboard mouse-based bot detection

Before you begin the configuration, ensure you have enabled the keyboard and mouse-based bot detection feature on the appliance.

At the command prompt, type:

```
add bot profile <name> -KMDetection ( ON | OFF )
<!--NeedCopy-->
```

Example:


```
add bot profile profile1 -KMDetection ON
```

Configure bot expression for JavaScript insertion

Configure bot expression to evaluate the traffic and insert JavaScript. The JavaScript is inserted only if the expression is evaluated as true.

At the command prompt, type:

```
1 bind bot profile <name> -KMDetectionExpr -name <string> -expression <
     expression> -enabled ( ON | OFF ) - comment <string>
2 <!--NeedCopy-->
```

Example:

```
bind bot profile profile1 -KMDetectionExpr -name test -expression http.req.
url.startswith("/testsite")-enabled ON
```

Configure JavaScript file name inserted in the HTTP response for keyboard-mouse based bot detection

To collect the user action details, the appliance sends a JavaScript file name in the HTTP response. The JavaScript file collects all the data in a KM record and sends it to the appliance.

At the command prompt, type:

```
1 set bot profile profile1 - KMJavaScriptName <string>
2 <!--NeedCopy-->
```

Example:

```
set bot profile profile1 -KMJavaScriptName script1
```

Configure behavior biometrics size

You can configure the maximum size of mouse and keyboard behavior data that can be sent as KM record to the appliance and processed by ADM server.

At the command prompt, type:

```
1 set bot profile profile1 -KMEventsPostBodyLimit <positive_integer>
2 <!--NeedCopy-->
```

Example:

```
set bot profile profile1 - KMEventsPostBodyLimit 25
```


After you have configured the Citrix ADC appliance to configure the JavaScript and collect keyboard and mouse behavior biometrics, the appliance sends the data to the Citrix ADM server. For more information on how the Citrix ADM server detects bots from behavior biometrics, see Bot Violations topic.

Configure keyboard and mouse bot expression settings by using the GUI

1. Navigate to **Security > Citrix Bot Management and Profiles**.
2. On the **Citrix Bot Management Profiles** page, select a profile and click **Edit**.
3. In the **Keyboard and mouse based bot detection** section, set the following parameters:
   a) Enable detection. Select the check box to detect the bot based keyboard and mouse dynamics behavior.
   b) Event post body limit. Size of the keyboard and mouse dynamics data sent by the browser to be processed by the Citrix ADC appliance.
4. Click **OK**.



5. On the **Citrix Bot Management Profile** page, go to the **Profile Settings** section and click **Keyboard and Mouse Based Bot Expression Settings**.
6. In the **Keyboard and Mouse Based Bot Expression Settings** section, click **Add**.
7. In the **Configure Citrix Bot Management Profile Bot Keyboard and Mouse Expression Binding** page, set the following parameters:
   a) Expression Name. Name of the bot policy expression for detection keyboard and mouse dynamics.
   b) Expression. Bot policy expression.
   c) Enabled. Select the check box to enable the keyboard and bot keyboard and mouse expression binding.
   d) Comments. A brief description about the bot policy expression and its binding to the bot profile.

e) Click **OK** and **Close**.

8. In the **Keyboard and Mouse Based Bot Expression Settings** section, Click **Update**.



Bot Management

September 14, 2021

Following are some of the troubleshooting scenarios covered in Citrix ADC bot management.

1. How to handle false positive cases?

   You can use the bot allow list functionality to manage false positive cases and these transactions can be bypassed.

2. How to find more details about bad bot traffic?

   You can use the audit logging functionality to get details about the traffic classified as bad bots.

3. Why should you change the default signature name?

   You can change the default signature name if there are conflicts detected at the end point resources served by the Citrix ADC appliance.

Bot Management

September 21, 2021

What is Citrix ADC bot management?


```
Citrix ADC bot management detects and distinguishes traffic from good
    bots, bad bots, and human clients. The bot management functionality
    protects your web applications from bad bots by applying a
    configured action on incoming requests.
```

1. Why Citrix ADC must manage bots for your web application?

   Malicious bots constitute 30% of your internet traffic. Malicious bots impact web applications in various ways such as initiating a DoS attack, spamming email addresses, slowing down the application using downloader programs, downloading the content from websites and so forth. In addition, bots can easily bypass some of the well known detection mechanisms leading to loss of data, revenue, and reputation to your organization.

2. What are the techniques used for detecting an incoming bot?

   The appliance uses detection techniques such as IP reputation, rate limiting, device fingerprinting, TPS, and Bot trap detection techniques. In addition, you can configure a customized block list on the Citrix ADC GUI to categorize organization specific bad bots.

3. What is a bot signature file and its purpose?

   The bot signature file contains the footprint of known good and bad bots. The signature file is updated periodically to include the latest bot signatures for better bot protection.

4. What type of Citrix ADC license must I purchase?

   Bot management is available with the ADC Premium license.

5. Where can I find bot logs for troubleshooting?

   Citrix ADC audit logs provide detected bot details. For more information, see Audit Logging topic.

6. Is there an auto update functionality for the bot signature files?

   Yes, Citrix ADC bot management supports auto update functionality.

7. Is there a pre-requisite for using the bot IP reputation technique?

   Enable the IP reputation feature before enabling and configuring the IP reputation in the bot profile.

Bot signature alert Articles

September 14, 2021

Citrix bot management announces signature updates that you can download and apply on your appliance. When you detect a bot attack, you will receive an email notification about the new signature update. You can download the signature and apply it on your appliance.

Bot signature update for November 2020

September 14, 2021

New signatures rules are generated for the bots identified in the week 2020-11-11. You can download and configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 5 applicable for Citrix ADC 13.0 platform.

New Bots Insight

Following is a list of bot signature rules, category, and its type.

Category	Bot Type	Signature Count
Scraper	Good Bot	3
Marketing	Good Bot	23
Feed Fetcher	Good Bot	2
Tool	Bad Bot	3
Search Engine	Good Bot	34
Crawler	Good Bot	6
Uncategorized	Bad Bot	6
Virus Scanner	Good Bot	1
Screenshot Creator	Good Bot	7
Scraper	Bad Bot	1
Tool	Good Bot	7

Bot signature update for January 2021

September 20, 2021

Some of existing bot signatures are updated. You can download and configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 6 is applicable for Citrix ADC platforms with 13.0 61.x builds or later.

Updated bot signatures in this version

Following is a list of bot signature rule IDs, category and its type.

Bot signature rule	ID	Description
143	Crawler	Good Bot
561	Scraper	Good Bot
857	Site Monitor	Good Bot
892	Site Monitor	Bad Bot
894	Site Monitor	Bad Bot
980	Scraper	Bad Bot
1025	Site Monitor	Bad Bot
1029	Feed Fetcher	Bad Bot
1030	Screenshot Creator	Bad Bot
1034	Tool	Bad Bot
1039	Marketing	Bad Bot
1042	Site Monitor	Bad Bot
1047	Site Monitor	Bad Bot
1053	Site Monitor	Bad Bot
1072	Search Engine	Bad Bot
1073	Feed Fetcher	Bad Bot
1074	Uncategorized	Bad Bot
1078	Screenshot Creator	Bad Bot

Bot signature rule	ID	Description
1109	Marketing	Bad Bot
1132	Feed Fetcher	Bad Bot
1138	Marketing	Bad Bot
1150	Search Engine	Bad Bot
1164	Search Engine	Bad Bot
1167	Marketing	Bad Bot
1173	Tool	Bad Bot
1174	Marketing	Bad Bot
1176	Search Engine	Bad Bot
1178	Speed Tester	Bad Bot
1185	Screenshot Creator	Bad Bot
1209	Uncategorized	Bad Bot
1244	Site Monitor	Bad Bot
1251	Search Engine	Bad Bot
1254	Site Monitor	Bad Bot
1256	Uncategorized	Bad Bot
1259	Tool	Bad Bot
1287	Search Engine	Bad Bot
1296	Search Engine	Bad Bot
1312	Uncategorized	Bad Bot
1316	Marketing	Bad Bot
1322	Site Monitor	Bad Bot
1325	Screenshot Creator	Bad Bot
1328	Search Engine	Bad Bot
1330	Marketing	Bad Bot
1337	Tool	Bad Bot
1360	Search Engine	Bad Bot
1367	Search Engine	Bad Bot
1374	Tool	Bad Bot

Bot signature rule	ID	Description
1380	Uncategorized	Bad Bot
1388	Search Engine	Bad Bot
1400	Feed Fetcher	Bad Bot
1413	Uncategorized	Bad Bot
1420	Feed Fetcher	Bad Bot
1422	Site Monitor	Bad Bot
1442	Uncategorized	Bad Bot
1447	Search Engine	Bad Bot
1460	Marketing	Bad Bot
1467	Tool	Bad Bot
1469	Tool	Bad Bot
1471	Search Engine	Bad Bot
1484	Uncategorized	Bad Bot
1493	Marketing	Bad Bot
1502	Site Monitor	Bad Bot
1504	Uncategorized	Bad Bot
1506	Uncategorized	Bad Bot
1518	Uncategorized	Bad Bot
1520	Search Engine	Bad Bot
1531	Feed Fetcher	Bad Bot
1533	Uncategorized	Bad Bot
1540	Search Engine	Bad Bot
1556	Marketing	Bad Bot
1560	Uncategorized	Bad Bot
1564	Tool	Bad Bot
1570	Site Monitor	Bad Bot
1575	Search Engine	Bad Bot
1586	Virus Scanner	Bad Bot
1588	Uncategorized	Bad Bot

Bot signature rule	ID	Description
1594	Tool	Bad Bot
1619	Marketing	Bad Bot
1623	Tool	Bad Bot
1626	Search Engine	Bad Bot
1632	Feed Fetcher	Bad Bot
1648	Search Engine	Bad Bot
1652	Marketing	Bad Bot
1660	Marketing	Bad Bot
1713	Tool	Bad Bot
1719	Search Engine	Bad Bot
1722	Uncategorized	Bad Bot
1744	Uncategorized	Bad Bot
1754	Uncategorized	Bad Bot
1757	Uncategorized	Bad Bot
1762	Uncategorized	Bad Bot
1769	Uncategorized	Bad Bot
1771	Marketing	Bad Bot
1779	Tool	Bad Bot
1782	Tool	Bad Bot
1785	Speed Tester	Bad Bot
1786	Tool	Bad Bot
1792	Site Monitor	Bad Bot
1869	Tool	Bad Bot
1928	Marketing	Bad Bot
1942	Site Monitor	Bad Bot
1949	Marketing	Bad Bot
1954	Marketing	Bad Bot
1964	Uncategorized	Bad Bot
1969	Search Engine	Bad Bot

Bot signature rule	ID	Description
2294	Search Engine	Bad Bot
2303	Uncategorized	Bad Bot
2308	Scraper	Bad Bot
2335	Marketing	Bad Bot
2374	Uncategorized	Bad Bot
2377	Uncategorized	Bad Bot
2385	Tool	Bad Bot
2389	Uncategorized	Bad Bot
2414	Uncategorized	Bad Bot
2421	Uncategorized	Bad Bot
2424	Uncategorized	Bad Bot
2427	Uncategorized	Bad Bot
2429	Search Engine	Bad Bot
2437	Uncategorized	Bad Bot
2440	Search Engine	Bad Bot
2443	Uncategorized	Bad Bot
2453	Marketing	Bad Bot
2472	Marketing	Bad Bot
2474	Feed Fetcher	Bad Bot
2482	Uncategorized	Bad Bot
2500	Screenshot Creator	Bad Bot
2503	Uncategorized	Bad Bot
2507	Uncategorized	Bad Bot
2516	Tool	Bad Bot
2536	Marketing	Bad Bot
2543	Tool	Bad Bot
2548	Tool	Bad Bot
2557	Marketing	Bad Bot
2561	Uncategorized	Bad Bot

Bot signature rule	ID	Description
2572	Uncategorized	Bad Bot
2578	Uncategorized	Bad Bot
2584	Uncategorized	Bad Bot
2588	Uncategorized	Bad Bot
2592	Search Engine	Bad Bot
2600	Tool	Bad Bot
2606	Uncategorized	Bad Bot
2611	Uncategorized	Bad Bot
2622	Tool	Bad Bot
2625	Tool	Bad Bot
2631	Tool	Bad Bot
2635	Tool	Bad Bot
2637	Screenshot Creator	Bad Bot
2641	Search Engine	Bad Bot
2655	Uncategorized	Bad Bot
2657	Marketing	Bad Bot
2663	Uncategorized	Bad Bot
2666	Tool	Bad Bot
2672	Feed Fetcher	Bad Bot
2674	Tool	Bad Bot
2681	Search Engine	Bad Bot
2684	Marketing	Bad Bot
2690	Uncategorized	Bad Bot
2704	Uncategorized	Bad Bot
2707	Uncategorized	Bad Bot
2714	Feed Fetcher	Bad Bot
2722	Uncategorized	Bad Bot
2726	Feed Fetcher	Bad Bot
2730	Screenshot Creator	Bad Bot

Bot signature rule	ID	Description
2736	Uncategorized	Bad Bot
2749	Uncategorized	Bad Bot
2753	Tool	Bad Bot
2756	Tool	Bad Bot
2760	Speed Tester	Bad Bot
2780	Tool	Bad Bot
2785	Site Monitor	Bad Bot
2789	Uncategorized	Bad Bot
2797	Tool	Bad Bot
2801	Tool	Bad Bot
2808	Tool	Bad Bot
2810	Uncategorized	Bad Bot
2813	Uncategorized	Bad Bot
2816	Uncategorized	Bad Bot
2820	Link Checker	Bad Bot
2824	Link Checker	Bad Bot
2831	Screenshot Creator	Bad Bot
2843	Tool	Bad Bot
2846	Tool	Bad Bot
2849	Marketing	Bad Bot
2851	Uncategorized	Bad Bot
2855	Uncategorized	Bad Bot
2859	Tool	Bad Bot
2873	Uncategorized	Bad Bot
2875	Screenshot Creator	Bad Bot
2879	Uncategorized	Bad Bot
2881	Uncategorized	Bad Bot
2886	Site Monitor	Bad Bot
2899	Uncategorized	Bad Bot

Bot signature rule	ID	Description
2916	Uncategorized	Bad Bot
2924	Tool	Bad Bot
2932	Marketing	Bad Bot
2935	Link Checker	Bad Bot
2939	Marketing	Bad Bot
2942	Uncategorized	Bad Bot
2955	Search Engine	Bad Bot
2960	Tool	Bad Bot
2964	Uncategorized	Bad Bot
2972	Marketing	Bad Bot
2978	Vulnerability Scanner	Bad Bot
2980	Tool	Bad Bot
2985	Marketing	Bad Bot
2993	Uncategorized	Bad Bot
2999	Screenshot Creator	Bad Bot
3003	Feed Fetcher	Bad Bot
3005	Uncategorized	Bad Bot
3013	Uncategorized	Bad Bot
3016	Uncategorized	Bad Bot
3021	Search Engine	Bad Bot
3026	Uncategorized	Bad Bot
3030	Marketing	Bad Bot
3065	Marketing	Bad Bot
3068	Uncategorized	Bad Bot
3072	Marketing	Bad Bot
3077	Marketing	Bad Bot
3080	Uncategorized	Bad Bot
3086	Scraper	Bad Bot
3092	Search Engine	Bad Bot

Bot signature rule	ID	Description
3100	Uncategorized	Bad Bot
3104	Tool	Bad Bot
3111	Uncategorized	Bad Bot
3116	Site Monitor	Bad Bot
3118	Tool	Bad Bot
3120	Marketing	Bad Bot
3122	Search Engine	Bad Bot
3126	Marketing	Bad Bot
3141	Tool	Bad Bot
3143	Uncategorized	Bad Bot
3145	Scraper	Bad Bot
3150	Uncategorized	Bad Bot
3173	Link Checker	Bad Bot
3176	Uncategorized	Bad Bot
3186	Speed Tester	Bad Bot
3190	Scraper	Bad Bot
3203	Search Engine	Bad Bot
3216	Uncategorized	Bad Bot
3220	Tool	Bad Bot
3223	Link Checker	Bad Bot
3241	Uncategorized	Bad Bot
3245	Site Monitor	Bad Bot
3285	Uncategorized	Bad Bot
3304	Marketing	Bad Bot
3307	Link Checker	Bad Bot
3316	Tool	Bad Bot
3326	Marketing	Bad Bot
3333	Search Engine	Bad Bot
3340	Search Engine	Bad Bot

Bot signature rule	ID	Description
3344	Marketing	Bad Bot
3350	Uncategorized	Bad Bot
3355	Marketing	Bad Bot
3365	Uncategorized	Bad Bot
3378	Uncategorized	Bad Bot
3388	Tool	Bad Bot
3396	Uncategorized	Bad Bot
3400	Uncategorized	Bad Bot
3421	Uncategorized	Bad Bot
3439	Uncategorized	Bad Bot
3447	Feed Fetcher	Bad Bot
3451	Tool	Bad Bot
3459	Screenshot Creator	Bad Bot
3469	Vulnerability Scanner	Bad Bot
3475	Uncategorized	Bad Bot
3485	Search Engine	Bad Bot
3493	Tool	Bad Bot
3502	Marketing	Bad Bot
3507	Search Engine	Bad Bot
3523	Uncategorized	Bad Bot
3535	Speed Tester	Bad Bot
3549	Uncategorized	Bad Bot
3556	Uncategorized	Bad Bot
3561	Uncategorized	Bad Bot
3565	Uncategorized	Bad Bot
3572	Search Engine	Bad Bot
3578	Uncategorized	Bad Bot
3610	Search Engine	Bad Bot
3617	Uncategorized	Bad Bot

Bot signature rule	ID	Description
3621	Marketing	Bad Bot
3632	Tool	Bad Bot
3635	Marketing	Bad Bot
3653	Uncategorized	Bad Bot
3661	Search Engine	Bad Bot
3704	Uncategorized	Bad Bot
3707	Uncategorized	Bad Bot
3711	Uncategorized	Bad Bot
3730	Search Engine	Bad Bot
3740	Site Monitor	Bad Bot
3759	Search Engine	Bad Bot
3764	Uncategorized	Bad Bot
3770	Uncategorized	Bad Bot

Bot signature update for March 2021

September 20, 2021

Some of existing bot signatures are updated. You can download and configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 7 is applicable for Citrix ADC platforms with 13.0 61.x builds or later.

Updated bot signatures in this version

Following is a list of bot signature rule IDs, category, and its type.

Bot signature rule	ID	Description
278	Scraper	Good Bot

Bot signature rule	ID	Description
378	Scraper	Good Bot
379	Scraper	Good Bot
380	Scraper	Good Bot
381	Scraper	Good Bot
382	Scraper	Good Bot
383	Scraper	Good Bot
384	Scraper	Good Bot
385	Scraper	Good Bot
386	Scraper	Good Bot
387	Scraper	Good Bot
389	Scraper	Good Bot
390	Scraper	Good Bot
391	Scraper	Good Bot
494	Scraper	Good Bot
627	Search Engine	Good Bot
660	Search Engine	Good Bot
3840	Crawler	Good Bot

Bot signature update for August 2021

September 20, 2021

New signatures are added and some of existing bot signatures are updated. You can download and configure these signature rules to protect your appliance from bot attacks.

Bot signature version

Signature version 8 is applicable for Citrix ADC platforms with 13.0 61.x builds or later.

Updated bot signatures in this version

Following is a list of bot signature rule IDs, category, and its type.

Bot signature rule	ID	Description
236	Scraper	Good Bot
378	Scraper	Good Bot
381	Scraper	Good Bot
382	Scraper	Good Bot
390	Scraper	Good Bot
544	Scraper	Good Bot
702	Search Engine	Good Bot
979	Scraper	Bad Bot
3791	Speed Tester	Good Bot
3797	Marketing	Good Bot
3800	Marketing	Good Bot
3824	Crawler	Bad Bot
3833	Search Engine	Good Bot
3849	Crawler	Good Bot
3871	Marketing	Good Bot
3963	Marketing	Good Bot
4027	Search Engine	Good Bot

New bot signatures in this version

Bot signature rule	ID	Description
4028	Marketing	Good Bot
4029	Tool	Good Bot
4030	Scraper	Good Bot
4031	Scraper	Good Bot
4032	Uncategorized	Bad Bot

Bot signature rule	ID	Description
4033	Crawler	Good Bot
4034	Crawler	Good Bot
4035	Marketing	Good Bot
4036	Vulnerability Scanner	Good Bot
4037	Vulnerability Scanner	Good Bot
4038	Uncategorized	Bad Bot
4039	Tool	Good Bot
4040	Crawler	Good Bot
4041	Tool	Good Bot
4042	Crawler	Good Bot
4043	Screenshot Creator	Good Bot
4044	Scraper	Bad Bot
4045	Scraper	Bad Bot
4046	Scraper	Bad Bot
4047	Uncategorized	Bad Bot
4048	Feed Fetcher	Good Bot
4049	Uncategorized	Bad Bot
4050	Crawler	Good Bot
4051	Crawler	Good Bot
4052	Tool	Good Bot
4053	Tool	Good Bot
4054	Scraper	Bad Bot
4055	Uncategorized	Good Bot
4056	Marketing	Good Bot
4057	Screenshot Creator	Good Bot
4058	Crawler	Good Bot
4059	Uncategorized	Bad Bot
4060	Search Engine	Good Bot
4061	Search Engine	Good Bot

Bot signature rule	ID	Description
4062	Search Engine	Good Bot
4063	Search Engine	Good Bot
4064	Tool	Good Bot
4065	Scraper	Good Bot
4066	Marketing	Good Bot
4067	Marketing	Good Bot
4068	Uncategorized	Bad Bot
4069	Uncategorized	Bad Bot
4070	Uncategorized	Bad Bot
4071	Tool	Good Bot
4072	Tool	Bad Bot
4073	Uncategorized	Bad Bot
4074	Uncategorized	Bad Bot
4075	Tool	Bad Bot
4076	Marketing	Good Bot
4077	Scraper	Good Bot
4078	Crawler	Good Bot
4079	Crawler	Good Bot
4080	Tool	Bad Bot
4081	Search Engine	Good Bot
4082	Tool	Good Bot
4083	Uncategorized	Bad Bot
4084	Uncategorized	Bad Bot
4085	Tool	Good Bot
4086	Tool	Good Bot
4087	Tool	Bad Bot
4088	Search Engine	Good Bot
4089	Marketing	Good Bot
4090	Tool	Good Bot

Bot signature rule	ID	Description
4091	Tool	Good Bot
4092	Tool	Good Bot
4093	Tool	Good Bot
4094	Uncategorized	Good Bot
4095	Site Monitor	Good Bot
4096	Site Monitor	Good Bot
4097	Site Monitor	Good Bot
4098	Crawler	Good Bot
4099	Search Engine	Good Bot
4100	Search Engine	Good Bot
4101	Search Engine	Good Bot
4102	Search Engine	Good Bot
4103	Marketing	Good Bot
4104	Marketing	Good Bot
4105	Marketing	Good Bot
4106	Marketing	Good Bot
4107	Marketing	Good Bot
4108	Marketing	Good Bot
4109	Search Engine	Good Bot
4110	Crawler	Good Bot
4111	Crawler	Good Bot
4112	Crawler	Good Bot
4113	Vulnerability Scanner	Good Bot
4114	Crawler	Good Bot
4115	Tool	Good Bot
4116	Uncategorized	Bad Bot
4117	Uncategorized	Bad Bot
4118	Uncategorized	Bad Bot
4119	Uncategorized	Bad Bot

Bot signature rule	ID	Description
4120	Marketing	Good Bot
4121	Marketing	Good Bot
4122	Marketing	Good Bot
4123	Marketing	Good Bot
4124	Marketing	Good Bot
4125	Marketing	Good Bot
4126	Marketing	Good Bot
4127	Marketing	Good Bot
4128	Marketing	Good Bot
4129	Marketing	Good Bot
4130	Marketing	Good Bot
4131	Tool	Good Bot
4132	Marketing	Good Bot
4133	Marketing	Good Bot
4134	Tool	Good Bot
4135	Marketing	Good Bot
4136	Marketing	Good Bot
4137	Marketing	Good Bot
4138	Marketing	Good Bot
4139	Marketing	Good Bot
4140	Marketing	Good Bot
4141	Marketing	Good Bot
4142	Marketing	Good Bot
4143	Marketing	Good Bot
4144	Marketing	Good Bot
4145	Search Engine	Good Bot
4146	Search Engine	Good Bot
4147	Search Engine	Good Bot
4148	Search Engine	Good Bot

Bot signature rule	ID	Description
4149	Search Engine	Good Bot
4150	Search Engine	Good Bot
4151	Search Engine	Good Bot
4152	Search Engine	Good Bot
4153	Search Engine	Good Bot
4154	Search Engine	Good Bot
4155	Search Engine	Good Bot
4156	Screenshot Creator	Good Bot
4157	Search Engine	Good Bot
4158	Search Engine	Good Bot
4159	Search Engine	Good Bot
4160	Screenshot Creator	Good Bot
4161	Search Engine	Good Bot
4162	Search Engine	Good Bot
4163	Tool	Good Bot
4164	Search Engine	Good Bot
4165	Marketing	Good Bot
4166	Uncategorized	Bad Bot
4167	Tool	Bad Bot
4168	Speed Tester	Good Bot
4169	Scraper	Bad Bot
4170	Tool	Good Bot
4171	Scraper	Bad Bot
4172	Web Crawler	Good Bot
4173	Tool	Good Bot
4174	Crawler	Good Bot
4175	Crawler	Good Bot
4176	Tool	Good Bot
4177	Search Engine	Good Bot

Bot signature rule	ID	Description
4178	Tool	Good Bot
4179	Web Crawler	Good Bot
4180	Tool	Good Bot
4181	Site Monitor	Good Bot
4182	Site Monitor	Good Bot
4183	Site Monitor	Good Bot
4184	Site Monitor	Good Bot
4185	Search Engine	Good Bot
4186	Tool	Good Bot
4187	Tool	Good Bot
4188	Screenshot Creator	Good Bot
4189	Marketing	Good Bot
4190	Search Engine	Good Bot
4191	Search Engine	Good Bot
4192	Search Engine	Good Bot
4193	Search Engine	Good Bot
4194	Tool	Good Bot
4195	Search Engine	Bad Bot
4196	Tool	Good Bot
4197	Tool	Good Bot
4198	Marketing	Good Bot
4199	Marketing	Good Bot
4200	Vulnerability Scanner	Good Bot
4201	Tool	Good Bot
4202	Tool	Good Bot
4203	Uncategorized	Bad Bot
4204	Uncategorized	Bad Bot
4205	Search Engine	Good Bot
4206	Marketing	Good Bot

Bot signature rule	ID	Description
4207	Marketing	Good Bot
4208	Search Engine	Good Bot
4209	Search Engine	Good Bot
4210	Speed Tester	Good Bot
4211	Tool	Good Bot
4212	Feed Fetcher	Good Bot
4213	Feed Fetcher	Good Bot
4214	Scraper	Bad Bot
4215	Tool	Good Bot
4216	Tool	Good Bot
4217	Tool	Bad Bot
4218	Scraper	Bad Bot
4219	Marketing	Good Bot
4220	Tool	Good Bot
4221	Tool	Bad Bot
4222	Site Monitor	Good Bot
4223	Marketing	Good Bot
4224	Search Engine	Good Bot
4225	Search Engine	Good Bot
4226	Search Engine	Good Bot
4227	Marketing	Good Bot
4228	Marketing	Good Bot
4229	Tool	Good Bot
4230	Uncategorized	Bad Bot
4231	Screenshot Creator	Good Bot
4232	Tool	Good Bot
4233	Site Monitor	Good Bot
4234	Site Monitor	Good Bot
4235	Site Monitor	Good Bot

Bot signature rule	ID	Description
4236	Site Monitor	Good Bot
4237	Site Monitor	Good Bot
4238	Site Monitor	Good Bot
4239	Uncategorized	Bad Bot
4240	Marketing	Good Bot
4241	Marketing	Good Bot
4242	Marketing	Good Bot
4243	Marketing	Good Bot
4244	Marketing	Good Bot
4245	Marketing	Good Bot
4246	Marketing	Good Bot
4247	Search Engine	Good Bot
4248	Search Engine	Good Bot
4249	Screenshot Creator	Good Bot
4250	Search Engine	Good Bot
4251	Search Engine	Good Bot
4252	Crawler	Good Bot
4253	Crawler	Good Bot
4254	Crawler	Good Bot
4255	Tool	Good Bot
4256	Uncategorized	Good Bot
4257	Tool	Good Bot
4258	Crawler	Good Bot
4259	Crawler	Good Bot
4260	Tool	Good Bot
4261	Tool	Good Bot
4262	Tool	Good Bot
4263	Marketing	Good Bot
4264	Crawler	Bad Bot

Bot signature rule	ID	Description
4265	Search Engine	Good Bot
4266	Uncategorized	Good Bot
4267	Tool	Good Bot
4268	Tool	Good Bot
4269	Search Engine	Good Bot
4270	Search Engine	Good Bot
4271	Search Engine	Good Bot
4272	Search Engine	Good Bot
4273	Search Engine	Good Bot
4274	Search Engine	Good Bot
4275	Search Engine	Good Bot
4276	Uncategorized	Bad Bot
4277	Uncategorized	Bad Bot
4278	Uncategorized	Bad Bot
4279	Marketing	Good Bot
4280	Crawler	Good Bot
4281	Uncategorized	Bad Bot
4282	Marketing	Good Bot
4283	Marketing	Good Bot
4284	Marketing	Good Bot
4285	Marketing	Good Bot
4286	Marketing	Good Bot
4287	Marketing	Good Bot
4288	Marketing	Good Bot
4289	Marketing	Good Bot
4290	Marketing	Good Bot
4291	Marketing	Good Bot
4292	Marketing	Good Bot
4293	Marketing	Good Bot

Bot signature rule	ID	Description
4294	Marketing	Good Bot
4295	Search Engine	Good Bot
4296	Search Engine	Good Bot
4297	Search Engine	Good Bot
4298	Search Engine	Good Bot
4299	Search Engine	Good Bot
4300	Search Engine	Good Bot
4301	Search Engine	Good Bot
4302	Search Engine	Good Bot
4303	Search Engine	Good Bot
4304	Search Engine	Good Bot
4305	Search Engine	Good Bot
4306	Screenshot Creator	Good Bot
4307	Search Engine	Good Bot
4308	Search Engine	Good Bot
4309	Search Engine	Good Bot
4310	Search Engine	Good Bot
4311	Screenshot Creator	Good Bot
4312	Search Engine	Good Bot
4313	Search Engine	Good Bot
4314	Search Engine	Good Bot
4315	Search Engine	Good Bot
4316	Search Engine	Good Bot
4317	Search Engine	Good Bot
4318	Screenshot Creator	Good Bot
4319	Screenshot Creator	Good Bot
4320	Uncategorized	Bad Bot
4321	Uncategorized	Good Bot
4322	Crawler	Good Bot

Bot signature rule	ID	Description
4323	Tool	Good Bot
4324	Tool	Good Bot
4325	Tool	Good Bot
4326	Scraper	Bad Bot
4327	Search Engine	Good Bot
4328	Marketing	Good Bot
4329	Uncategorized	Bad Bot
4330	Site Monitor	Good Bot
4331	Search Engine	Good Bot
4332	Search Engine	Good Bot
4333	Uncategorized	Bad Bot
4334	Scraper	Good Bot
4335	Marketing	Good Bot
4336	Marketing	Good Bot
4337	Tool	Good Bot
4338	Tool	Good Bot
4339	Tool	Good Bot
4340	Crawler	Good Bot
4341	Crawler	Good Bot
4342	Vulnerability Scanner	Good Bot
4343	Vulnerability Scanner	Good Bot
4344	Scraper	Good Bot
4345	Marketing	Good Bot
4346	Marketing	Good Bot
4347	Marketing	Good Bot
4348	Marketing	Good Bot
4349	Marketing	Good Bot
4350	Marketing	Good Bot
4351	Marketing	Good Bot

Bot signature rule	ID	Description
4352	Marketing	Good Bot
4353	Marketing	Good Bot
4354	Marketing	Good Bot
4355	Search Engine	Good Bot
4356	Search Engine	Good Bot
4357	Search Engine	Good Bot
4358	Search Engine	Good Bot
4359	Search Engine	Good Bot
4360	Search Engine	Good Bot
4361	Search Engine	Good Bot
4362	Search Engine	Good Bot
4363	Search Engine	Good Bot
4364	Search Engine	Good Bot
4365	Screenshot Creator	Good Bot
4366	Search Engine	Good Bot
4367	Search Engine	Good Bot
4368	Search Engine	Good Bot
4369	Search Engine	Good Bot
4370	Screenshot Creator	Good Bot
4371	Search Engine	Good Bot
4372	Search Engine	Good Bot
4373	Search Engine	Good Bot
4374	Search Engine	Good Bot
4375	Search Engine	Good Bot
4376	Screenshot Creator	Good Bot
4377	Crawler	Good Bot
4378	Crawler	Good Bot
4379	Search Engine	Good Bot
4380	Search Engine	Good Bot

Bot signature rule	ID	Description
4381	Search Engine	Good Bot
4382	Search Engine	Good Bot
4383	Crawler	Good Bot
4384	Search Engine	Good Bot
4385	Tool	Good Bot
4386	Uncategorized	Good Bot
4387	Crawler	Good Bot
4388	Crawler	Good Bot
4389	Tool	Good Bot
4390	Tool	Good Bot
4391	Tool	Good Bot
4392	Tool	Good Bot
4393	Tool	Good Bot
4394	Uncategorized	Good Bot
4395	Tool	Good Bot
4396	Site Monitor	Good Bot
4397	Site Monitor	Good Bot
4398	Tool	Bad Bot
4399	Tool	Bad Bot
4400	Tool	Bad Bot
4401	Tool	Bad Bot
4402	Tool	Bad Bot
4403	Tool	Bad Bot
4404	Search Engine	Good Bot
4405	Search Engine	Good Bot
4406	Search Engine	Good Bot
4407	Uncategorized	Good Bot

Cache Redirection

September 14, 2021

In a typical deployment, different clients ask web servers for the same content repeatedly. To relieve the origin web server of processing each request, a Citrix ADC appliance with cache redirection enabled can serve this content from a cache server instead of from the origin server.

The Citrix ADC appliance analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers.

Cache redirection is a policy-based feature. By default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. For testing or maintenance, you might want to skip policy evaluation and direct all requests to the cache or to the origin server.

You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

A Citrix ADC appliance configured for cache redirection can be deployed at the edge of a network, in front of the origin server, or anywhere along the network backbone. In an edge deployment, commonly used by Internet Service Providers (ISPs), cable companies, content delivery distribution networks, and enterprise networks, the Citrix ADC appliance resides directly in front of the clients. In a server-side deployment, the Citrix ADC appliance is closer to the origin servers.

Cache redirection is used most commonly with the HTTP service type, but it also supports the secure HTTPS protocol.

Cache redirection policies

September 14, 2021

A cache redirection virtual server applies cache redirection policies to each incoming request. By default, if a request matches one of the configured policies, it is considered non-cacheable, and the Citrix ADC appliance sends it to the origin server. Other requests are sent to a cache server. This behavior can be reversed, so that requests that match configured cache redirection policies are sent to cache servers.

The appliance provides a set of policies for cache redirection. If these built-in policies are not adequate for your deployment, you can configure user-defined cache redirection policies.

Note: Once you have determined which built-in cache redirection policies to use, or have created user-defined policies, proceed with configuring cache redirection. To use this feature, you must configure

at least one cache redirection virtual server, and, for normal operation, you must bind at least one cache redirection policy to that virtual server.

Built-in cache redirection policies

September 14, 2021

The Citrix ADC appliance provides built-in cache redirection policies that handle typical cache requests. These policies are based on HTTP methods, the URL or URL tokens of the incoming request, the HTTP version, or the HTTP headers and their values in the request.

Built-in cache redirection policies can be directly bound to a virtual server and do not need further configuration.

Cache redirection policies use two types of appliance expressions languages, classic and default syntax. For more information about these languages, see Policies and expressions.

Built-in classic cache redirection policies

Built-in cache redirection policies based on classic expressions are called *classic cache redirection policies*. For a complete description of classic expressions and how to configure them, see Policies and Expressions.

The classic cache redirection policies evaluate basic characteristics of traffic and other data. For example, classic cache redirection policies can determine whether an HTTP request or response contains a particular type of header or URL.

The Citrix ADC appliance provides the following built-in classic cache redirection policies:

Built-In Policy Name	Description
bypass-non-get	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, or if the HTTP request contains a pragma header.

Built-In Policy Name	Description
bypass-dynamic-url	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: cgi, asp, exe, cfm, ex, shtml, or htx. Also bypass the cache if the URL starts with any of the following: /cgi-bin/, /bin/, or /exec/.
bypass-urltokens	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie	Bypass the cache for any URL that has a cookie header and an extension other than .png or .jpg.

Built-in default syntax cache redirection policies

Built-in cache redirection policies based on default syntax expressions are called *default syntax cache redirection policies*. For a complete description of default syntax expressions and how to configure them, see Policies and Expressions.

In addition to the same types of evaluations done by classic cache redirection policies, default syntax cache redirection policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, directing the request to either cache or origin server).

Citrix ADC appliances provide the following two built-in actions for the default syntax cache redirection policies:

- CACHE
- ORIGIN

As implied by their names, they direct the request to the cache server or the origin server, respectively.

Note: If you are using the built-in default syntax cache redirection policy, you cannot modify the action.

The Citrix ADC appliance provides the following built-in default syntax cache redirection policies:

Built-In Policy Name	Description
bypass-non-get_adv	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control_adv	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, or if the HTTP request contains a pragma header.
bypass-dynamic-url_adv	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: cgi, asp, exe, cfm, ex, shtml, or htx. Also bypass the cache if the URL starts with any of the following: /cgi-bin/, /bin/, or /exec/.
bypass-urltokens_adv	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie_adv	Bypass the cache for any URL that has a cookie header and an extension other than .png or .jpg.

Display the built-in cache redirection policies

You can display the available cache redirection polices by using the command line interface or the configuration utility.

Display the built-in cache redirection policies by using the CLI

At the command prompt, type:

```
show cr policy [<policyName>]
```

Example:

```
1 > show cr policy
2 1)       Cache-By-Pass RULE: NS_NON_GET          Policy:bypass-non-get
3 2)       Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE ||
     NS_CACHECONTROL_NOCACHE || NS_HEADER_PRAGMA)    Policy:bypass-cache-
     control
4 3)       Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE ||
      NS_EXT_CFM || NS_EXT_EX || NS_EXT_SHTML || NS_EXT_HTX) || (
```



```
      NS_URL_PATH_CGIBIN || NS_URL_PATH_EXEC || NS_URL_PATH_BIN)
      Policy:bypass-dynamic-url
4)       Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-
   urltokens
5)       Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF &&
   NS_EXT_NOT_JPEG)     Policy:bypass-cookie
 Done
<!--NeedCopy-->
```

Display the built-in cache redirection policies by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Policies. The configured cache redirection policies appear in the details pane.
2. Select one of the configured policies to view details.

Configure a cache redirection policy

September 14, 2021

A cache redirection policy includes one or more expressions (also called *rules*). Each expression represents a condition that is evaluated when the client request is compared to the policy.

You do not explicitly configure actions for cache redirection policies. By default, the Citrix ADC appliance considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection policies based on the classic policy format are called *classic cache redirection policies*. Each such policy has a name and includes a classic expression or a set of classic expressions that are combined by using logical operators.

For classic cache redirection policies, you do not explicitly configure actions for the policies. By default, the Citrix ADC appliance considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection policies based on the newer policy format are called *Advanced redirection policies*. Such policy has a name and includes a default syntax expression, or a set of default syntax expressions that are combined by using logical operators, and the following built-in actions:

- CACHE
- ORIGIN

For more information about classic expressions and default syntax expressions, see Policies and Expressions.

Add a cache redirection policy by using the CLI

At the command prompt, type the following commands to add a cache redirection policy and verify the configuration:

```
-  add cr policy <policyName> **-rule** <expression>
-  show cr policy [<policyName>]
<!--NeedCopy-->
```

Examples:

Policy with a simple expression:

```
> add cr policy Policy-CRD-1 -rule "REQ.HTTP.URL != /*.jpeg"
 Done
> show cr policy Policy-CRD-1
         Cache-By-Pass RULE: REQ.HTTP.URL != '/*.jpeg'   Policy:Policy-
            CRD-1
 Done
<!--NeedCopy-->
```

Policy with a compound expression:

```
> add cr policy Policy-CRD-2 -rule "REQ.HTTP.METHOD == POST && (REQ.
   HTTP.URL == /*.cgi || REQ.HTTP.URL != /*.png)"
 Done
> show cr policy Policy-CRD-2
         Cache-By-Pass RULE: REQ.HTTP.METHOD == POST && (REQ.HTTP.URL
            == '/*.cgi' || REQ.HTTP.URL != '/*.png')   Policy:Policy-
            CRD-2
 Done
<!--NeedCopy-->
```

Policy that evaluates a header:

```
> add cr policy Policy-CRD-3 -rule "REQ.HTTP.HEADER If-Modified-Since
   EXISTS"
 Done
> show cr policy Policy-CRD-3
         Cache-By-Pass RULE: REQ.HTTP.HEADER If-Modified-Since EXISTS
               Policy:Policy-CRD-3
 Done
<!--NeedCopy-->
```


Add a default syntax cache redirection policy by using the CLI

At the command prompt, type the following commands to add a cache redirection policy and verify the configuration:

```
1 -  add cr policy <policyName> **-rule** <expression> [-action<string>]
     [-logAction<string>]
2 -  show cr policy [<policyName>]
3 <!--NeedCopy-->
```

Examples:

Policy with a simple expression:

```
1 > add cr policy crpol1 -rule !(HTTP.REQ.URL.ENDSWITH(".jpeg" )) -action
     origin
2  Done
3 > show cr policy crpoll
4          Policy: crpol1 Rule: !(HTTP.REQ.URL.ENDSWITH(".jpeg")) Action:
               ORIGIN
5  Done
6 <!--NeedCopy-->
```

Policy with a compound expression:

```
1 > add cr policy crpol11 -rule "http.req.method.eq(post) && (HTTP.REQ.
    URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))"  -action
    cache
2  Done
3 > show cr policy crpol11
4          Policy: crpol11    Rule: http.req.method.eq(post) && (HTTP.REQ
              .URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))
              Action: CACHE
5  Done
6 <!--NeedCopy-->
```

Policy that evaluates a header:

```
1 > add cr policy crpol12 -rule http.req.header("If-Modified-Since").
    exists -action origin
2  Done
3 > show cr policy crpol12
4          Policy: crpol12    Rule: http.req.header("If-Modified-Since").
              exists   Action: ORIGIN
5  Done
6 <!--NeedCopy-->
```


Modify or remove a cache redirection policy by using the CLI

- To modify a cache redirection policy, use the set cr policy command, which is just like add cr policy command, except that you enter the name of an existing policy.
- To remove a policy, use the `rm cr policy` command, which accepts only the <name> argument. If the policy is bound to a virtual server, you have to unbind the policy, before you can remove it.

For the details of unbinding a cache redirection policy, see Unbind a policy from a cache redirection virtual server.

Configure a cache redirection policy with a simple expression by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Policies.
2. In the details pane, click Add.
3. In the Create Cache Redirection Policy dialog box, in the Name* text box, type the name of the policy, and then in the Expression area, click Add.
4. To configure a simple expression, enter the expression. Following is an example of an expression that checks for a .jpeg extension in a URL:

   - Expression Type-General
   - Flow Type -REQ
   - Protocol -HTTP
   - Qualifier -URL
   - Operator - !=
   - Value - /*.jpeg

   The simple expression in the following example checks for an If-Modified-Since header in a request:

   - Expression Type -General
   - Flow Type -REQ
   - Protocol -HTTP
   - Qualifier -HEADER
   - Operator -EXISTS
   - Header Name -If-Modified-Since

5. When you are finished entering the expression, click OK or Create, and then click Close.

Configure a cache redirection policy with a compound expression by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Policies.

2. In the details pane, click Add.
3. In the Name text box, enter a name for the policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of content this policy was created to detect.
4. Choose the type of compound expression that you want to create. Your choices are:
   - **Match Any Expression**. The policy matches the traffic if one or more individual expressions match the traffic.
   - **Match All Expressions**. The policy matches the traffic only if every individual expression matches the traffic.
   - **Tabular Expressions**. Switches the Expressions list to a tabular format with three columns. In the rightmost column, you place one of the following operators:
     - The AND [&&] operator, to require that, to match the policy, a request must match both the current expression and the following expression.

The OR [	] operator, to require that, to match the policy, a request must match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.

     -

     You can also group expressions in nested subgroups by selecting an existing expression and clicking one of the following operators:
     - The BEGIN SUBGROUP [+ ( ] operator, which tells the Citrix ADC appliance to begin a nested subgroup with the selected expression. (To remove this operator from the expression, click -(.)
     - The END SUBGROUP [+)] operator, which tells the Citrix ADC appliance to end the current nested subgroup with the selected expression. (To remove this operator from the expression, click -) .)
   - **Advanced Free-Form**. Switches off the Expressions Editor entirely and turns the Expressions list into a text area in which you can type a compound expression. This is both the

most powerful and the most difficult method of creating a policy expression, and is recommended only for those thoroughly familiar with the Citrix ADC classic expressions language.

For more information about creating classic expressions in the Advanced Free-Form text area, see Configuring Classic Policies and Expressions.

Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that you want to use it.

5. If you chose Match Any Expression, Match All Expressions, or Tabular Expressions, click **Add** to display the Add Expression dialog box.

   You should leave the expression type set to General for cache redirection policies.

6. In the Flow Type drop-down list, choose a flow type for your expression.

   The flow type determines whether the policy examines incoming or outgoing connections. You have two choices:

   - **REQ**. Configures the Citrix ADC appliance to examine incoming connections, or requests.
   - **RES**. Configures the appliance to examine outgoing connections, or responses.

7. In the Protocol drop-down list, choose a protocol for your expression.

   The protocol determines the type of information that the policy examines in the request or response. Depending upon whether you chose REQ or RES in the previous drop-down list, either all four or only three of the following choices are available:

   - **HTTP**. Configures the appliance to examine the HTTP header.
   - **SSL**. Configures the appliance to examine the SSL client certificate. Available only if you chose REQ (requests) in the previous drop-down list.
   - **TCP**. Configures the appliance to examine the TCP header.
   - **IP**. Configures the appliance to examine the source or destination IP address.

8. Choose a qualifier for your expression from the Qualifier drop-down list.

   The contents of the Qualifier drop-down list depend on which protocol you chose. The following table describes the choices available for each protocol.

   Table 1. Cache Redirection Policy Qualifiers Available for Each Protocol

Protocol	Qualifier	Definition
HTTP	METHOD	HTTP method used in the request.
-	URL	Contents of the URL header.

Protocol	Qualifier	Definition
-	URLTOKENS	URL tokens in the HTTP header.
-	VERSION	HTTP version of the connection.
-	HEADER	Header portion of the HTTP request.
-	URLLEN	Length of the contents of the URL header.
-	URLQUERY	Query portion of the contents of the URL header.
-	URLQUERYLEN	Length of the query portion of the URL header.
SSL	CLIENT.CERT	SSL client certificate as a whole.
-	CLIENT.CERT.SUBJECT	Contents of the client certificate subject field.
-	CLIENT.CERT.ISSUER	Client certificate issuer.
-	CLIENT.CERT.SIGALGO	Signature algorithm used in the client certificate.
-	CLIENT.CERT.VERSION	Client certificate version.
-	CLIENT.CERT.VALIDFROM	Date from which the client certificate is valid. (The start date.)
-	CLIENT.CERT.VALIDTO	Date after which the client certificate is no longer valid. (The end date.)
-	CLIENT.CERT.SERIALNUMBER	Client certificate serial number.
-	CLIENT.CIPHER.TYPE	Encryption method used in the client certificate.
-	CLIENT.CIPHER.BITS	Number of significant bits in the encryption key.
-	CLIENT.SSL.VERSION	SSL version of the client certificate.

Protocol	Qualifier	Definition
TCP	SOURCEPORT	Source port of the TCP connection.
-	DESTPORT	Destination port of the TCP connection.
-	MSS	Maximum segment size (MSS) of the TCP connection.
IP	SOURCEIP	Source IP address of the connection.
-	DESTIP	Destination IP address of the connection.

9. Choose the operator for your expression from the Operator drop-down list.

   Your choices depend on the qualifier you chose in the previous step. The complete list of operators that can appear in this drop-down list is:

   - == . Matches the following text string exactly.
   - != . Does not match the following text string.
   - > . Is greater than the following integer.
   - CONTAINS . Contains the following text string.
   - CONTENTS . The contents of the designated header, URL, or URL query.
   - EXISTS . The specified header or query exists.
   - NOTCONTAINS . Does not contain the following text string.
   - NOTEXISTS . The specified header or query does not exist.

   If you want this policy to operate on requests sent to a specific Host, you can leave the default, the equals (==) sign.

10. If the Value text box is visible, type the appropriate string or number into the text box.

    For example, if you want this policy to select requests sent to the host shopping.example.com, you would type that string in the Value text box.

11. If you chose HEADER as the qualifier, type the header you want in the Header Name text box.
12. Click OK to add your expression to the Expression list.
13. Repeat steps 4 through 11 to create additional expressions.
14. Click Close to close the Add Expression dialog box and return to the Create Cache Redirection Policy dialog box.

Cache redirection configurations

September 14, 2021

Depending on your deployment and network topology, you can configure one of the following types of cache redirection:

- **Transparent**. A transparent cache can reside on a variety of points along a network backbone to alleviate traffic along the delivery route. In transparent mode, the cache redirection virtual server intercepts all traffic flowing to the Citrix ADC appliance and applies cache redirection policies to determine whether content should be served from the cache or from the origin server.
- **Forward proxy**. A forward proxy cache server resides on the edge of an enterprise LAN and faces the WAN. In the forward proxy mode, the cache redirection virtual server resolves the hostname of the incoming request by using a DNS server and forwards requests for non-cacheable content to the resolved origin servers. Cacheable requests are sent to the configured cache servers.
- **Reverse proxy**. Reverse proxy caches are configured for specific origin servers. Incoming traffic directed to the reverse proxy, can either be served from a cache server or be sent to the origin server with or without modification to the URL.

Configure transparent redirection

September 14, 2021

When you configure transparent cache redirection, the Citrix ADC appliance evaluates all traffic it receives, to determine whether it is cacheable. This mode alleviates traffic along the delivery route and is often used when the cache server resides on the backbone of an ISP or carrier.

By default, cacheable requests are sent to a cache server, and non-cacheable requests to the origin server. For example, when the Citrix ADC appliance receives a request that is directed to a web server, it compares the HTTP headers in the request with a set of policy expressions. If the request does not match the policy, the appliance forwards the request to a cache server. If the request does match a policy, the appliance forwards the request, unchanged, to the web server.

For details on how to modify this default behavior, see Direct policy hits to the cache instead of the origin.

To configure transparent redirection, first enable cache redirection and load balancing, and configure edge mode. Then, create a cache redirection virtual server with a wildcard IP address (*), so that this virtual server can receive traffic coming to the appliance on any IP address the appliance owns. To this virtual server, bind cache redirection policies that describe the types of requests that should not be cached. Then, create a load balancing virtual server that will receive traffic from the cache redirection

virtual server for cacheable requests. Finally, create a service that represents a physical cache server and bind it to the load balancing virtual server.

Enable cache redirection and load balancing

September 14, 2021

The appliance cache redirection and load balancing features are not enabled by default. They must be enabled before any cache redirection configuration can take effect.

Enable cache redirection and load balancing by using the CLI

At the command prompt, type the following command to enable cache redirection and load balancing and verify the settings:

```
-  enable ns feature cr lb
-  show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature cr lb
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 4)     Content Switching               CS                   ON
 5)     Cache Redirection               CR                   ON
 6)     Sure Connect

            ...
            ...
            ...

 23)    HTML Injection                  HTMLInjection        ON
 24)    appliance Push                  push                 OF
 Done
<!--NeedCopy-->
```


Enable cache redirection and load balancing by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. To enable cache redirection, in the details pane, under Modes and Features, click Configure advanced features.
   a) In Configure Advanced Features dialog box, select the check box next to the Cache Redirection, and then click OK.
   b) In Enable/Disable Feature(s)? dialog box, click Yes.
3. To enable load balancing, in the details pane, under Modes and Features, click Configure basic features.
   a) In Configure Basic Features dialog box, select the check box next to the Load Balancing, and then click OK.
   b) In Enable/Disable Feature(s)? dialog box, click Yes.

Configure edge mode

September 14, 2021

When deployed at the edge of a network, the Citrix ADC appliance dynamically learns about the servers on that network. Edge mode enables the appliance to dynamically learn about up to 40,000 HTTP servers and proxy TCP connections for these servers.

This mode turns on the collection of statistics for the dynamically learned services and is typically used in transparent deployments for cache redirection.

Enable edge mode by using the CLI

At the command prompt, type the following commands to enable edge mode and verify the setting:

```
1 -  enable ns mode Edge
2 -  show ns mode
3 <!--NeedCopy-->
```

Example:

```
1 > enable ns mode edge
2  Done
3
4 > show ns mode
5
6         Mode                          Acronym             Status
7         -------                       -------             ------
```



```
          ...
          ...
          ...
 6)     MAC-based forwarding          MBF                 ON
 7)     Edge configuration            Edge                ON
 8)     Use Subnet IP                 USNIP               OFF
          ...
          ...
          ...
 16)    Bridge BPDUs                  BridgeBPDUs         OFF
 Done
<!--NeedCopy-->
```

Enable edge mode by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In Configure Modes dialog box, select the check box next to the Edge Configuration, and then click OK.
4. In Enable/Disable Feature(s)? dialog box, click Yes.

Configure a cache redirection virtual server

September 14, 2021

By default, a cache redirection virtual server forwards cacheable requests to the load balancing virtual server for the cache, and forwards non-cacheable requests to the origin server (except in a reverse proxy configuration, in which non-cacheable requests are sent to a load balancing virtual server). There are three types of cache redirection virtual servers: transparent, forward proxy, and reverse proxy.

A transparent cache redirection virtual server uses an IP address of * and a port number, usually 80, that can accept HTTP traffic sent to any IP address that the appliance represents. As a result, you can configure only one transparent cache redirection virtual server. Any additional cache redirection virtual servers that you configure must be forward proxy or reverse proxy redirection servers.

add a cache redirection virtual server in transparent mode by using the cli

At the command prompt, type the following commands to add a cache redirection virtual server and verify the configuration:


```
1 -  add cr vserver <name> <serviceType> [<IPAddress> <port> ] [-
     cacheType <cacheType>] [-redirect <redirect>]
2 -  show cr vserver [<name>]
3 <!--NeedCopy-->
```

Example:

```
1 add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect
      POLICY
2 > show cr vserver Vserver-CRD-1
3         Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
4         State: UP  ARP:DISABLED
5         Client Idle Timeout: 180 sec
6         Down state flush: ENABLED
7         Disable Primary Vserver On Down : DISABLED
8         Default:        Content Precedence: RULE        Cache:
             TRANSPARENT
9         On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
10        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
11  Done
12 <!--NeedCopy-->
```

Modify or remove a cache redirection virtual server by using the CLI

- To modify a virtual server, use the set cr vserver command, which is just like using the add cr vserver command, except that you enter the name of an existing virtual server.
- To remove a virtual server, use the `rm cr vserver` command, which accepts only the <name> argument.

Add a cache redirection virtual server in transparent mode by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Cache Redirection) dialog box, specify values for the following parameters as shown:
   - Name*—name
   - Port*—port

   *A required parameter

4. In the Protocol drop-down list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the standard port for the selected protocol, enter a new value in the Port field.
5. Click the Advanced tab.
6. Verify that Cache Type is set to TRANSPARENT and Redirect is set to POLICY.
7. Click Create, and then click Close. The Cache Redirection Virtual Servers pane displays the new virtual server.
8. Select the new cache redirection virtual server to display the details of its configuration.

Bind policies to the cache redirection virtual server

September 14, 2021

Cache redirection policies are not automatically bound to the cache redirection virtual server. A policy based cache redirection virtual server cannot function unless you bind at least one policy to it.

Bind policies to a cache redirection virtual server by using the CLI

At the command prompt, type:

```
-  bind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
 Done

> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
```



```
15          Disable Primary Vserver On Down : DISABLED
16          Default:        Content Precedence: RULE        Cache:
               TRANSPARENT
17          On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
18          Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
19
20  1)      Cache bypass  Policy: bypass-cache-control
21  2)      Cache bypass  Policy: bypass-dynamic-url
22  3)      Cache bypass  Policy: bypass-urltokens
23  4)      Cache bypass  Policy: bypass-cookie
24   Done
25  <!--NeedCopy-->
```

Bind a user-defined policy to a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. Click the virtual server that you want to configure, and click Open.
3. On the Policies tab, select type of the policy and then click Insert Policy.
4. Under Policy Name column, select the policy that you want to bind.
5. Click OK.

Unbind a policy from a cache redirection virtual server

September 14, 2021

When you unbind a policy from the cache redirection virtual server, the Citrix ADC appliance no longer applies the policy when evaluating client requests.

Unbind a policy from a cache redirection virtual server by using the command CLI

At the command prompt, type:

```
1  -  unbind cr vserver <name> -policyName <string>
2  -  show cr vserver [<name>]
3  <!--NeedCopy-->
```

Example:

```
1  unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
2  > show cr vserver Vserver-CRD-1
3          Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
```



```
4          State: UP  ARP:DISABLED
5          Client Idle Timeout: 180 sec
6          Down state flush: ENABLED
7          Disable Primary Vserver On Down : DISABLED
8          Default:        Content Precedence: RULE        Cache:
               TRANSPARENT
9          On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
10         Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
11
12 1)      Cache bypass  Policy: bypass-cache-control
13  Done
14 <!--NeedCopy-->
```

Unbind a user-defined policy from a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. Click the virtual server that you want to configure, and then click Open.
3. On the Policies tab, under Policy Name, select the policy that you want to unbind.
4. Click Unbind Policy, and then click OK.

Create a load balancing virtual server

September 14, 2021

The cache redirection virtual server on the Citrix ADC appliance can send requests to either a cache server farm, if the request is cacheable, or to the origin server farm if the request is not cacheable.

Each cache server is represented on the appliance by a service, which is bound to a load balancing virtual server that receives requests from the cache redirection virtual server and forwards those requests to the servers.

For details on configuring load balancing virtual servers and other configuration options, see Load Balancing.

Create a load balancing virtual server by using the CLI

At the command prompt, type the following commands to create a load balancing virtual server and verify the configuration:

```
1 -  add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
2 -  show lb vserver [<name>]
3 <!--NeedCopy-->
```


Example:

```
> add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:00:08.470
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
 Done
<!--NeedCopy-->
```

Create a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name*-name
   - IP Address*- IPAddress
   - Port*-port

   *A required parameter
4. In the Protocol list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the well-known port for the selected protocol, enter a new value in the Port field.
5. Click Create, and then click Close. The Load Balancing Virtual Servers pane displays the new virtual server.

Configure an HTTP service

September 14, 2021

On the Citrix ADC appliance, a service represents a physical server on the network. In the transparent cache redirection configuration, the service represents the cache server. Cacheable requests are sent by the cache redirection virtual server to the load balancing virtual server, which in turn forwards each request to the correct service, which passes it on to the cache server.

Configure an HTTP service by using the CLI

At the command prompt, type the following commands to create an HTTP service and verify the configuration:

```
-  add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
-  show service [<name>]
<!--NeedCopy-->
```

Example:

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType
   TRANSPARENT
 Done
> show service Service-HTTP-1
        Service-HTTP-1 (10.102.29.40:80) - HTTP
        State: DOWN
        Last state change was at Fri Jul  2 09:14:17 2010
        Time since last state change: 0 days, 00:00:13.820
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cache Type: TRANSPARENT Redirect Mode:
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED

```



```
1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: N/A
 Done
<!--NeedCopy-->
```

Modify or remove a service by using the CLI

- To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service.
- To remove a service, use the `rm service` command, which accepts only the <name> argument.

Add an HTTP service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name*—name
   - Server*— IP
   - Port*—port

   *A required parameter
4. In the Protocol* drop-down list, select a supported protocol (for example, **HTTP**).
5. Click Create, and then click Close.

Bind/unbind a service to/from a load balancing virtual server

September 14, 2021

You must bind a service to the load balancing virtual server. This enables the load balancer to forward the request to the server that the service represents. If your configuration changes, you can unbind a service from the load balancing virtual server.

Bind a service to a load balancing virtual server by using the CLI

At the command prompt, type:

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:42:25.610
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

Unbind a service from a load balancing virtual server by using the CLI

To unbind a service, use the `unbind lb vserver` command instead of `bind lb vserver`.

Bind/unbind a service from a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers
2. In the details pane, select the virtual server from which you want to bind/unbind the service, and then click Open.

3. On the Services tab, in the Active column, select/clear the check box next to the Service Name.
4. Click OK.

Disable the use the proxy port setting for transparent caching

September 14, 2021

If the use source IP (USIP) option is disabled on a cache service configured on the Citrix ADC appliance, the appliance forwards client requests to the cache service by using a appliance-owned subnet IP (SNIP) address or mapped IP (MIP) address as the source IP address and a random port as the source port. The randomly selected port is called the proxy port.

However, if you want to configure a fully transparent cache (a cache configuration in which the cache service receives the client's IP address and port number), you must not only enable the USIP option, either globally or on the cache service, but also disable the Use Proxy Port setting, either globally or on the cache service. Disabling the Use Proxy Port setting enables the appliance to use the client's source port as the source port when it connects to the cache service, and ensures a fully transparent cache configuration.

For more information about configuring the Use Proxy Port option globally or on a service, see Configuring the Source Port for Server-Side Connections.

Assign a port range to the Citrix ADC appliance

September 14, 2021

Sharing of the client IP address may create a conflict that makes network devices, such as routers, cache servers, origin servers, and other Citrix ADC appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

A method to solve this problem is to assign a source port range to the Citrix ADC appliance. This allotment enables network devices to unambiguously identify the Citrix ADC appliance that sent the request.

Assign a source port range to a Citrix ADC appliance by using the CLI

At the command prompt, type:

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```


Assign a source port range to a Citrix ADC appliance by using the appliance GUI

1. In the navigation pane, click System, and then click Settings.
2. In the Settings group, click the Change global system settings link.
3. In the Cache Redirection Port Range group, specify the port range for the appliance by typing a port number for Start Port and a port number for End Port.
4. Click OK.

Enable load balancing virtual servers to redirect requests to cache

September 14, 2021

If a load balancing virtual server is configured to listen on a particular IP address and port combination, it takes precedence over the cache redirection virtual server for any requests destined for that address-port combination. Therefore, the cache redirection virtual server does not process those requests.

If you want to override this functionality and let the cache redirection virtual server decide whether the request should be served from the cache or not, configure the particular load balancing virtual server to be cacheable.

Such a configuration is typically used when an ISP uses a Citrix ADC appliance at the edge of its network and all traffic flows through the appliance.

Enable load balancing virtual servers to redirect requests to the cache by using the CLI

At the command prompt, type:

```
-  set lb vserver <name> [-cacheable ( YES | NO)]
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-CR  - cacheable YES
> show lb vserver vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 01:05:51.510
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
```



```
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Cacheable: YES  PQ: OFF SC: OFF
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

For transparent cache redirection, the appliance intercepts all traffic and evaluates every request to determine whether it is cacheable. Non-cacheable requests are sent unchanged to the origin server.

When using transparent cache redirection, you may want to turn off cache redirection for load balancing virtual servers that always direct traffic to origin servers.

Turn off caching for a load balancing virtual server by using the CLI

To turn off caching for a load balancing virtual, use the unset lb vserver command instead of set lb vserver. Specify a value of NO value for the **cacheable** parameter.

Enable or disable load balancing virtual servers to redirect requests to the cache by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server from which you want to enable/disable the caching, and then click Open.
3. On the Advanced tab, select/clear Cache Redirection check box.
4. Click OK.

Configure forward proxy redirection

September 14, 2021

A forward proxy is a single point of contact for a client or group of clients. In this configuration, the Citrix ADC appliance redirects non-cacheable requests to an origin server and redirects cacheable requests to either a forward proxy cache or a transparent cache.

When the appliance is configured as a forward proxy, users must modify their browsers so that the browser sends requests to the forward proxy instead of the destination servers.

A forward proxy cache redirection virtual server on the appliance compares the request with a policy for caching. If the request is not cacheable, the appliance queries a DNS load balancing virtual server for resolution of the destination, and then sends the request to the origin server. If the request is cacheable, the appliance forwards the request to a load balancing virtual server for the cache.

The appliance relies on a host domain name or IP address in the request's HOST header to determine the requested destination. If there is no HOST header in the request, the appliance inserts a HOST header based on the destination IP address in the request.

Typically, the Citrix ADC appliance acts as a forward proxy in an enterprise LAN. In such a configuration, the appliance resides at the edge of an enterprise LAN and intercepts client requests before they are fanned out to the WAN. Configuring the appliance in the forward proxy mode reduces traffic on the WAN.

To configure forward proxy cache redirection, first enable load balancing and cache redirection on the appliance. Then, configure a DNS load balancing virtual server and associated services. Also configure a load balancing virtual server and bind to it appropriate services for the cache. Configure a forward proxy cache redirection virtual server and bind the DNS and load balancing virtual servers to it. You must also configure caching policies and bind them to the cache redirection virtual server. To complete the setup, configure the client browsers to use the forward proxy.

For details on how to enable cache redirection and load balancing on the appliance, see Enable cache redirection and load balancing.

For details on how to create a load balancing virtual server, see Create a load balancing virtual server.

For details on how to configure services that represent the cache server, see Configure an HTTP service.

For details on how to bind the service to a virtual server, see Bind/unbind a service to/from a load balancing virtual server.

For details on how to create a forward proxy cache redirection server, see Configure a cache redirection virtual server, and create a virtual server of type TRANSPARENT or FORWARD.

For details on binding cache redirection policies to the cache redirection virtual server, see Configure a cache redirection policy.

Create a DNS service

September 14, 2021

A DNS service is a representation, on the Citrix ADC appliance, of a physical DNS server in the network. A DNS load balancing virtual server sends DNS requests to the DNS server in the network through such a service.

Create a DNS service by using the CLI

At the command line, type the following commands to create a DNS service and verify the configuration :

```
-  add service <name> <IP> <serviceType> <port>
-  show service [<name>]
<!--NeedCopy-->
```

Example:

```
add service Service-DNS-1 10.102.29.41 DNS 53
show service Service-DNS-1
        Service-DNS-1 (10.102.29.41:53) - DNS
        State: DOWN
        Last state change was at Fri Jul  2 10:14:32 2010
        Time since last state change: 0 days, 00:00:13.550
        Server Name: 10.102.29.41
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Probe timed out.
```



```
                Response Time: 2000.0 millisec
 Done
<!--NeedCopy-->
```

Add a DNS service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name*—name
   - Server*—IP
   - Port*—port

*A required parameter

1. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
2. Click Create, and then click Close.

Create a DNS load balancing virtual server

September 14, 2021

The DNS virtual server enables the forward proxy to perform DNS resolution before forwarding a client request to an origin server. The DNS load balancing virtual server is associated with the DNS service that represents the physical DNS server on the network.

Create a DNS load balancing virtual server by using the CLI

At the command line, type the following commands to create a DNS load balancing virtual server and verify the configuration:

```
-  add lb vserver <name> <serviceType>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
```



```
4          Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
5          State: DOWN
6          Last state change was at Fri Jul  2 10:32:28 2010
7          Time since last state change: 0 days, 00:00:08.10
8          Effective State: DOWN  ARP:DISABLED
9          Client Idle Timeout: 120 sec
10         Down state flush: ENABLED
11         Disable Primary Vserver On Down : DISABLED
12         No. of Bound Services :  0 (Total)       0 (Active)
13         Configured Method: LEASTCONNECTION
14         Mode: IP
15         Persistence: NONE
16  Done
17 <!--NeedCopy-->
```

Create a DNS load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, in the Name box, type a name for the virtual server.
4. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
5. Click Create, and then click Close. The DNS Virtual Servers pane displays the new virtual server.

Bind the DNS service to the virtual server

September 14, 2021

For the DNS server to respond to DNS requests, the service representing the DNS server must be bound to the DNS virtual server.

Bind the DNS service to the load balancing virtual server by using the CLI

At the command prompt, type the following commands to bind the DNS service to the load balancing virtual server and verify the configuration:

```
1 -  bind lb vserver <name> <serviceName>
2 -  show lb vserver <name>
3 <!--NeedCopy-->
```

Example:


```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:12:16.80
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
<!--NeedCopy-->
```

Unbind a DNS service from the load balancing virtual server by using the CLI

Use the `unbind lb vserver` command instead of `bind lb vserver`.

Bind/Unbind a DNS service to/from a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers
2. In the details pane, select the virtual server to/from which you want to bind/unbind the DNS service, and then click Open.
3. On the Services tab, in the Active column, select/clear the check box next to the Service Name.
4. Click OK.

Configure a client web browser to use a forward proxy

September 14, 2021

When you configure the Citrix ADC appliance as forward proxy cache redirection virtual server in the network, you must configure the client Web browser to send requests to the forward proxy. Typically,

when you use a forward proxy, the only route to the servers in the network is through the forward proxy.

Refer the documentation for your browser to configure the browser to use a forward proxy. Specify the IP address and port number of the forward proxy cache redirection virtual server for this configuration.

Configure reverse proxy redirection

September 14, 2021

A reverse proxy resides in front of one or more Web servers and shields the origin server from client requests. Often, a reverse proxy cache is a front-end for all client requests to a server. An administrator assigns a reverse proxy cache to a specific origin server. The reverse proxy cache is unlike transparent and forward proxy caches, which cache frequently requested content for all requests to any origin server, and the choice of a server is based on the request.

Unlike a transparent proxy cache, the reverse proxy cache has its own IP address and can replace destination domains and URLs in a non-cacheable request with new destination domains and URLs.

You can deploy reverse proxy cache redirection at the origin-server side or at the edge of a network. When deployed at the origin server, the reverse proxy cache redirection virtual server is a front-end for all requests to the origin server.

In the reverse proxy mode, when the appliance receives a request, a cache redirection virtual server evaluates the request and forwards it to either a load balancing virtual server for the cache or a load balancing virtual server for the origin. The incoming request can be transformed by changing the host header or the host URL before they it is sent to the back-end server.

To configure reverse proxy cache redirection, first enable cache redirection and load balancing. Then, configure a load balancing virtual server and services to send cacheable requests to the cache servers. Also configure a load balancing virtual server and associated services for the origin servers. Then, configure a reverse proxy cache redirection virtual server and bind relevant cache redirection policies to it. Finally, configure mapping policies and bind them to the reverse proxy cache redirection virtual server.

The mapping policies have an associated action that enables the cache redirection virtual server to forward any non-cacheable request to the load balancing virtual server for the origin.

Be sure to create the default cache server destination.

For details on how to enable cache redirection and load balancing on the appliance, see Enable cache redirection and load balancing.

For details on how to create a load balancing virtual server, see Create a load balancing virtual server.

For details on how to configure services that represent the cache server, see Configure an HTTP service.

For details on how to bind the service to a virtual server, see Bind/unbind a service to/from a load balancing virtual server.

For details on how to create a reverse proxy cache redirection server, see Configure a cache redirection virtual server, and create a virtual server of type REVERSE.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Bind policies to the cache redirection virtual server.

Configure mapping policies

If an incoming request is non-cacheable, the reverse-proxy cache redirection virtual server replaces the domain and URL in the request with the domain and URL of a target origin server and forwards the request to the load balancing virtual server for the origin.

A mapping policy enables the reverse proxy cache redirection virtual server to replace the destination domain and URL and forward the request to the load balancing virtual server for the origin.

A mapping policy must first translate the domain and the URL, and then pass the request on to the origin load balancing virtual server.

A mapping policy can map a domain, a URL prefix, and a URL suffix, as follows:

- Domain mapping: You can map a domain without a prefix or suffix. The domain mapping is the default mapping for the virtual server (for example, mapping www.mycompany.com to www.myrealcompany.com).
- Prefix mapping: You can replace a specified pattern prefixed as part of the URL (for example, mapping www.mycompany.com/sports/index.html to www.mycompany.com/news/index.html).
- Suffix mapping: You can replace the file suffix in the URL (for example, mapping www.mycompany.com/sport to www.mycompany.com/sports/index.asp).

The source and the destination strings being mapped must be similar. If you specify a source domain, you must specify a destination domain, and if you specify a source suffix, you must specify a destination suffix. Similarly, if you specify an exact URL from the source, the target URL must also be an exact URL.

Once you configure mapping policies for the reverse proxy mode, you must bind them to the cache redirection virtual server.

You can use combinations of the source URL, target URL, and source and target domains to configure all three types of domain mapping.

Configure a mapping policy for reverse proxy mode by using the CLI

At the command prompt, type the following command to add a policy map and verify the configuration:

```
1 -  add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <
     string>] [-tu <string>]
2 -  show policy map [<mapPolicyName>]
3 <!--NeedCopy-->
```

Example:

The following command maps a domain in a client request to a target domain:

```
1 > add policy map myMappingPolicy -sd www.mycompany.com -td www.
     myrealcompany.com
2  Done
3 > show policy map myMappingPolicy
4 1)      Name: myMappingPolicy
5         Source Domain: www.mycompany.com        Source Url:
6         Target Domain: www.myrealcompany.com    Target Url:
7  Done
8 <!--NeedCopy-->
```

Following is an example of mapping a URL suffix to a different URL suffix:

```
1 > add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.
     myrealcompany.com -su /news.html -tu /realnews.html
2  Done
3 > show policy map myOtherMappingPolicy
4 1)      Name: myOtherMappingPolicy
5         Source Domain: www.mycompany.com        Source Url: /news.html
6         Target Domain: www.myrealcompany.com    Target Url: /realnews.
             html
7  Done
8 <!--NeedCopy-->
```

Configure a mapping policy for reverse proxy mode by using the GUI

1. Navigate to **Traffic Management > Cache Redirection > Map Policies**.
2. In the details pane, click Add.
3. In the Create Map Policy dialog box, specify values for the following parameters as shown:
   - Name*- mapPolicyName

- Source Domain*-sd
- Target Domain*-td
- Source URL-su
- Target URL-tu

*A required parameter

4. Click Create, and then click Close. The Map pane displays the new mapping policy.

Bind the mapping policy to the cache redirection virtual server by using the CLI

At the command prompt, type the following commands to bind the mapping policy to the cache redirection virtual server and verify the configuration:

```
-  bind cr vserver <name> -policyName <string> [<targetVserver>]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-
    CR
 Done
> show cr vserver Vserver-CRD-3
        Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
        State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default: Vserver-LB-CR  Content Precedence: RULE        Cache:
           REVERSE
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Policy:         Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)      Map: myMappingPolicy Target: Vserver-LB-CR
 Done
<!--NeedCopy-->
```

Bind the mapping policy to the cache redirection virtual server by using the GUI

1. Navigate to **Traffic Management > Cache Redirection > Virtual Servers**.
2. In the details pane, select the virtual server from which you want to bind the mapping policy, and then click **Open**.

3. In the **Configure Virtual Server**(Cache Redirection), on the **Policies** tab, select **Map**, and then click **Insert Policy**.
4. In the **Policy Name** column, select the policy from the drop-down list.
5. In the **Target** column, click the down arrow, and then select the virtual server from the drop-down list.
6. Click **OK**.

Selective cache redirection

September 14, 2021

Selective cache redirection sends requests for particular types of content, for example, images, to one cache server or group of cache servers and sends other types of content to a different cache server or group of cache servers. You can configure advanced cache redirection in transparent, reverse proxy, or forward proxy modes.

In selective cache redirection, the Citrix ADC appliance intercepts a client request and forwards non-cacheable requests to the original destination in the client request. For cacheable requests, the appliance sends the requests to the destination cache server that can serve content of a specific content type.

Selective cache redirection involves configuring content switching policies in addition to cache redirection policies. The appliance first evaluates the cache redirection policies that are bound to the cache redirection virtual server. If a request matches a cache redirection policy, the cache redirection virtual server sends the request to the origin server or a load balancing virtual server for the origin. If no cache redirection policies match the request, the appliance evaluates the content switching policies bound to the cache redirection virtual server. If a content switching policy matches the request, the cache redirection virtual server redirects the request to a load balancing virtual server for the cache.

To configure selective cache redirection, first enable cache redirection, load balancing, and content switching on the Citrix ADC appliance. Then, configure a load balancing virtual server for the cache and an associated HTTP service. After this, configure a cache redirection virtual server and bind both the cache redirection and content switching policies to it. Once you have bound the policies, you can configure the virtual server to give precedence to either rule based or URL based content-switching policies.

When configured for transparent mode cache redirection in an edge deployment topology, the appliance sends all cacheable HTTP traffic to a transparent cache farm. Clients access the Internet through the appliance, which is configured as a Layer 4 switch that receives traffic on port 80.

The appliance can direct requests for images (for example, .png and .jpg files) to one server in the

transparent cache farm, and all other requests for static content to other servers in the farm. For this configuration, you configure content switching policies to send images to the image cache and send all other cacheable content to a default cache.

Note: The configuration described here is for transparent selective cache redirection. Therefore, it does not require a load balancing virtual server for the origin, as would a reverse proxy configuration.

To configure this type of selective cache redirection, first enable cache redirection, load balancing, and content switching. Then, configure a load balancing virtual server for the cache and configure an associated HTTP service. Then, configure a cache redirection virtual server and create and bind both cache redirection and content switching policies to this virtual server.

For details on how to enable cache redirection and load balancing on the appliance, see Enable cache redirection and load balancing.

Enable content switching

September 14, 2021

To configure selective cache redirection, after you enable both the load balancing and cache redirection features on the appliance, you must enable content switching.

Enable content switching by using the CLI

At the command prompt, type:

```
-  enable ns feature CS

-  show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature cs
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 4)     Content Switching               CS                  ON
```



```
5)     Cache Redirection              CR                    ON
          ...
          ...
          ...
23)    HTML Injection                 HTMLInjection         ON
24)    appliance Push                 push                  OFF
Done
<!--NeedCopy-->
```

Enable cache redirection and load balancing by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure basic features.
3. In Configure Basic Features dialog box, select the check box next to the Content Switching, and then click OK.
4. In Enable/Disable Feature(s)? dialog box, click Yes.

Configure a load balancing virtual server for the cache

September 14, 2021

Create a load balancing virtual server and an HTTP service for each type of cache server that will be used. For example, if you want to serve JPEG files from one cache server and GIF files from another cache server, and use a third cache server for the rest of the content, create an HTTP service and virtual server for each of the three types of cache servers. Then bind each service to its respective virtual server.

For details on how to create a load balancing virtual server, see Create a load balancing virtual server.

For details on how to configure services that represent the cache server, see Configure an HTTP service.

For details on how to bind the service to a virtual server, see Bind/unbind a service to/from a load balancing virtual server.

For details on how to create a transparent proxy cache redirection server, see Configure a cache redirection virtual server, and create a virtual server of type TRANSPARENT.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Bind policies to the cache redirection virtual server.

Configure a cache redirection policy for a specific type of content

To identify requests that contain a .png or .jpeg extension as cacheable, you configure a cache redirection policy and bind it to the cache redirection virtual server.

Note: If a request matches a policy, the Citrix ADC appliance forwards it to the origin server. As a result, in the following procedure, you configure policies to match requests that do *not* have ".png" or ".jpeg" extensions.

To configure cache redirection for a specific type of content, configure a policy that uses a simple expression, as described in Configure a cache redirection policy.

Configure policies for content switching

September 14, 2021

You must create a content switching policy to identify specific types of content to be cached in one cache server or farm and identify other types of content to serve from another cache server or farm. For example, you can configure a policy to determine the location for image files with .png and .jpeg extensions.

After defining the content switching policy, you bind it to a cache redirection virtual server and specify a load balancing virtual server. Requests that match the policy are forwarded to the named load balancing virtual server. Requests that do not match the content switching policy are forwarded to the default load balancing virtual server for the cache.

For more details about the content switching feature and configuring content switching policies, see Content switching.

You must first create the content switching policy and then bind it to the cache redirection virtual server.

Create a content switching policy by using the command CLI

At the command line, type:

```
-  add cs policy <policyName> [-url <string> | -rule <expression>]
-  show cs policy [<policyName>]
<!--NeedCopy-->
```

Examples:

```
> add cs policy Policy-CS-JPEG -rule "REQ.HTTP.URL == '/*.jpeg'"
 Done
```



```
> show cs policy Policy-CS-JPEG
         Rule: REQ.HTTP.URL == '/*.jpeg'         Policy: Policy-CS-JPEG
         Hits: 0
 Done
>

> add cs policy Policy-CS-GIF -rule "REQ.HTTP.URL == '/ *.png'"
 Done
> show cs policy Policy-CS-GIF
         Rule: REQ.HTTP.URL == '/ *.png'         Policy: Policy-CS-GIF
         Hits: 0
 Done
>

> add cs policy Policy-CS-JPEG-URL -url /*.jpg
 Done
> show cs policy Policy-CS-JPEG-URL
         URL: /*.jpg     Policy: Policy-CS-JPEG-URL
         Hits: 0
 Done
>

> add cs policy Policy-CS-GIF-URL -url /*.png
 Done
> show cs policy Policy-CS-GIF-URL
         URL: /*.png     Policy: Policy-CS-GIF-URL
         Hits: 0
 Done
<!--NeedCopy-->
```

Create a URL-based content switching policy by using the GUI

1. Navigate to Traffic Management > Content Switching > Policies.
2. In the details pane, click Add.
3. In the Create Content Switching Policy dialog box, in the Name text box, type a name for the policy.
4. Select the URL radio button.
5. In the Value text box, type the string value (for example, **/sports**).
6. Click Create and click Close. The policy you created appears in the Content Switching Policies page.

Create a rule-based content switching policy by using the GUI

1. Navigate to Traffic Management > Content Switching > Policies.
2. In the details pane, click Add.
3. In the Create Content Switching Policy dialog box, in the Name text box, type a name for the policy.
4. Select the Expression radio button, and then click Configure.
5. In the Create Expression dialog box, choose the expression syntax that you want to use.
   - If you want to use default syntax, accept the default and proceed to the next step.
   - If you want to use classic syntax, click Switch to Classic Syntax.

   The Expression portion of the dialog box changes to match your choice. The default syntax Expression view has fewer elements than does the classic syntax Expression view. In the default syntax Expression view, instead of a preview window, a button provides access to an expression evaluator. The evaluator evaluates the expression you entered, to verify that it is valid, and displays an analysis of the expression's effect.
6. Enter your policy expressions.

   For information about using the advanced syntax, see Configure advanced policy expression: Get started.
7. Click **Create** and click **Close**. The policy you created appears in the **Content Switching Policies** pane.

Bind the content switching policy to a cache redirection virtual server by using the CLI

At the command prompt, type the following commands to bind the content switching policy to a cache redirection virtual server and verify the configuration:

```
-  bind cs vserver <name> <targetVserver> [-policyName <string>]
-  show cs vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> bind cs vserver Vserver-CR-1 lbcachejpeg -policyName Policy-CS-JPEG
 Done
> bind cs vserver Vserver-CR-1 lbcachegif -policyName Policy-CS-GIF
 Done
> show cs vserver Vserver-CR-1
        Vserver-CR-1 (10.102.29.60:80) - HTTP   Type: CONTENT
        State: UP
```



```
        Last state change was at Fri Jul  2 12:53:45 2010
        Time since last state change: 0 days, 00:00:58.920
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Cacheable: YES
        Vserver IP and Port insertion: OFF
        Case Sensitivity: ON
        Push: DISABLED  Push VServer:
        Push Label Rule: none

1)      Policy: Policy-CS-JPEG  Target: lbcachejpeg     Priority: 0
       Hits: 0
2)      Policy: Policy-CS-GIF   Target: lbcachegif      Priority: 0
       Hits: 0
 Done
>
<!--NeedCopy-->
```

Bind the content switching policy to a cache redirection virtual server by using the GUI

1. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the policy (for example, **Vserver-CS-1**), and then click **Open**.
3. In the Configure Virtual Server (Content Switching) dialog box, on the **Policies** tab, click CSW, and then click **Insert Policy**.
4. In the **Policy Name** column, select the policy that you want to configure for the content switching virtual server.
5. In the **Target** column, click the green arrow, and select the target load balancing virtual server from the list.
6. Click **OK**.

Configure precedence for policy evaluation

September 14, 2021

You can configure a content switching policy based on either a rule, which is a generic configuration to accommodate various content types, or a URL, which is more specific and defines exactly the type of content that has to be sent to a particular cache server. Essentially, the same content can be defined by either a rule based policy or a URL based policy.

Once you bind content switching policies of either type to a cache redirection virtual server, you can configure the virtual server to give precedence to either rule based or URL based policies. This will, in turn, decide which servers the particular requests are directed to.

To configure precedence for policy evaluation, use the precedence parameter, which specifies the type of policy (URL or RULE) that takes precedence on the content redirection virtual server.

Possible values: RULE, URL

Default value: RULE

Configure precedence for policy evaluation by using the CLI

At the command prompt, type the following commands to configure precedence for policy evaluation and verify the configuration:

```
-  set cr vserver <name> [-precedence (RULE | URL)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```


Configure precedence for policy evaluation by using the GUI

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure precedence, (for example, **Vserver-CS-1**), and then click Open.
3. In the Configure Virtual Server (Content Switching) dialog box, on the Advanced tab, next to Precedence, click Rule or URL, and then click OK.

Administer a cache redirection virtual server

September 14, 2021

To administer a cache redirection virtual server, you need to view cache redirection statistics. You might need to enable or disable cache redirection servers, or direct policy hits to the cache instead of the origin. Administrative tasks also include backing up a cache redirection virtual server and managing client connections.

View cache redirection virtual server statistics

September 14, 2021

You can view properties of a cache redirection virtual server and statistics on the traffic that has passed through a cache redirection virtual server. You can also view the cache redirection virtual servers and policies that you have bound to load balancing virtual servers.

To view statistics for a specific cache redirection virtual servers, use the name parameter to specify the name of the virtual server for which statistics will be displayed. Otherwise, statistics for all cache redirection virtual servers are displayed. Maximum Length: 127

View statistics for a cache redirection virtual server by using the CLI

At the command prompt, type:

```
stat cr vserver [<name>]
```

Example:

```
1 > stat cr vserver Vserver-CRD-1
2
3 Vserver Summary
4                          IP  port     Protocol        State
```



```
Vser...CRD-1          0.0.0.0    80          HTTP            UP

VServer Stats:
                                          Rate (/s)
                                             Total
Requests                                          0
                        0
Responses                                         0
                        0
Request bytes                                     0
                        0
Response bytes                                    0
                        0

 Done
>
<!--NeedCopy-->
```

View statistics for a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers
2. In the details pane, select the virtual server for which you want to view statistics, (for example, **Vserver-CRD-1**), and then click Statistics.

Omit the server name to display basic statistics for all cache redirection virtual servers. Include the server name to display detailed statistics for that virtual server, including number and size of requests and responses that pass through the virtual server

View the statistics of a cache redirection virtual server by using the monitoring and dashboard utilities

1. To view the statistics by using the monitoring utilities, click the Monitoring tab.
2. In the Select Group drop-down menu, choose CR Virtual Servers. A list of cache redirection virtual servers appears.
3. To view the statistics by using the dashboard utilities, click the Dashboard tab.
4. Click Applet Client or Web Start Client next to Statistical Utility.
5. In the Select Group drop-down menu, choose CR Virtual Servers. The dashboard displays summary statistics for the cache redirection virtual servers.
6. To see a chart of virtual server activity, click Chart. A graphical representation of the virtual server statistics appears.

Enable or disable a cache redirection virtual server

September 14, 2021

When you create a cache redirection virtual server, it is enabled by default. If you disable a cache redirection virtual server, its state changes to OUT OF SERVICE and it stops redirecting cacheable client requests. However, the Citrix ADC appliance continues to respond to ARP and ping requests for the IP address of this virtual server.

Enable or disable a cache redirection virtual server by using the CLI

At the command line, type one of the following commands:

```
-  enable cr vserver <name>
-  show cr vserver <name>
-  disable cr vserver <name>
-  show cr vserver <name>
<!--NeedCopy-->
```

Examples:

```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>

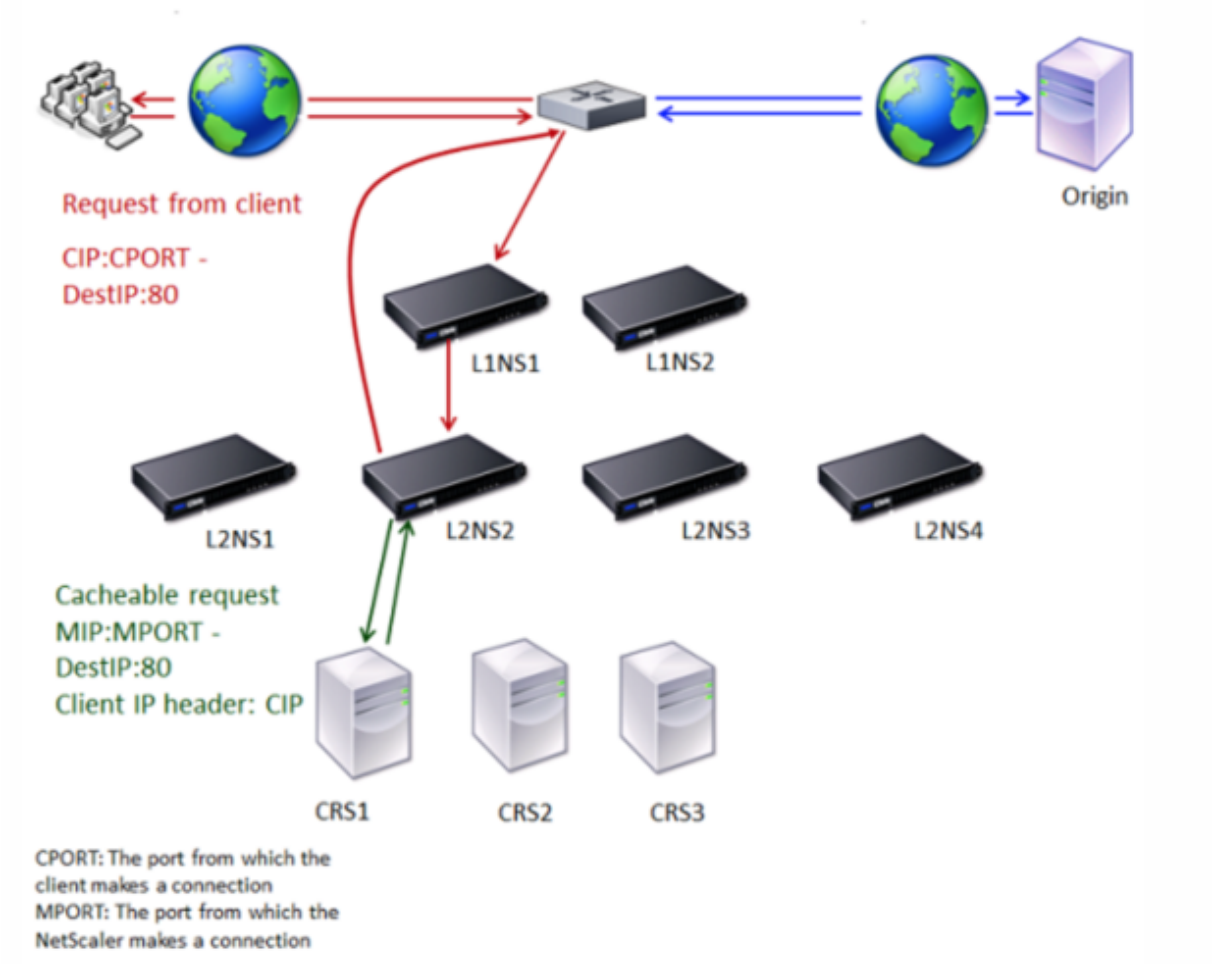
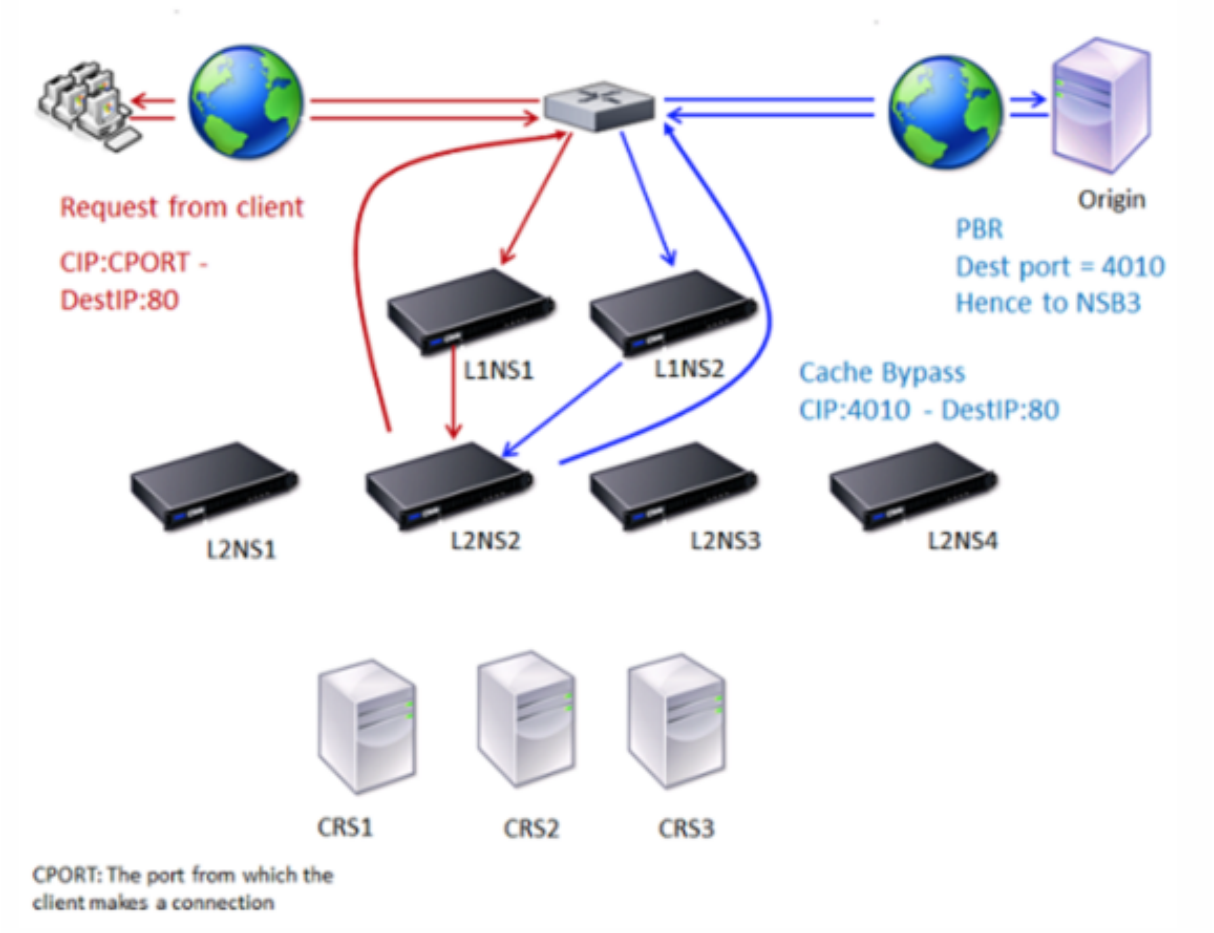
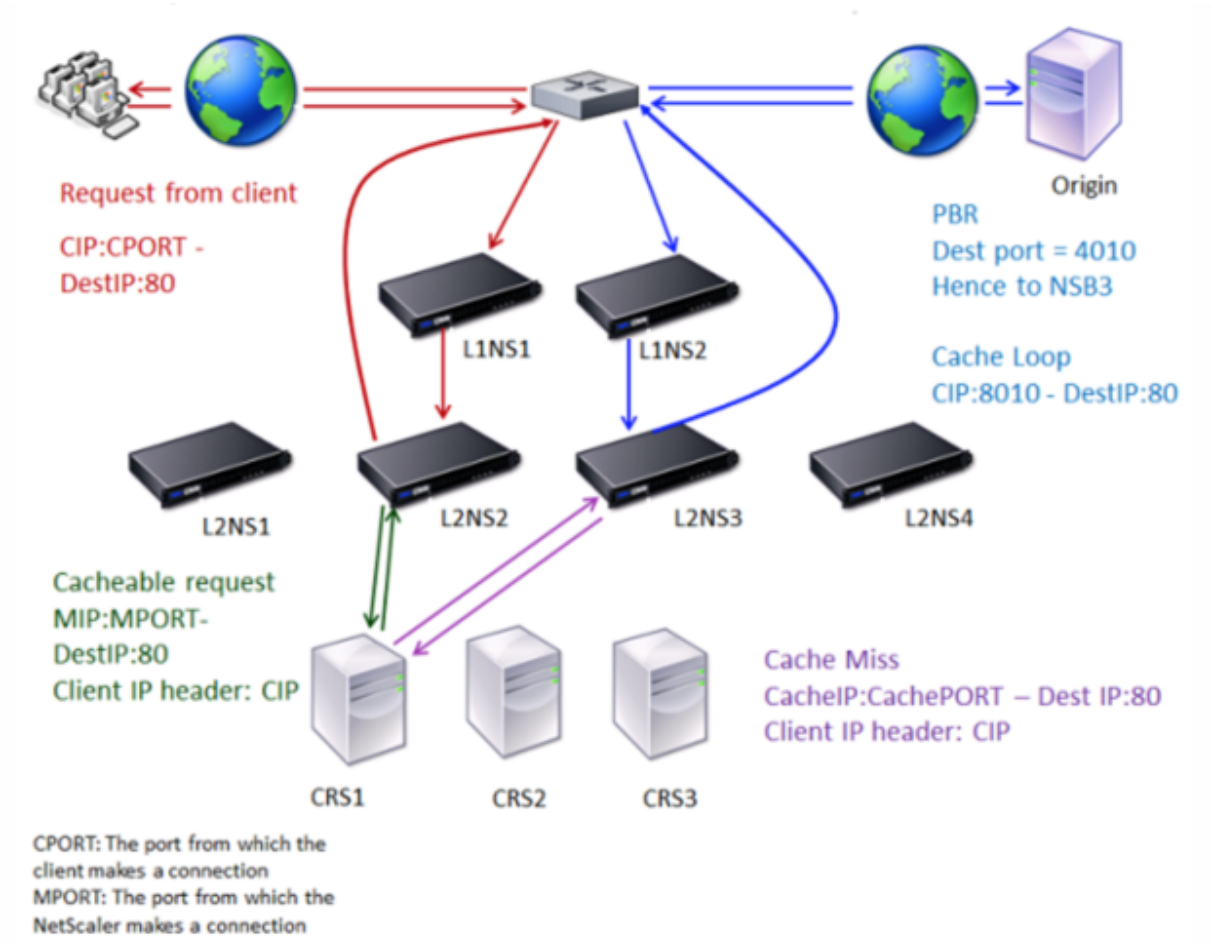
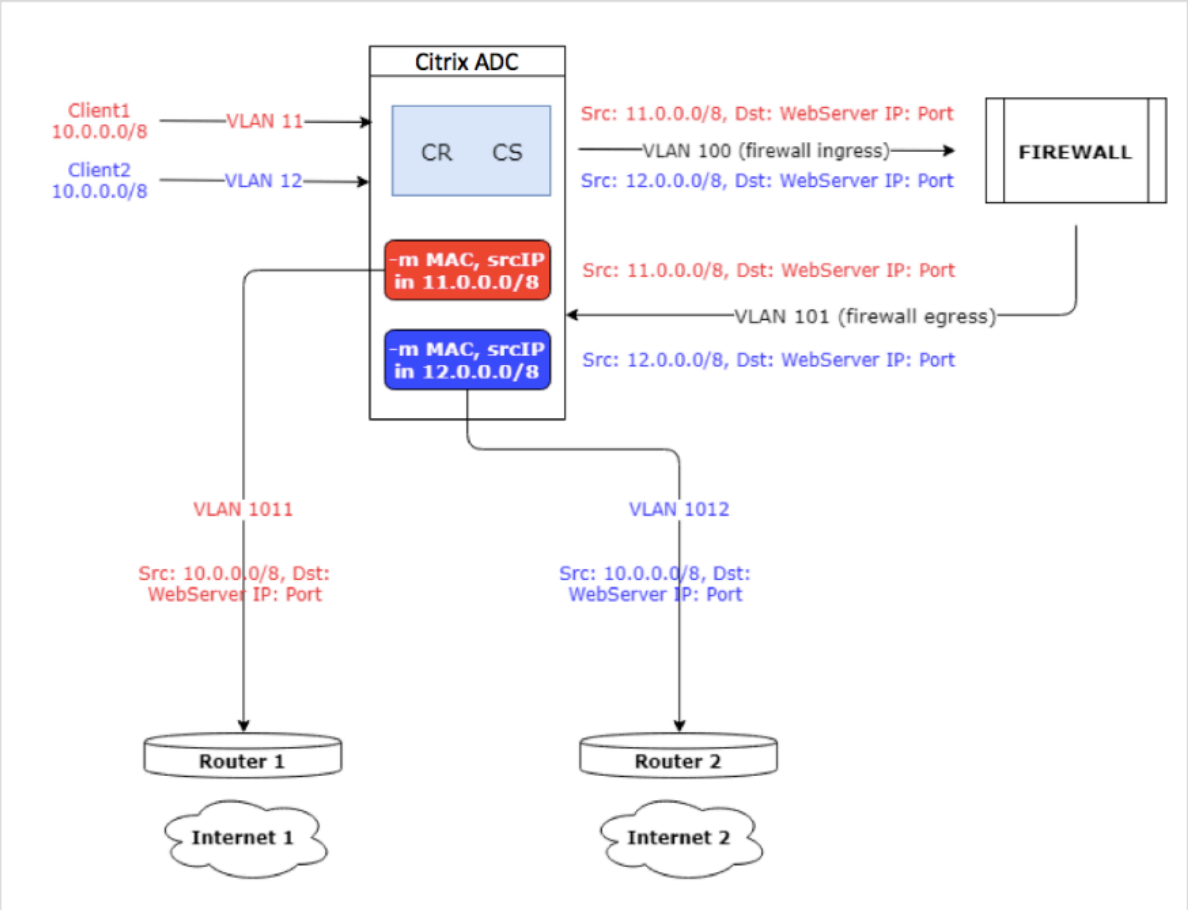
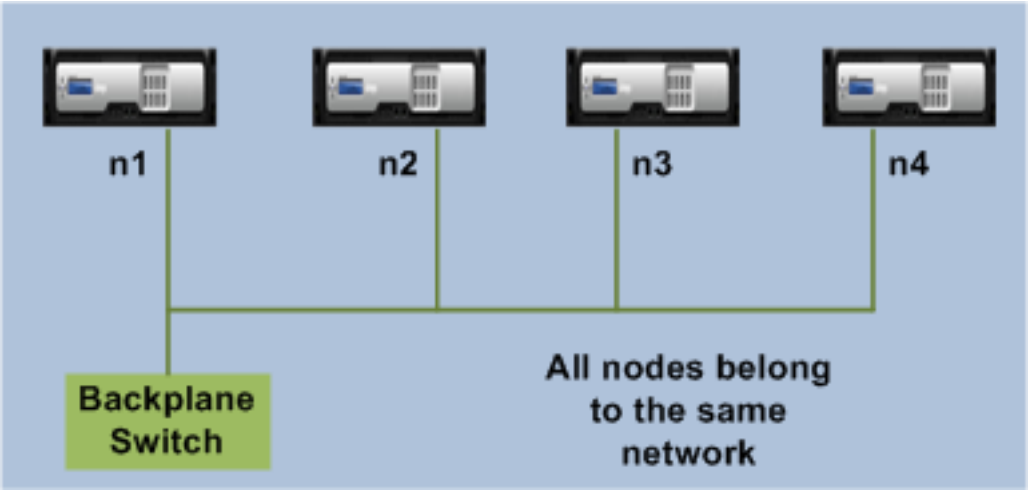
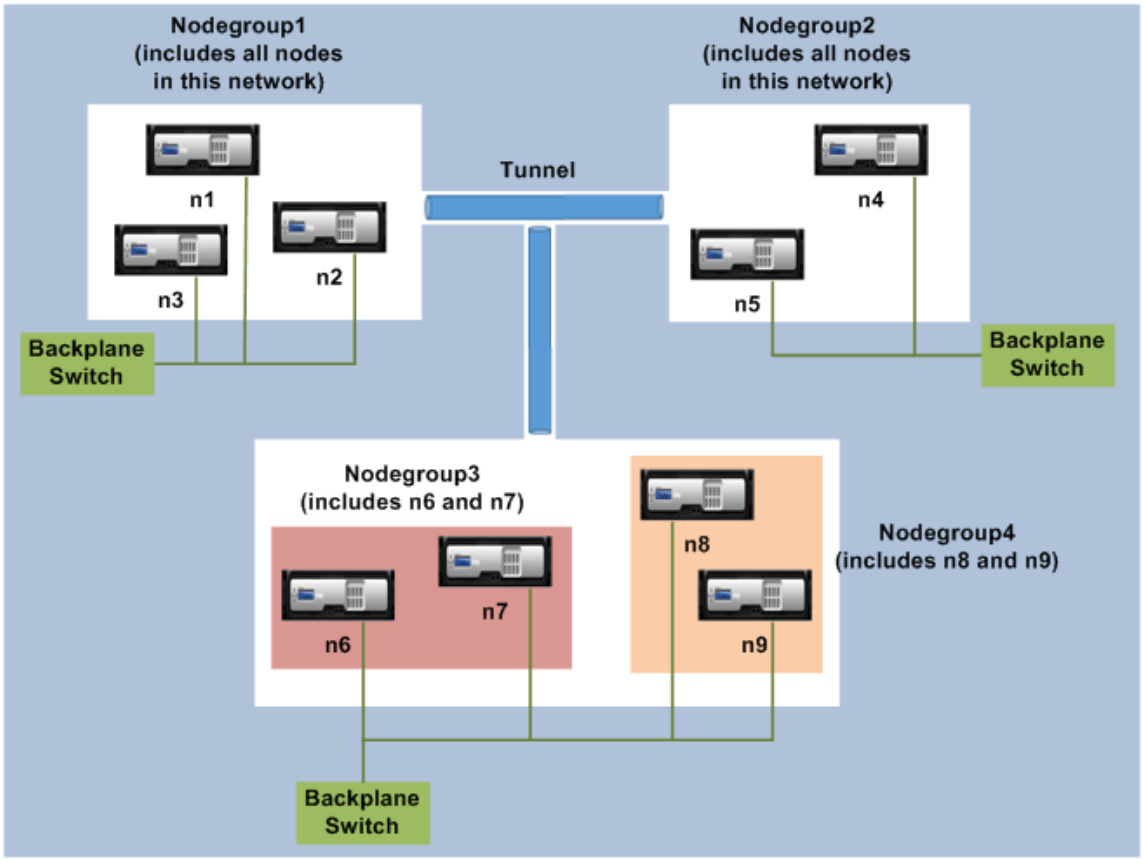
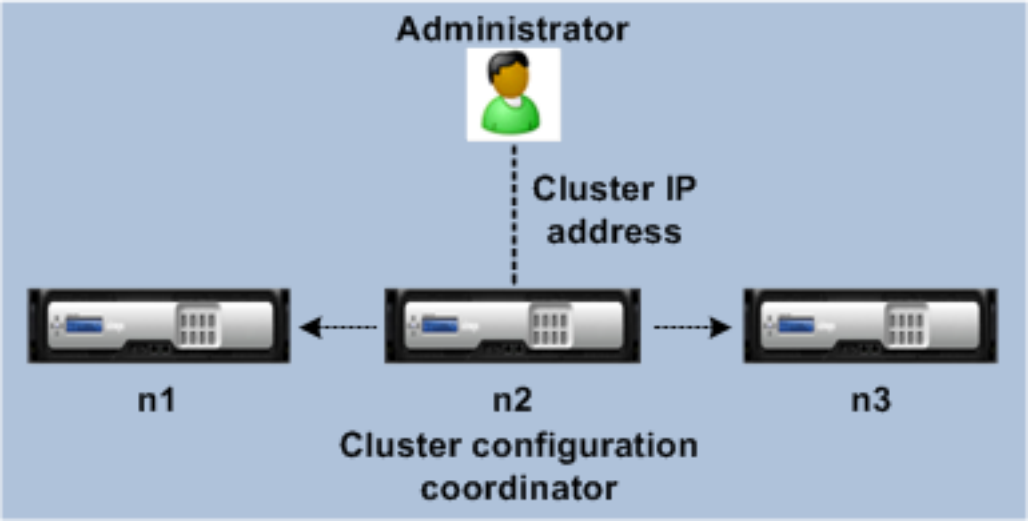
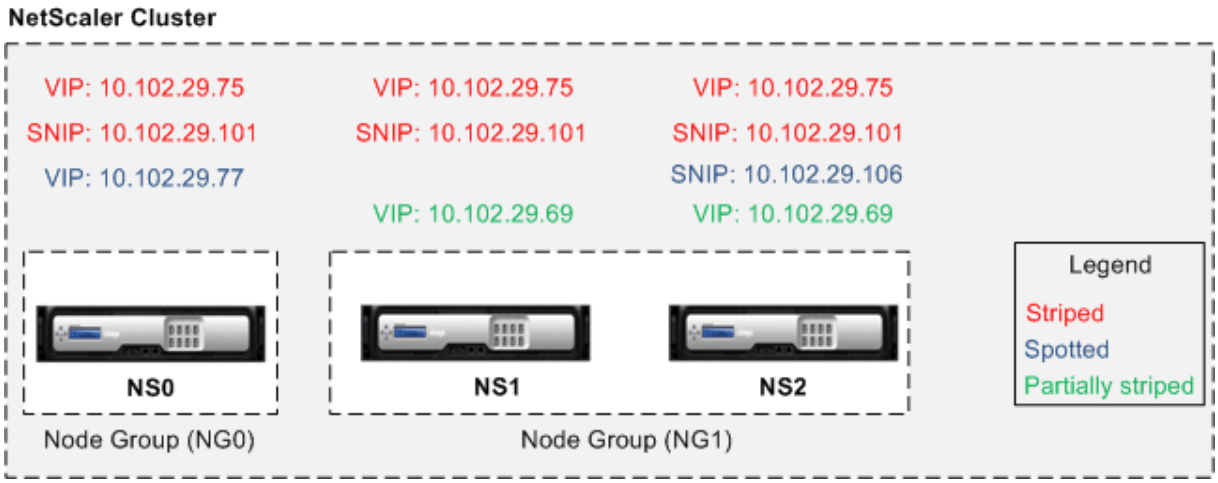
> disable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: OUT OF SERVICE  ARP:DISABLED
        Client Idle Timeout: 180 sec
```



```
24          Down state flush: ENABLED
25          Disable Primary Vserver On Down : DISABLED
26          Default:        Content Precedence: URL Cache: TRANSPARENT
27          On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
28          Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
29
30  1)      Cache bypass  Policy: bypass-cache-control
31  2)      Cache bypass  Policy: Policy-CRD
32   Done
33  >
34  <!--NeedCopy-->
```

Enable or disable a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the navigation pane, expand Cache Redirection, and then click Virtual Servers.
3. In the details pane, select the virtual server that you want to enable or disable, (for example, **Vserver-CRD-1**), and then click Statistics.
4. In the Proceed dialog box, click Yes.

Direct policy requests to cache instead of origin web server

September 14, 2021

By default, when a request matches a policy, the Citrix ADC appliance forwards the request either to the origin server directly, or to a load balancing virtual server for the origin, depending on how you have configured cache redirection.

You can change the default behavior so that when a request matches a policy, the request is forwarded to a load balancing virtual server for the cache.

To change the destination for a policy request to the origin or the cache, use the `onPolicyMatch` parameter, which specifies where to send requests that match the cache redirection policy.

The valid options are:

1. `CACHE` - Directs all matching requests to the cache.
2. `ORIGIN` - Directs all matching requests to the origin server.

> **Note:**
>
> For this option to work, you must select the cache redirection type as `POLICY`.

Possible values: CACHE, ORIGIN

Default value: ORIGIN

Change the destination for a policy request to the origin or the cache by using the CLI

At the command prompt, type the following commands to change the destination for a policy hit and verify the configuration:

```
set cr vserver <name> [-onPolicyMatch (ORIGIN | CACHE)]
<!--NeedCopy-->
```

```
show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Change the destination for a policy hit to the origin or the cache by using the GUI

1. Navigate to **Traffic Management > Cache Redirection > Virtual Servers**.
2. In the details pane, select the virtual server for which you want to change the destination for a policy request, (for example, **Vserver-CRD-1**), and then click Open.
3. In the **Configure Virtual Server (Cache Redirection)** dialog box, click **Advanced**.
4. Select **CACHE** or **ORIGIN** from the **Redirect To** drop-down list.
5. Click **OK**.

Back up a cache redirection virtual server

September 14, 2021

Cache redirection can fail if the primary virtual server fails, or if it is unable to handle excessive traffic. You can specify a backup virtual server to take over the processing of traffic when the primary virtual server fails.

To specify a backup cache redirection virtual server, use the backupVServer parameter, which specifies Backup Virtual Server. Maximum Length: 127

Specify a backup cache redirection virtual server by using the CLI

At the command prompt, type the following commands to specify a backup cache redirection virtual server and verify the configuration:

```
-  set cr vserver <name> [-backupVServer <string>]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

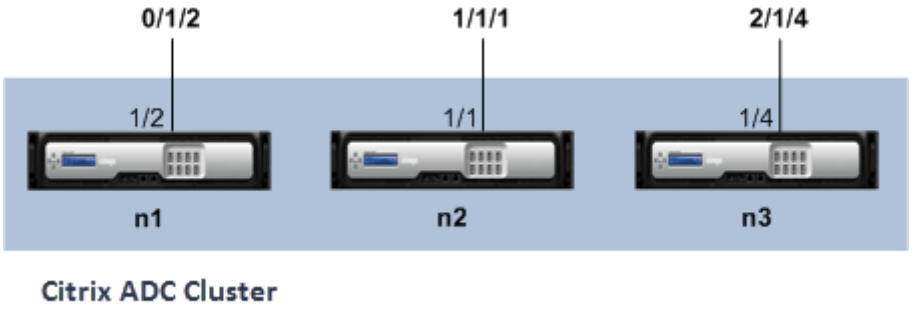
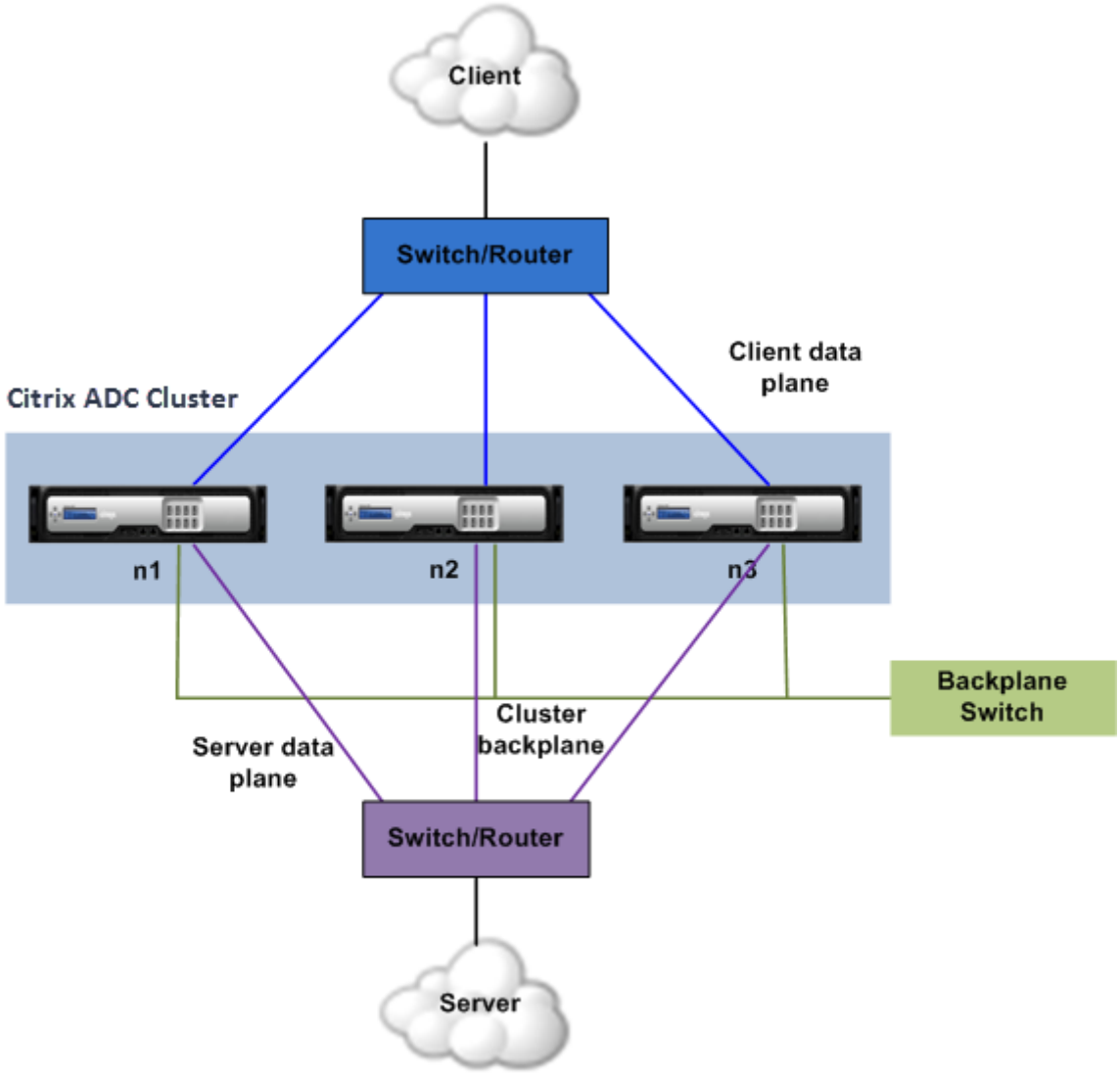
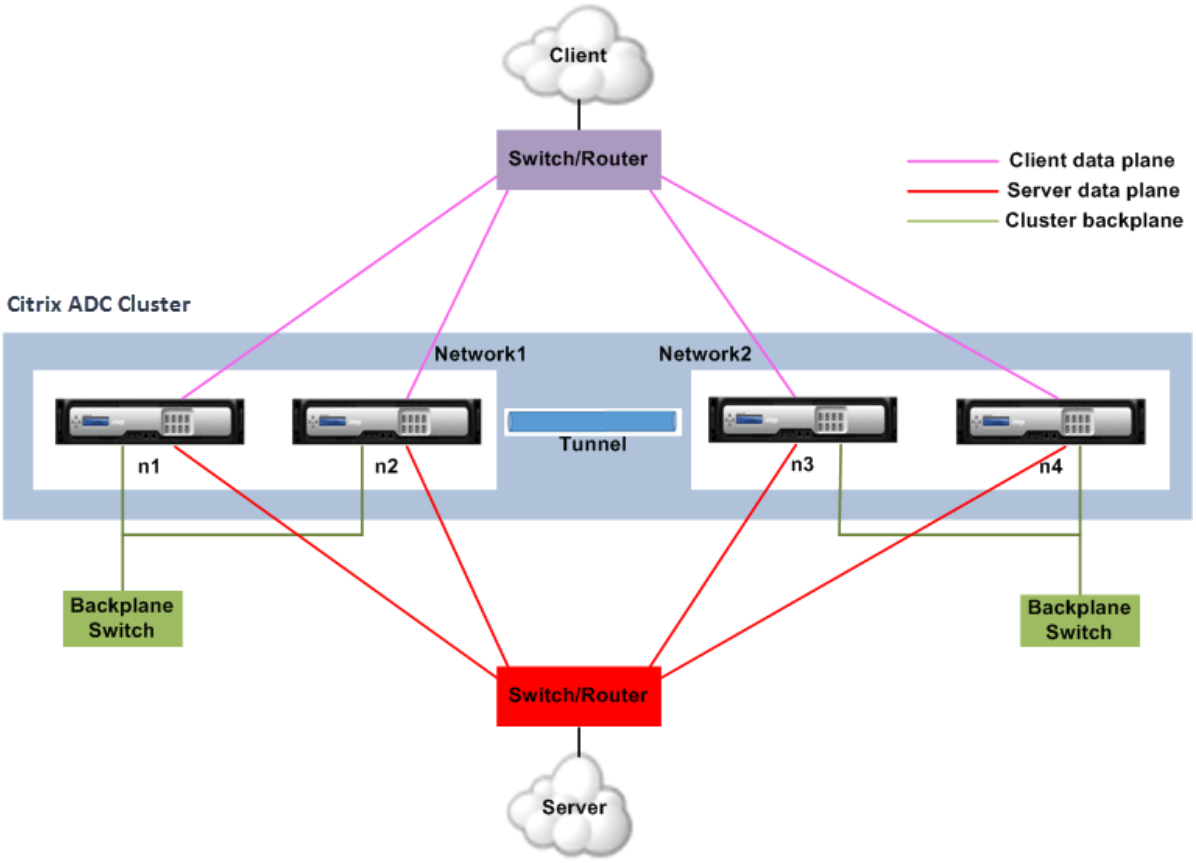
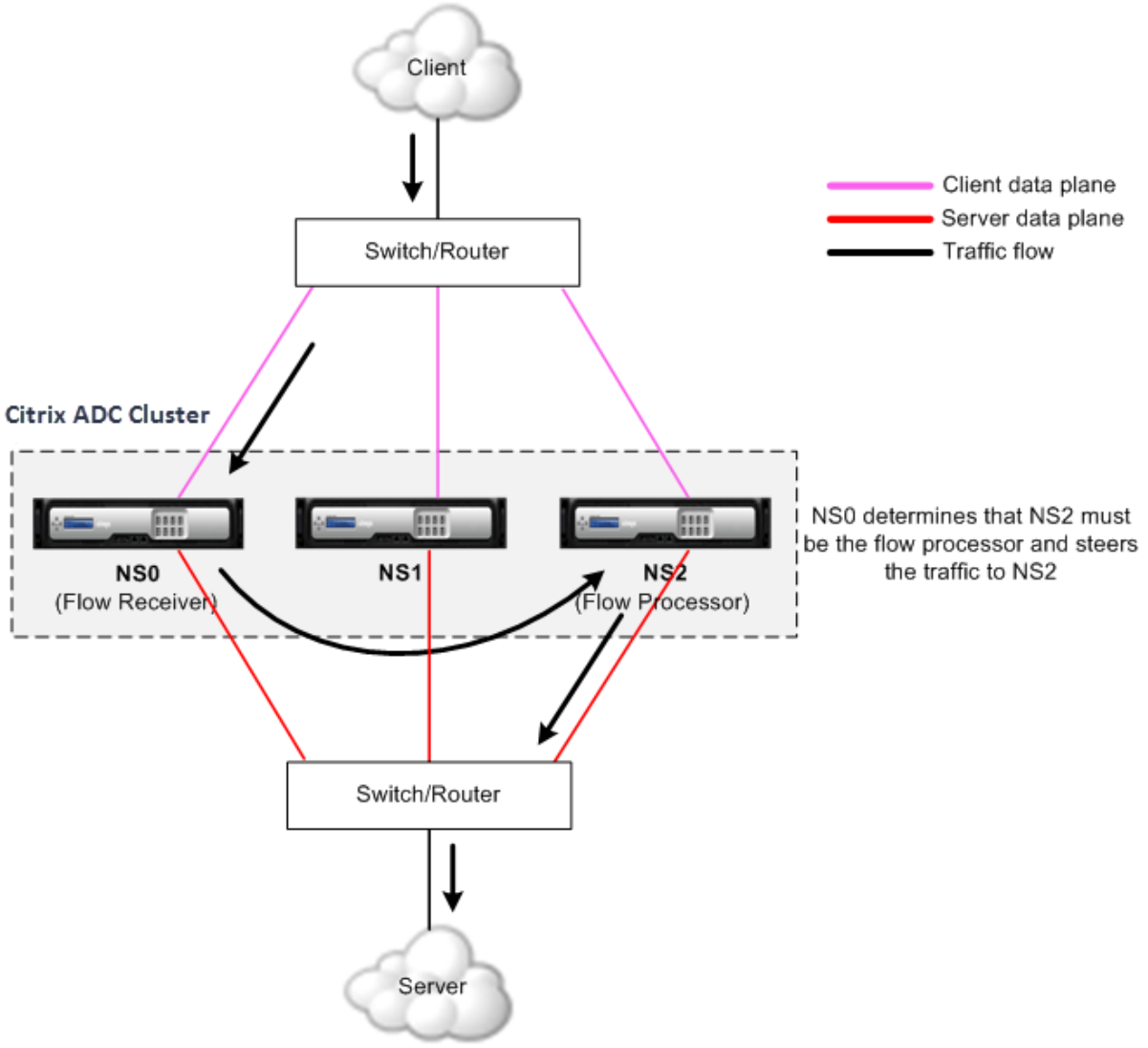
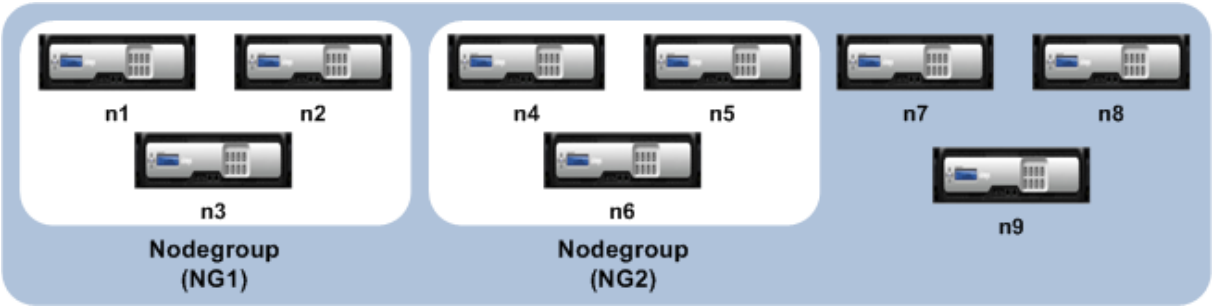
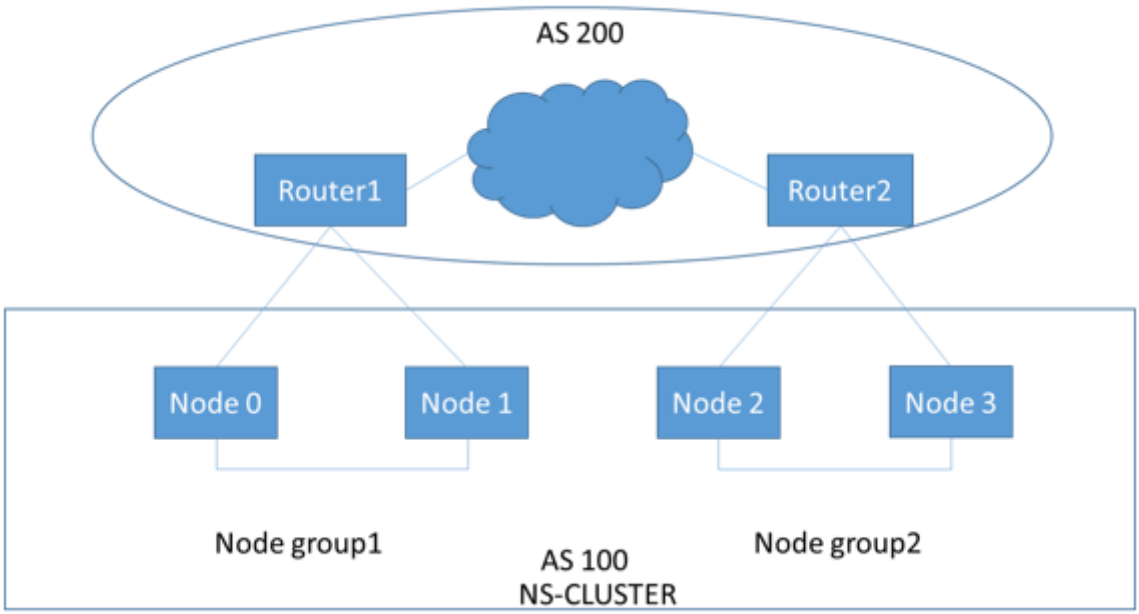
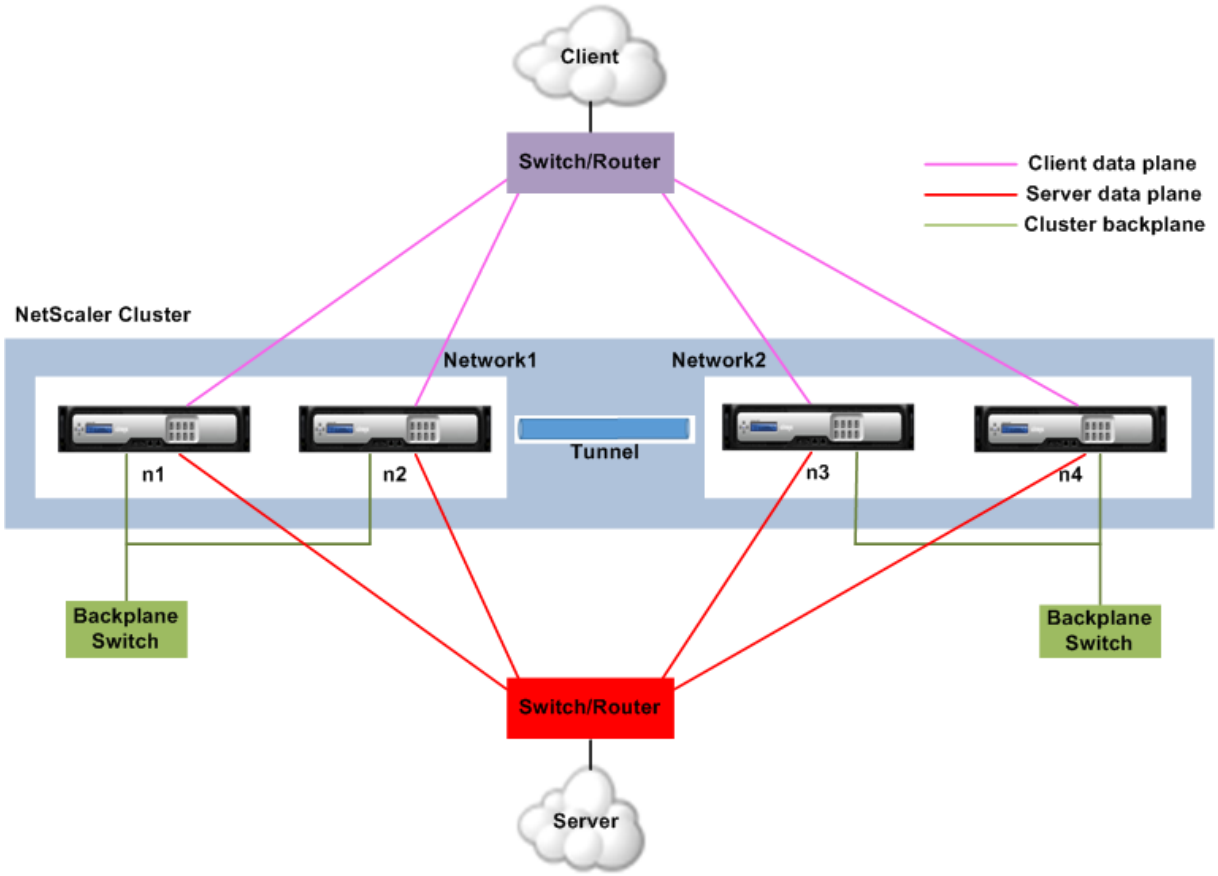
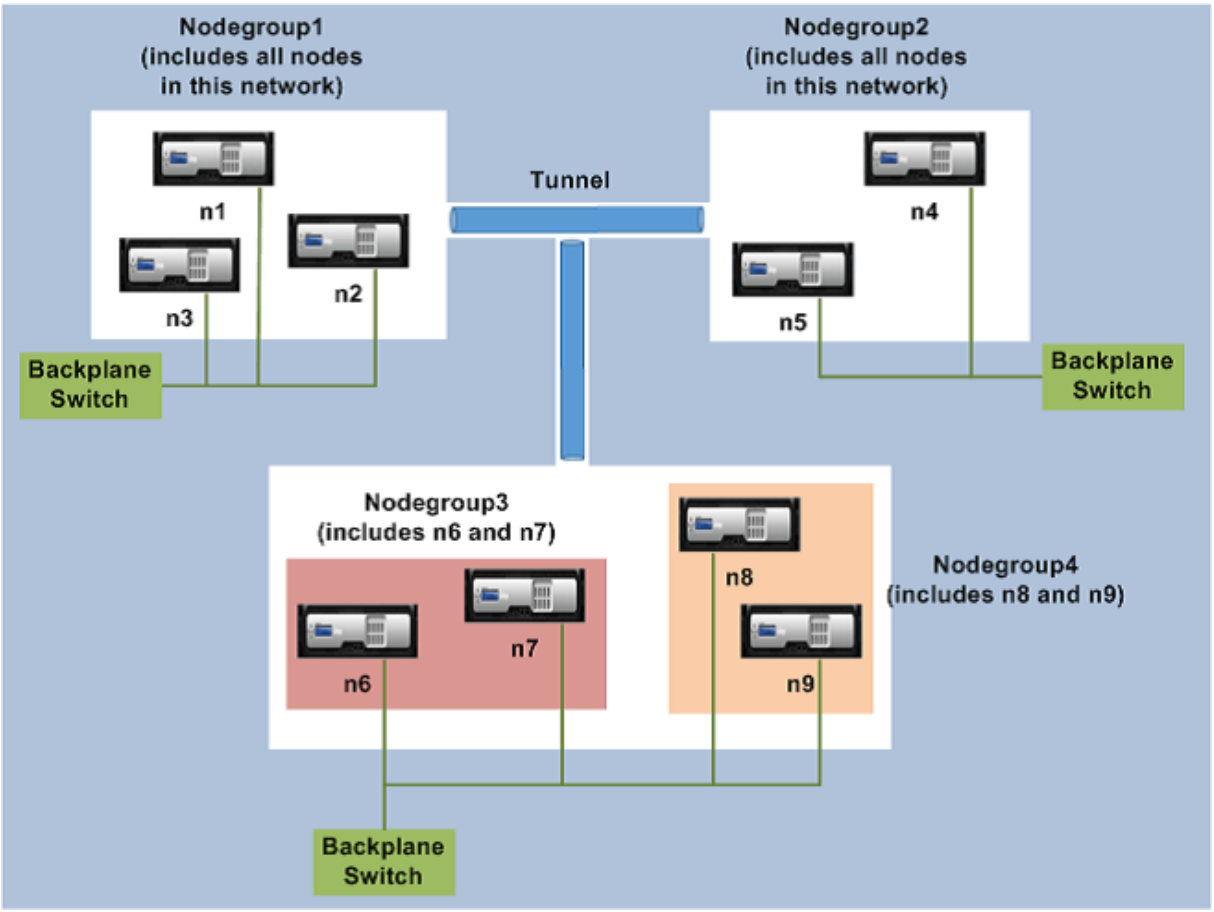
1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Specify a backup cache redirection virtual server by using the GUI

1. Navigate to **Traffic Management > Cache Redirection > Virtual Servers**.

2. In the details pane, select the virtual server for which you want to change the destination for a policy request, (for example, **Vserver-CRD-1**), and then click Open.
3. In the Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. In the Backup Virtual Server drop-down list, select the virtual server.
5. Click OK.

Manage client connections for a virtual server

September 14, 2021

You can configure timeouts on a cache redirection virtual server so that client connections are not kept open indefinitely. You can also insert Via headers in requests. To possibly reduce network congestion, you can reuse open TCP connections. You can enable or disable delayed cleanup of cache redirection virtual server connections.

You can configure the appliance to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.

The following settings can be made on a virtual server:

- When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, appliance always responds.
- When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, appliance responds even if one virtual server is UP.
- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, appliance responds even if one virtual server set to ACTIVE is UP.

This document includes the following information:

- Configure client timeout
- Insert Via headers in the requests
- Reuse TCP connections
- Configure delayed connection cleanup

Configure client timeout

You can specify expiration of client requests by setting a timeout value for the cache redirection virtual server. The timeout value is the number of seconds for which the cache redirection virtual server waits to receive a response for the client request.

To configure a time-out value, use the cltTimeout parameter, which specifies the time, in seconds, after which the Citrix ADC appliance closes any idle client connections. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

Configure client timeout by using the CLI

At the command prompt, type the following commands to configure client timeout and verify the configuration:

```
-  set cr vserver <name> [-cltTimeout <secs>]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Configure client timeout by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. In the Client Time-out(secs) text box, enter the time-out value in seconds.
5. Click OK.

Insert Via headers in the requests

A Via header lists the protocols and recipients between the start and end points for a request or a response and informs the server of proxies through which the request was sent. You can configure the cache redirection virtual server to insert a Via header in each HTTP request. The via parameter is enabled by default when you create a cache redirection virtual server.

To enable or disable Via-header insertion in client requests, use the via parameter, which specifies the state of the system in inserting a Via header in the HTTP requests.

Possible values: ON, OFF

Default value: ON

Enable or disable Via-header insertion in client requests by using the CLI

At the command prompt, type:

```
-  set cr vserver <name> [-via (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```


Enable or disable Via-header insertion in client requests by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. Select the Via check box.
5. Click OK.

Reuse TCP connections

You can configure the Citrix ADC appliance to reuse TCP connections to the cache and origin servers across client connections. This can improve performance by saving the time required to establish a session between the server and the appliance. The reuse option is enabled by default when you create a cache redirection virtual server.

To enable or disable the reuse of TCP connections, use the reuse parameter, which specifies the state of reuse of TCP connections to the cache or origin servers across client connections.

Possible values: ON, OFF

Default value: ON

Enable or disable the reuse of TCP connections by using the CLI

At the command prompt, type:

```
-  set cr vserver <name> [-reuse (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2
```



```
13
14 1)      Cache bypass  Policy: bypass-cache-control
15 2)      Cache bypass  Policy: Policy-CRD
16  Done
17 <!--NeedCopy-->
```

Enable or disable the reuse of TCP connections by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. Select the Reuse check box.
5. Click OK.

Configure delayed connection cleanup

The down state flush option performs delayed cleanup of connections on a cache redirection virtual server. The down state flush option is enabled by default when you create a cache redirection virtual server.

To enable or disable the down state flush option, set the downStateFlush parameter.

Possible values: ENABLED, DISABLED

Default value: ENABLED

Enable of disable the down state flush option by using the CLI

At the command prompt, type the following commands to configure delayed connection clean up and verify the configuration:

```
1 -  set cr vserver <name> [-downStateFlush (ENABLED | DISABLED)]
2 -  show cr vserver <name>
3 <!--NeedCopy-->
```

Example:

```
1 > set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
2  Done
3 > show cr vserver Vserver-CRD-1
4         Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
5         State: UP  ARP:DISABLED
6         Client Idle Timeout: 6000 sec
```



```
7          Down state flush: ENABLED
8          Disable Primary Vserver On Down : DISABLED
9          Default:        Content Precedence: URL Cache: TRANSPARENT
10         On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
11         Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
12         Backup: Vserver-CRD-2
13
14 1)      Cache bypass  Policy: bypass-cache-control
15 2)      Cache bypass  Policy: Policy-CRD
16  Done
17 <!--NeedCopy-->
```

Enable or disable the reuse of TCP connections by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, click Advanced tab.
4. Select the Down state flush check box.
5. Click OK.

Enable external TCP health check for UDP virtual servers

September 14, 2021

In public clouds, you can use the Citrix ADC appliance as a second-tier load balancer when the native load balancer is used as a first tier. The native load balancer can be an application load balancer (ALB) or a network load balancer (NLB). Most of the public clouds do not support UDP health probes in their native load balancers. To monitor the health of the UDP application, public clouds recommend adding a TCP-based endpoint to your service. The endpoint reflects the health of the UDP application.

The Citrix ADC appliance supports external TCP-based health check for a UDP virtual server. This feature introduces a TCP listener on the VIP of the cache redirection virtual server and the configured port. The TCP listener reflects the status of the virtual server.

To enable external TCP health check for UDP virtual servers by using CLI

At the command prompt, type the following command to enable an external TCP health check with the tcpProbePort option:


```
add cr vserver <name> <serviceType> -tcpProbePort <tcpProbePort>

<!--NeedCopy-->
```

Example:

```
add cr vserver Vserver-CR-1 HTTP -tcpProbePort 80
<!--NeedCopy-->
```

To enable external TCP health check for UDP virtual servers by using GUI

1. Navigate to **Traffic Management > Cache Redirection > Virtual Servers**, and then create a virtual server.
2. Click **Add** to create a virtual server.
3. In the **Basic Settings** pane, add the port number in the **TCP Probe Port** field.
4. Click **OK**.

N-Tier cache redirection

September 14, 2021

To efficiently handle large amounts of cached data, typically several gigabytes per second, an Internet Service Provider (ISP) deploys several dedicated cache servers. The cache redirection feature of the Citrix ADC appliance can help load balance the cache servers, but a single appliance or a couple of appliances might not efficiently handle the large volume of traffic.

You can solve the problem by deploying the Citrix ADC appliances in two tiers (layers), where the appliances in the upper tier load balance those in the lower tier and the appliances in the lower tier load balance the cache servers. This arrangement is called *n-tier cache redirection*.

For purposes such as auditing and security, an ISP has to track client details such as the IP address, information provided, and the time of the interaction. Therefore, client connections through a Citrix ADC appliance have to be fully transparent. However, if you configure transparent cache redirection, with the Citrix ADC appliances deployed in parallel, the IP address of the client has to be shared among all the appliances. Sharing of the client IP address creates a conflict that makes network devices, such as routers, cache servers, origin servers, and other Citrix ADC appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

How N-tier cache redirection Is implemented

To solve the problem, appliance n-tier cache redirection splits the source port range among the appliances in the lower tier and includes the client IP address in the request sent to the cache servers. The upper-tier Citrix ADC appliances are configured to do sessionless load balancing in order to avoid unnecessary load on the appliances.

When the lower-tier Citrix ADC appliance communicates with a cache server, it uses a mapped IP address (MIP) to represent the source IP address. Therefore, the cache server can identify the appliance from which it received the request and send the response to the same appliance.

The lower-tier Citrix ADC appliance inserts the client IP address into the header of the request sent to the cache server. The client IP in the header helps the appliance to determine the client to which the packet should be forwarded when it receives the response from a cache server, or the origin server in case of a cache miss. The origin server determines the response to be sent according to the client IP inserted in the request header.

The origin server sends the response to an upper-tier appliance, including the source port number from which the origin server received the request. The entire source port range, 1024 to 65535, is distributed among the lower-tier Citrix ADC appliances. Each lower-tier appliance is exclusively assigned a group of addresses within the range. This allotment enables the upper-tier appliance to unambiguously identify the lower-tier Citrix ADC appliance that sent the request to the origin server. The upper-tier appliance can therefore forward the response to the correct lower-tier appliance.

The upper-tier Citrix ADC appliances are configured to do policy-based routing, and the routing policies are defined to determine the IP address of the destination appliance from the source port range.

Setup necessary for configuring N-Tier CRD

The following setup is necessary for the functioning of n-tier cache redirection:

For each upper-tier Citrix ADC appliance:

- Enable Layer 3 mode.
- Define policies for policy-based routes (PBRs) so that traffic is forwarded according to the range of the destination port.
- Configure a load balancing virtual server.
- Configure the virtual server to listen to all the traffic coming from the client. Set the Service Type/Protocol to be ANY and IP Address as asterisk (*).
- Enable sessionless load balancing with MAC-based redirection mode to avoid unnecessary load on the upper-tier Citrix ADC appliances.
- Make sure that the Use Proxy Port option is enabled.
- Create a service for each lower-tier appliance and bind all the services to the virtual server.

For each lower-tier Citrix ADC appliance,

- Configure the cache redirection port range on the appliance. Assign an exclusive range to each lower-tier appliance.
- Configure a load balancing virtual server and enable MAC-based redirection.
- Create a service for each cache server that is to be load balanced by this appliance. When creating the service, enable insertion of client IP in the header. Then, bind all the services to the load balancing virtual server.
- Configure a transparent mode cache redirection virtual server with the following settings:
  - Enable the Origin USIP option.
  - Add a source IP expression to include the client IP in the header.
  - Enable the Use Port Range option.

How N-tier cache redirection works during a cache hit

The following figure shows how cache redirection works when a client request is cacheable and the response is sent from a cache server.

Figure 1. Cache Redirection in Case of a Cache Hit



Two Citrix ADC appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four Citrix ADC appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1, and the request is cacheable. L2NS2 includes the client IP in the request header.
4. CRS1 sends the response to L2NS2 because L2NS2 used its MIP as the source IP address when connecting to CRS1.
5. With the help of the client IP address in the request header, L2NS2 identifies the client from which the request came. L2NS2 directly sends the response to the router, avoiding unnecessary load on the appliance in the upper tier.
6. The router forwards the response to Client A.

How N-tier cache redirection works during a cache bypass

The following figure shows how cache redirection works when a client request is sent to an origin server for a response.

Figure 2. Cache Redirection in Case of a Cache Bypass



Two Citrix ADC appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four Citrix ADC appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. The request is uncacheable (cache bypass). Therefore, L2NS2 sends the request to the origin server through the router.
4. The origin server sends the response to an upper-tier appliance, L1NS2.
5. According to the PBR policies, L1NS2 forwards the traffic to the appropriate appliance in the lower tier, L2NS2.
6. L2NS2 uses the client IP address in the request header to identify the client from which the request came and sends the response directly to the router, avoiding unnecessary load on the appliance in the upper tier.

7. The router forwards the response to Client A.

How N-tier cache redirection works during a cache miss

The following figure shows how cache redirection works when a client request is not cached.

Figure 3. Cache Redirection in Case of a Cache Miss



Two Citrix ADC appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four Citrix ADC appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1 because the request is cacheable.

4. CRS1 does not have the response (cache miss). CRS1 forwards the request to the origin server through the appliance in the lower tier. L2NS3 intercepts the traffic.
5. L2NS3 takes the client IP from the header and forwards the request to the origin server. The source port included in the packet is the L2NS3 port from which the request is sent to the origin server.
6. The origin server sends the response to an upper-tier appliance, L1NS2.
7. According to the PBR policies, L1NS2 forwards the traffic to the appropriate appliance in the lower tier, L2NS3.
8. L2NS3 forwards the response to the router.
9. The router forwards the response to Client A.

Configure the upper-tier Citrix ADC appliances

September 14, 2021

Configure each of the upper-tier Citrix ADC appliances as follows.

Configure an upper-tier appliance for n-tier cache redirection by using the command CLI

At the command prompt, type the following commands:

- `add service \<name\>@ \<serviceIP\> \<serviceType\> \<port\>`

  Run this command for each service to be added.

- `add lb vserver \<name\>@ ANY \* \<port\> -persistenceType \<persistenceMethod \> -lbMethod \<lbMethod\> -m MAC -sessionless ENABLED -cltTimeout \< client\\_Timeout\\_Value\>`

- `bind lb vserver \<name\>@ \<serviceName\>`

  Run this command for each service to be bound.

- `enable ns mode l3`

- `add ns pbr \<name\> \<action\> -srcPort \<sourcePortNumber\> -destPort \<startPortNumber-endPortNumber\> -nexthop \<serviceIpAddress\> -protocol TCP`

- `apply ns pbrs`

  Run this command after adding all the necessary PBRs.

Configure an upper-tier appliance for n-tier cache redirection by using the GUI

1. Enable L3 mode:
   a) In the navigation pane, click System, and then click Settings.
   b) In the Settings group, click the Configure modes link.
   c) Select the Layer 3 Mode (IP Forwarding) check box.
   d) Click OK.
2. Configure policy-based routing (PBR):
   a) Navigate to System > Network > PBRs.
   b) In the Policy-Based Routing (PBRs) pane, click Add.
   c) Type a name for the PBR.
   d) Select the action as Allow.
   e) In the Next Hop box, type the IP address of the service, which represents a lower-tier appliance.
   f) Select TCP from the Protocol drop-down list.
   g) Type the source port and the range of the destination port corresponding to the lower-tier appliance being added.
   h) Click Create.
   i) In the details pane, select the PBR and click Apply.
   j) Repeat Step (i) to Step (vii) for each lower-tier appliance.
3. Create a service for each lower-tier appliance:
   a) Navigate to Traffic Management > Load Balancing > Services.
   b) In the details pane, click Add.
   c) Specify the name, protocol, IP address, and port. The protocol should be ANY.
   d) Click Create.
4. Configure a load balancing virtual server:
   a) Navigate to Traffic Management > Load Balancing > Virtual Servers.
   b) In the details pane, click Add.
   c) Specify the name, protocol, IP address, and port. The protocol should be ANY and the IP address should be *.
   d) In the Services tab, select the services that represent the lower-tier Citrix ADC appliances.
   e) In the Advanced tab, select the Redirection Mode as MAC Based and select the Sessionless check box.
   f) Click Create.

Configure the lower-tier Citrix ADC appliances

September 14, 2021

Configure each of the lower-tier Citrix ADC appliances as follows.

Configure a lower-tier appliance for n-tier cache redirection by using the CLI

At the command prompt, type the following commands:

- `add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP"-cachetype transparent`

  Repeat for each cache server.

- `add lb vserver <name>@ <serviceType> -m MAC`
- `bind lb vserver <name>@ <cacheServiceName>`

  Repeat for each cache server.

- `add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ. HEADER("ClientIP")"-originusip ON -usePortRange ON`
- `set ns param-crPortRange <startPortNumber-endPortNumber>`

Configure a lower-tier appliance for n-tier cache redirection by using the GUI

1. Create a service for each cache server. To create a service:
   a) Navigate to Traffic Management > Load Balancing > Services.
   b) In the details pane, click Add, and specify the name and protocol. Clear the Directly Addressable check box.
   c) In the Advanced tab, select the Override Global check box and the Client IP check box, and then in the Header box, type ClientIP.
   d) In the Cache Type box, select Transparent Cache.
   e) Click Create.
2. Configure a load balancing virtual server:
   a) Navigate to Traffic Management > Load Balancing > Virtual Services.
   b) In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be an asterisk (*).
   c) In the Services tab, select the services that represent the cache servers.
   d) In the Advanced tab, for Redirection Mode, select MAC Based.
   e) Click Create.
3. Configure a cache redirection virtual server:
   a) Navigate to Traffic Management > Load Balancing > Virtual Services.
   b) In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be *.
   c) For Cache Type, select Transparent.

d) On the Advanced tab, in the Cache Server box, select the new load balancing virtual server and check the Origin USIP and Use Port Range check boxes. In the Source IP Expression box, type HTTP.REQ.HEADER("ClientIP").
e) Click Create.

4. Assign a source port range for the appliance:
a) In the navigation pane, click System, and then click Settings.
b) In the Settings group, click the Change global system settings link.
c) In the Cache Redirection Port Range group, specify the port range for the appliance by typing a port number for Start Port and a port number for End Port.
d) Click OK.

Translate destination IP address of a request to origin IP address

September 14, 2021

You can configure the forward proxy cache redirection virtual server on the Citrix ADC appliance to translate the destination IP address of the request landing on the cache redirection virtual server to the origin server IP address. This translation occurs irrespective of whether the request is sent to the cached servers or the origin server.
Previously, forward proxy cache redirection virtual server in service provider environment could not be effectively used to send traffic across firewall because of the limitations in cache redirection using content switching policies. The cache redirection virtual server did not translate the origin IP address into the destination IP when the packet was sent to cache. The destination IP address was that of the origin server only when the requests were served from cached server.

Note: Translation of destination IP address of a request to origin IP address is not supported for a transparent cache redirection virtual server. For a transparent cache redirection virtual server, this option must be set to OFF.

Use case

In a deployment that has Citrix ADC appliance configured for forward proxy cache redirection, firewall, and reused client IP addresses, firewall cannot distinguish/use the reused IP addresses. Therefore, these reused IP addresses must be translated to different IP addresses. To translate the reused IP addresses, the Citrix ADC appliance must perform the following:

1. Query a DNS load balancing virtual server for resolution of the destination.
2. Update the origin IP address and port number in the destination.
3. Send the request back to the firewall.

Consider the following deployment that has a Citrix ADC appliance configured for forward proxy cache redirection, firewall, two routers (Router 1 and Router 2). Network traffic flows to Internet 1 through Router 1 and to Internet 2 through Router 2 respectively.



In this example, input requests from clients come from two different VLANs, VLAN11 or VLAN12. The client IP address (10.0.0.0) is reused.
Based on the cache redirection and content switching policies, the request can go directly to the origin server or to the firewall.

- If the request has to bypass the firewall and go to the internet, then based on the input request VLAN, either Router 1 or Router 2 is selected and the request is sent to Internet 1 or Internet 2.
- If the request has to go through the firewall, then the source IP of the request must be translated to specific IP address. The translated IP address can be used to identify the VLAN through which request has come. For example, if the input request is coming from VLAN11, then the source IP address is translated to 11.x.x.x. If the request is coming from VLAN12, then the source IP address is translated to 12.x.x.x.

After the firewall processes the request, the request is sent back to the appliance. Using the combination of listen policy and net profiles, the appliance then translates the source IP address back to the

original IP address and sends the request to Router 1 or Router 2 based on the input VLAN ID.

Note: The mode of the load balancing virtual server that is bound to the cache must always be set to MAC mode. Though IP mode for this feature is not blocked, setting to IP mode leads to unexpected behavior.

To translate the destination IP address and port number of the request to origin IP address by using the CLI

At the command prompt, type;

```
set cr vserver <vsname> -useoriginIpPortForCache <YES|NO>
<!--NeedCopy-->
```

Example:

```
set cr vserver cvsrv1 -useoriginIpPortForCache YES
<!--NeedCopy-->
```

When useoriginIpPortForCache is set to Yes and if the request must be served from the cached servers, then the request's destination IP is translated to the origin server IP address.

Note: If useoriginIpPortForCache is enabled, always set the load balancing virtual server that is bound to the cache to MAC mode.

To translate the destination IP address and port of the request to origin IP address by using the GUI

1. Navigate to **Traffic Management** > **Cache Redirection** > **Virtual Servers** and click **Add**.
2. Specify the details of the cache redirection virtual server.
3. Select **Use Origin IP Port** for cache to enable translation of the destination IP address of the request to origin IP address.
4. Click **OK**.

Clustering

September 14, 2021

> **Note**
>
> This feature is available with a Citrix ADC Advanced or Premium edition license.

A Citrix ADC cluster is a group of nCore appliances working together as a single system image. Each appliance of the cluster is called a node. The cluster can have one appliance or as many as 32 Citrix ADC nCore hardware or virtual appliances as nodes.

The client traffic is distributed between the nodes to provide high availability, high throughput, and scalability.

To create a cluster, you must perform the following steps:

- Add the appliances as cluster nodes.
- Set up communication between the nodes.
- Set up links to the client and server networks.
- Configure the appliances, and configure the distribution of client and server traffic.

Supportability matrix for Citrix ADC cluster

September 14, 2021

Clustering in the Citrix ADC appliance supports a wide-spread of features in Citrix ADC configurations.

The following table lists the Citrix ADC features and provides the supportability status across different Citrix ADC releases of cluster setups. The supportability status of some Citrix ADC features in a 13.0 Citrix ADC BLX cluster is different than a 13.0 Citrix ADC non-BLX (MPX, or VPX, SDX ADC) cluster.

> **Important**
>
> The “Node-level” entry in the table indicates that the feature is supported only on individual cluster nodes.

Citrix ADC features	11.1	12.1	13.0	13.0 Citrix ADC BLX cluster
SSL FIPS	No	No	No	No
SSL Certificate Bundle	No	No	No	No
SSL interception	NA	No	No	No
Content switching actions	Yes	Yes	Yes	Yes

Citrix ADC features	11.1	12.1	13.0	13.0 Citrix ADC BLX cluster
Policy-based logging for content switching policies	Yes	Yes	Yes	Yes
Rate limiting	Yes	Yes	Yes	Yes
Action analytics	Yes	Yes	Yes	No
GSLB	Yes	Yes	Yes	Yes
RTSP	Yes	Yes	Yes	Yes
DNSSEC	No	No	No	No
DNS64	No	No	No	No
FTP	Yes	Yes	Yes	No
TFTP	No	Yes	Yes	Yes
Connection mirroring	No	No	No	no
Integrated caching	Node-Level	Node-Level	Node-Level	No
Large shared cache	Node-Level	Node-Level	Node-Level	No
Front end optimization	Node-Level	Node-Level	Node-Level	No
Application firewall	Yes	Yes	Yes	No
HTTP Denial-of-Service Protection (HDOSP)	Node-Level	Node-Level	Node-Level	Deprecated
Priority queuing (PQ)	Node-Level	Node-Level	Node-Level	Deprecated
Sure connect (SC)	Node-Level	Node-Level	Node-Level	Deprecated
AppQoE	Yes	Yes	Yes	No

Citrix ADC features	11.1	12.1	13.0	13.0 Citrix ADC BLX cluster
Surge protection	Node-Level	Node-Level	Node-Level	Yes
MPTCP	Yes	Yes	Yes	No
Striped SNIPs	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.
MSR	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.
IS-IS (IPv4 and IPv6)	Yes	Yes	Yes	No
Jumbo Frames	Yes	Yes	Yes	No
IP-IP tunneling	Yes	Yes	Yes	No
Link load balancing	Yes	Yes	Yes	Yes
FIS (Failover Interface Set)	Yes	Yes	Yes	No
Link redundancy (LR)	Yes	Yes	Yes	No
NAT46	No	No	Yes	Yes
NAT64	No	No	Yes	Yes
RNAT6	Yes	Yes	Yes	Yes
LSN/CGNAT	No	Yes	Yes	No
IPv6 ReadyLogo	No	Yes	Yes	No
Traffic domains	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	No

Citrix ADC features	11.1	12.1	13.0	13.0 Citrix ADC BLX cluster
Route monitor	Yes; Only with DR.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes **Note:** Supported in L2 clusters. Not supported in L3 clusters.	No
GRE tunneling (CB)	No	No	No	No
Layer 2 mode	Yes	Yes	Yes	No
Net profiles	Yes	Yes	Yes	No
HTTPS callout	Yes	Yes	Yes	Yes
AAA-TM	Node-level	Yes	Yes	No
AppFlow	Node-level	Node-Level	Node-Level	No
Web Insight	Yes	Yes	Yes	No
HDX Insight	Yes	Yes	Yes	No
VMAC/VRRP	Yes	Yes	Yes	No
NetScaler Push	No	No	No	No
Stateful Connection Failover	No	No	No	No
Graceful Shutdown	No	Yes	Yes	Yes
DBS Autoscale	No	No	Yes	Yes
DSR using TOS	No	No	No	Yes
Finer Startup-RR Control	Node-Level	Node-Level	Node-Level	No
XML XSM	No	No	No	No
DHCP RA	No	No	No	No
Bridge Group	Yes	Yes	Yes	No
Network Bridge	No	No	No	No

Citrix ADC features	11.1	12.1	13.0	13.0 Citrix ADC BLX cluster
Web Interface on Citrix ADC (WIonNS)	Yes	Yes	Yes	No
EdgeSight Monitoring	Deprecated	Deprecated	Deprecated	No
Metrics tables - Local	No	No	No	No
DNS Caching	Node-Level	Node-Level	Node-Level	Node-Level
Call Home	Node-Level	Node-Level	Node-Level	No
Citrix Gateway ICA Proxy mode	Yes	Yes	Yes	No
Citrix Gateway (SSL VPN / full VPN and clientless VPN)	Node-Level	Node-Level	Node-Level	No
Citrix CloudBridge Connector	No	Yes	Yes	No
Policy Based Routing (PBR/PBR6)	Yes	Yes	Yes	No
IPv4 Policy Based Routing (PBR) with LLB virtual server as next hop	No	No	Yes	No
IPv6 Policy Based Routing (PBR6) with LLB virtual server as next hop	No	No	No	No
Subscriber awareness	No	No	No	No

Citrix ADC features	11.1	12.1	13.0	13.0 Citrix ADC BLX cluster
Dynamic Routing	Yes with v6 protocols (ospfv3, RIPng, ISIS6, BGP6) support	Yes with v6 protocols (ospfv3, RIPng, ISIS6, BGP6) support	Yes with v6 protocols (ospfv3, RIPng, ISIS6, BGP6) support	Yes
SYSLOG-TCP, load balancing of syslog servers, SNIP support, and FQDN support for syslog	Yes; **Note:** Supported from NetScaler 11.1 Build 54.16 onwards.	Yes	Yes	Yes
Bot management	No	No	Yes	No
VXLAN	No	No	No	No

Also, the following Citrix ADC configurations are supported:

Load balancing, load balancing persistency, DNS load balancing, SIP, maxClient, Spillover (connection and dynamic). Spillover based on bandwidth, DataStream, Compression control, Content filtering, TCP buffering, Cache redirection, distributed denial-of-service (DDoS). Client Keep-alive, Basic networking (IPv4 and IPv6), OSPF (IPv4 and IPv6), RIP (IPv4 and IPv6), RIP (IPv4 and IPv6). VLAN, ICMP, Fragmentation, MBF, ACL, Simple ACL, MSR, Path MTU discovery, IP-IP, SNMP, Policies (classic and advanced). Rewrite, Responder, HTTP callout, Web server logging, Audit logging (NSLOG and syslog). USIP, Location commands, HTML injection, NITRO API, AppExpert, KRPC.

Prerequisites

September 14, 2021

Citrix ADC appliances (MPX, VPX, SDX ADC, BLX) that are to be added to a cluster must meet the following prerequisites:

- All appliances must have the same software version and build.
- All appliances must be of the same platform type. This means that a cluster must have either

all hardware appliances (Citrix ADC MPX) or all Citrix ADC VPX appliances, or all Citrix BLX appliances, or all Citrix SDX ADC instances.

Note:

- For a cluster of hardware appliances (MPX), the appliances must be of the same model type.
- For the formation of the heterogeneous cluster, all appliances must be of MPX platform type.
- For a cluster of virtual appliances (VPX), the appliances must be deployed on the following hypervisors: XenServer, Hyper-V, VMware ESX, and KVM.
- For setting up a cluster of SDX Citrix ADC instances, see Set up a cluster of Citrix ADC instances.
- Jumbo frames are supported on a Citrix ADC cluster that is made up of Citrix ADC SDX instances.
- You can create L3 clusters of SDX instances.
- For information about setting up a Citrix ADC BLX cluster, see Citrix ADC BLX cluster.

- Appliances can belong to different networks.
- Be initially configured and connected to a common client-side and server-side network.
- For a cluster of virtual appliances (Citrix ADC VPX, or Citrix ADC BLX, or Citrix SDX ADC instance), that has large configurations, it is recommended to use 6 GB RAM for each node of the cluster.

Cluster overview

September 14, 2021

A Citrix ADC cluster is formed by grouping Citrix ADC appliances together. Based on the network location of the Citrix ADC appliances that you intend to add the cluster, you must be aware of the following cluster setups:

> **Note**
>
> Unless specified otherwise, cluster features and configurations are the same for L2 and L3 clusters.

- **L2 cluster:** In this cluster deployment, all cluster nodes belong to the same network.



- **L3 cluster (also referred to as 'cluster in INC mode'):** In this cluster deployment, cluster nodes can belong to different networks. The cluster nodes from a specific network must be grouped into node groups that include only nodes from that network. From the following figure, we see that nodes n1, n2, n3 are in the same network and are grouped into Nodegroup1.

  Similarly, the case for nodes n4 and n5, that are grouped in Nodegroup2. In the third network, there are two node groups. Nodegroup3 includes n6 and n7 and Nodegroup4 includes n8 and n9.

  > **Note**
  >
  > Supported from NetScaler 11.0 onwards.



Synchronization across cluster nodes

September 14, 2021

All configurations on a Citrix ADC cluster are performed on the cluster IP address, which is the management address of the cluster. The cluster node owns the cluster IP address that is referred to as the cluster configuration coordinator (CCO) as shown in the following figure:



The configurations that are available on the CCO are automatically propagated to the other cluster nodes and therefore all cluster nodes have the same configurations.

- Citrix ADC allows only a few configurations to be performed on individual cluster nodes through their NSIP address. In these cases, you must ensure configuration consistency manually across all nodes in the cluster. These configurations are not propagated across the other cluster nodes. For more information on operations supported on each cluster node, see Operations Supported on Individual Cluster Nodes.
- The following commands when ran on the cluster IP address are not propagated to other cluster nodes:
  - **shutdown.** Shuts down only the configuration coordinator.
  - **reboot.** Reboots only the configuration coordinator.
  - **rm cluster instance.** Removes the cluster instance from the node that you are running the command on.
- For a command to propagate to other cluster nodes:
  - The quorum must be configured on the cluster instance.
  - Most the cluster quorum with (n/2 + 1) of the cluster nodes must be active for the cluster to be operational.
  - A cluster can run with a minimum number of nodes when the majority rule (n/2 + 1) is relaxed.

When a node is added to a cluster, the configurations and the files (SSL certificates, licenses, DNS, and so on) that are available on the CCO are synchronized to the newly added cluster node. When an existing cluster node, that was intentionally disabled or that had failed, is once again added, the cluster compares the configurations available on the node with the configurations available on the CCO. If there is a mismatch in configurations, the node is synchronized by using one of the following:

- **Full synchronization.** If the difference between configurations exceeds 255 commands, all the configurations of the CCO are applied to the node that is rejoining the cluster. The node remains

operationally unavailable during the synchronization.

- **Incremental Synchronization.** If the difference between configurations is less than or equal to 255 commands, only the configurations that are not available are applied to the node that is rejoining the cluster. The operational state of the node remains unaffected.

> **Note**
>
> You can also manually synchronize the configurations and files. For more information, seeSynchronizing Cluster Configurations and Synchronizing Cluster Files.

Striped, partially striped, and spotted configurations

September 14, 2021

By virtue of command propagation, all nodes in a cluster have the same configurations. However, you might want some configurations to be available only on certain cluster nodes. While you cannot restrict the nodes on which the configurations are available, you can specify the nodes on which the configurations are active.

For example, you can:

- define a SNIP address to be active on only one node, or
- define a SNIP address to be active on all nodes, or
- define a VIP address to be active on only one node, or
- define a VIP address to be active on all nodes, or
- define a VIP address to be active only on two nodes of a 3-node cluster

Depending on the number of nodes the configurations are active on, cluster configurations are referred to as striped, partially striped, or spotted configurations.

Figure 1. Three-node cluster with striped, partially striped, and spotted configurations



The following table provides more details on the types of configurations:

Configuration Type	Active on	Applicable to	Configurations
Striped configuration	All the cluster nodes	All entries	No specific configuration required to make an entity striped. By default, all entities defined on a cluster IP address are striped on all the cluster nodes.
Partially striped configuration	A subset of cluster nodes	Refer to Cluster Node groups.	Bind the entities that you want to be partially striped, to a node group. The configuration is active only on the cluster nodes that belong to the node group.

Configuration Type	Active on	Applicable to	Configurations
Spotted configuration	Single cluster node	SNIP address, SNMP Engine ID, host name of cluster nodes, Entities that can be bound to a node group	A spotted configuration can be defined using one of two approaches. **SNIP address** When creating the SNIP address, specify the node on which you want the SNIP address to be active, as the owner node. **Example**, `add ns ip 10.102.29.106 255.255.255.0 -type SNIP -ownerNode 2` (assuming node NS2 ID is 2). **Note**: You cannot change the ownership of a spotted SNIP address at run time. To change the ownership, you must first delete the SNIP address and add it again by specifying the new owner. **Entities that can be bound to a node group.** By binding the entity to a single-member node group.

> **Note**
>
> - When you disable USIP, Citrix recommends you to use spotted SNIP addresses. You can use striped SNIP addresses only if there is a shortage of IP addresses. The use of striped IP addresses can result in ARP flux issues if no spotted IP addresses are present in the same subnet for ARP resolution.
> - When you enable USIP, Citrix recommends you to use striped SNIP addresses as a gateway for server initiated traffic.

ARP owner support for striped IP

In a cluster setup, you can configure a specific node to respond to the ARP request for a striped IP. The configured node responds to the ARP traffic.

A new parameter “arpOwner” is introduced in the “add, set, and unset IP” commands.

To enable the ARP owner on a node by using the CLI.

At the command prompt, type:

```
add ns ip <ip_address> -arpOwner <node_id>
```

> **Note**
>
> The ARP owner parameter is supported only in the L2 cluster.

Neighbor discovery owner support for striped IPv6 address

In a cluster setup, you can configure a specific node as neighbor discovery (ND) owner for the striped IPv6 address to determine the link-layer address. A client sends a Neighbor Solicitation (NS) message to all the nodes in the cluster setup. The ND owner responds with a Neighbor Advertisement (NA) message with the link-layer address for the striped IPv6 address, and serves traffic.

To enable ND owner on a node by using the CLI

At the command prompt, type:

```
add ns ip6 <IPv6Address> -ndOwner <node id>

set  ns ip6 <IPv6Address> -ndOwner <node id>
<!--NeedCopy-->
```

Example:


```
add ns ip6 2001::21/64 -ndOwner 1

set ns ip6 2001::21/64 -ndOwner 1
<!--NeedCopy-->
```

To enable ND owner on a node by using the GUI

1. Navigate to **System > Network > IPs**.
2. In the **IPs** page, go to the **IPv6s** tab and click **Add**.
3. In the **Create IPv6** page, select one of the node IDs listed in **NdOwner in Cluster** drop-down menu.

> **Note**
>
> The ND owner parameter is supported only in the L2 cluster.

Communication in a cluster setup

September 14, 2021

The interfaces of Citrix ADC appliances that are added to a cluster are prefixed with a node ID. It helps identify the cluster node to which the interface belongs. Therefore, the interface identifier c/u, where c is the controller number and u is the unit number, now becomes n/c/u, where n is the node ID. For example, in the following figure, interface 1/2 of node n1 is represented as 0/1/2, interface 1/1 of node n2 is represented as 1/1/1, and interface 1/4 of node n3 is represented as 2/1/4.

Figure 1. Interface naming convention in a cluster



- **Server communication**-
  The cluster communicates with the server through the physical connections between the cluster node and the server-side connecting device. The logical grouping of these physical connections is called the server data plane.

- **Client communication**- The cluster communicates with the client through the physical connections between the cluster node and the client-side connecting device. The logical grouping of these physical connections is called the client data plane.
- **Inter-node communication**- The cluster nodes can also communicate with each other. The manner in which they communicate depends on whether the node exists on the same network or across networks.
  - Cluster nodes within the same network communicate with each other by using the cluster backplane. The backplane is a set of interfaces in which one interface of each node is connected to a common switch, which is called the cluster backplane switch. The different types of traffic that goes through the backplane, which is used by internode communication are:
    * Node to Node Messaging (NNM)
    * Steered traffic
    * Configuration propagation and synchronization
  - Each node of the cluster uses a special MAC cluster backplane switch address to communicate with other nodes through the backplane. The cluster special MAC is of the form: `0x02 0x00 0x6F <cluster_id> <node_id> <reserved>)`, where `cluster_id` is the cluster instance ID, `node_id` is the node number of the Citrix ADC appliance that is added to a cluster.

The following figures shows the communication interfaces in L2 clusters and L3 clusters.

Figure 2. Cluster communication interfaces - L2 cluster



Figure 3. Cluster communication interfaces - L3 cluster



Traffic distribution in a cluster setup

September 14, 2021

In a cluster setup, external networks view the collection of Citrix ADC appliances as a single entity. So, the cluster must select a single node that must receive the traffic. The cluster does this selection by using the Equal Cost Multiple Path (ECMP) or cluster link aggregation traffic distribution mechanism. The selected node is called the flow receiver.

> **Note**
>
> For an L3 cluster (nodes across different networks), only the ECMP traffic distribution can be used.

The flow receiver gets the traffic and then, using internal cluster logic determines the node that must process the traffic. This node is called the flow processor. The flow receiver steers the traffic to the flow processor over the backplane if the flow receiver and the flow processor are on the same network. The traffic is steered through the tunnel if the flow receiver and the flow processor are on different networks.

Note

- The flow receiver and flow processor must be nodes capable of serving traffic.
- From NetScaler 11 onwards, you can disable steering on the cluster backplane. For more information, see Disabling Steering on the Cluster Backplane.

Figure 1. Traffic distribution in a cluster



The preceding figure shows a client request flowing through the cluster. The client sends a request to a virtual IP (VIP) address. A traffic distribution mechanism configured on the client data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the node that must process the traffic, and steers the request to that node (unless the flow receiver selects itself as the flow processor).

The flow processor establishes a connection with the server. The server processes the request and

sends the response to the subnet IP (SNIP) address that sent the request to the server.

- If the SNIP address is a striped or partially striped IP address, the traffic distribution mechanism configured on the server data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the flow processor, and steers the request to the flow processor through the cluster backplane.
- If the SNIP address is a spotted IP address, the node that owns the SNIP address receives the response from the server.

In an asymmetric cluster topology (all cluster nodes are not connected to the external switch), you must use linksets either exclusively or combined with ECMP or cluster link aggregation. For more information, see Using Linksets.

Cluster node groups

September 14, 2021

> **Note**
>
> Node groups are supported from NetScaler 10.1 onwards.

As the name indicates, a cluster node group is a group of cluster nodes.

Figure 1. Citrix ADC cluster with node groups



The preceding figure shows a cluster which has node groups NG1 and NG2 that include 3 cluster nodes each. The cluster also has 3 nodes that are not part of any node group.

A node group can be configured for the following:

- To define spotted and partially striped configurations. For more information, see Node groups for Spotted and Partially Striped Configurations.
- To configure redundancy of node groups. For more information, see Configuring Redundancy for Node groups.
  Note: Supported from NetScaler 10.5 Build 52.1115.e onwards.

- To define an L3 cluster (also called a cluster in INC mode). In an L3 cluster, cluster nodes can be from different networks. You must group nodes that belong to a network in a single node group. For example, if n1, n2, n3 are in network1 and n4, n5, n6 are in network2, then NG1 must include nodes of network1 and NG2 must include nodes of network2. For setting up an L3 cluster, see Creating a Citrix ADC cluster.

> **Note**
>
> - Supported from NetScaler 11 onwards.
> - The preceding functions of a node group are mutually exclusive. It means that a node group can provide only one of the preceding mentioned functionalities.

Cluster and node states

September 14, 2021

For a cluster to be functional, most of the nodes (n/2 + 1) must be operationally active (operational state is ACTIVE).

> **Important**
>
> From NetScaler release 10.5, you can configure the cluster to be functional even when the majority criteria is not satisfied. This configuration must be performed when creating a cluster.

For more information on states of a cluster node, refer to States of a cluster node.

Routing in a cluster

September 14, 2021

Routing in a cluster works in much the same way as routing in a standalone system. A few points to note:

- All routing configurations must be performed from the cluster IP address and the configurations are propagated to the other cluster nodes.
- Routes are limited to the maximum number of ECMP routes supported by the upstream router.
- Node-specific routing configurations must be performed by using the owner-node argument as follows:

```
1   router ospf
2     owner-node 0
```



```
3        ospf router-id 97.131.0.1
4      exit-owner-node
5    !
6  <!--NeedCopy-->
```

The following command displays the consolidated cluster configuration for all nodes in VTYSH.

```
show cluster-config
```

The following command displays the cluster status on each node.

```
show cluser node
```

IPv4 routing in L2 cluster

The following section contains sample configurations that help you to configure IPv4 OSPF and BGP routing in the L2 cluster.

Adding spotted SNIP address and enabling dynamic routing

In the following configuration, OSPF, and BGP routing are enabled. Also, spotted SNIP addresses are added and dynamic routing is enabled on these SNIP addresses.

```
1 en ns fea ospf bgp
2 add vlan 10
3 add ns ip 10.10.10.1 255.255.255.0 -dynamicrouting enabled -ownernode 1
4 add ns ip 10.10.10.2 255.255.255.0 -dynamicrouting enabled -ownernode 2
5 add ns ip 10.10.10.3 255.255.255.0 -dynamicrouting enabled -ownernode 3
6 bind vlan 10 -ipaddress 10.10.10.1 255.255.255.0
7 <!--NeedCopy-->
```

VTYSH IPv4 OSPF configuration

For configuring IPv4 OSPF in the L2 cluster, you must:

- Set the priority to zero.
- Configure the Router-id as a spotted configuration.

> **Note**
>
> The OSPF configuration guidelines for the L2 cluster are applicable for OSPFv3 also.

In the following sample configuration IPv4 OSPF is configured.

```
1       interface vlan10
2        IP OSPF PRIORITY 0
```



```
        !
       router ospf
           owner-node 1
             ospf router-id 97.131.0.1
           exit-owner-node
           owner-node 2
             ospf router-id 97.131.0.2
           exit-owner-node
           owner-node 3
             ospf router-id 97.131.0.3
           exit-owner-node
       network 10.10.10.0/24 area 0
       redistribute kernel
        !
<!--NeedCopy-->
```

VTYSH IPv4 BGP configuration

In the following VTYSH sample configuration, IPv4 BGP is configured.

```
     router bgp 100
       neighbor 10.10.10.10 remote-as 200
      owner-node 1
       neighbor 10.10.10.10 update-source 10.10.10.1
      exit-owner-node
      owner-node 2
       neighbor 10.10.10.10 update-source 10.10.10.2
      exit-owner-node
      owner-node 3
       neighbor 10.10.10.10 update-source 10.10.10.3
      exit-owner-node
     redistribute kernel
     !
<!--NeedCopy-->
```

> **Note**
>
> The update-source command is used for each neighbor with the owner-node argument in the following configuration to connect with proper source IP.

IPv6 routing in L2 cluster

The following section contains sample configurations that help you to configure IPv6 OSPF and BGP routing in the L2 cluster.

Enable IPv6 routing

Before configuring IPv6 routing in a L2 cluster, you must enable the IPv6 feature.

To enable IPv6 routing by using the CLI,

At the command prompt, type:

- `enable ns fea ipv6pt`

Adding spotted SNIP6 address and enabling dynamic routing

In the following configuration, OSPF, and BGP routing are enabled. Also, spotted SNIP6 addresses are added and dynamic routing is enabled on these SNIP6 addresses.

```
add ns ip6 3ffa::1/64 -dynamicrouting enabled -ownernode 1
add ns ip6 3ffa::2/64 -dynamicrouting enabled -ownernode 2
add ns ip6 3ffa::3/64 -dynamicrouting enabled -ownernode 3
add vlan 10
bind vlan 10 -ipaddress 3ffa::1/64
<!--NeedCopy-->
```

VTYSH IPv6 BGP configuration

In the following VTYSH sample configuration, IPv6 BGP is configured.

```
  router bgp 100
   neighbor 3ffa::10 remote-as 200
     owner-node 1
      neighbor 3ffa::10 update-source 3ffa::1
     exit-owner-node
     owner-node-2
      neighbor 3ffa::10 update-source 3ffa::2
     exit-owner-node
     owner-node-3
      neighbor 3ffa::10 update-source 3ffa::3
     exit-owner-node
   no neighbor 3ffa::10 activate
   address-family ipv6
    redistribute kernel
    neighbor 3ffa::10 activate
   exit-address-family
  !
<!--NeedCopy-->
```


Install IPv6 learned routes

The Citrix ADC cluster can use routes learned by various routing protocols after you install the routes in the Citrix ADC cluster routing table.

To install IPv6 learned routes to the internal routing table by using the CLI:

At the command prompt, type:

- `ns route-install ipv6 bgp`
- `ns route-install ipv6 ospf`
- `ns route-install default`

> **Note**
>
> - If you have to exchange IPv4 routes on a IPv6 neighbor, you must remove the `no neighbor 3ffa::10 active` VTYSH command from the earlier configuration.
> - The `update-source` VTYSH command must be used for each owner node to specify the right IPv6 source IP while connecting to the BGP peer as given in the BGP IPv4 configuration.

Routing in a L3 cluster

The routing in a L3 cluster works only when the following configurations are done on the Citrix ADC appliance.

- Enable the dynamic routing for a VLAN.

```
set vlan <id> -dynamicrouting enabled
<!--NeedCopy-->
```

- To reach all cluster nodes, the VIP, CLIP, and Citrix ADC IP (NSIP) must be advertised by routing protocols along with the `set vlan` command.

Deployment scenario for BGP in L3 cluster

Consider an example where all the cluster nodes are grouped in the AS 100 network, and the upstream routers are in a different AS 200.

The following figure depicts the AS 100 and AS 200 deployment in a cluster setup.



In this deployment, CLIP advertises CCO to upstream routers. Some cluster nodes drop the advertised traffic, as an AS loop is detected.

To overcome the issue, configure the following command in VTYSH BGP router mode for each neighbor.

At the VTYSH command prompt, type:

```
neighbor <peer_ip> allowas-in 1
```

As a best practice, Citrix recommends you to configure any one of the following:

- Configure route-maps to learn only desired networks such as; default route, Citrix ADC IP (NSIP), and NSIP subnets on cluster nodes.
- Configure upstream routes to advertise only desired networks such as; CLIP and Citrix ADC IP (NSIP) in cluster.

IP addressing for a cluster

September 14, 2021

In addition to the standard types of Citrix ADC-owned IP addresses—Citrix ADC NSIP, Virtual IP (VIP), and Subnet IP (SNIP)—a clustered Citrix ADC appliance can have a cluster management IP (CLIP) address. It can also have striped and spotted IP addresses.

- **CLIP address.** An IP address owned by the cluster coordinator node (CCO). The CLIP address can float between different nodes in a cluster setup. If the CLIP is moved to a different node of the cluster, that node becomes the CCO. The CCO is the Citrix ADC appliance that is responsible for management tasks in the cluster. A network administrator uses the CLIP address to connect to the cluster to perform configuration and management tasks, such as accessing the unified GUI, reporting, tracing packet flow, and collecting logs. You can add multiple CLIP addresses in a cluster on the same or different networks. Only configurations performed on the CCO through the cluster IP address are propagated to other nodes in the cluster.
- **Striped IP address**. A logical IP address available on all nodes of the cluster, it can be either a VIP or SNIP address.
- **Spotted IP address**. A logical IP (preferably SNIP address) is available only on one node. A spotted IP address has visibility on only that node. To minimize traffic-steering overhead, Citrix recommends that you use a spotted SNIP address for back-end communication with the server.

The following table provides the details of the configurations.

IP Address	NSIP	VIP	SNIP
Spotted	Yes	Yes	Yes
Striped	No	Yes	Yes

For example, in a four-node cluster group, you must configure each node with a spotted SNIP address. For more information on how to configure a spotted IP configuration, see Striped, Partially Striped, and Spotted Configurations.

You can define a SNIP address to be active on only one node, or active on all nodes. If the virtual IP address and subnet IP address are available only on a specific node, it is of spotted configuration. The configuration is defined as striped if the subnet IP address and virtual server IP address are available on all nodes. Spotted SNIP addresses help in reducing the steering and backplane traffic.

Best practices for VLAN bindings and route configuration while joining a node to the cluster

VLAN IP bindings

When you bind a VLAN with the spotted IP address, the Citrix ADC cluster must be configured with the spotted IP addresses in the same subnet on all the nodes. For example, in a two-node cluster with Node 0 and Node 1, you can have the following configuration:

```
1 add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
    dynamicRouting ENABLED -ownerNode 1
```



```
2 add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 0
3 add vlan 100
4 bind vlan 100 -IPAddress 192.254.101.101 255.255.255.0
5 <!--NeedCopy-->
```

Routing configuration

When routing configuration is required with the spotted IP address as the default gateway, then the ADC cluster must be configured with the spotted IP addresses in the same subnet on all the nodes. For example, in a two-node cluster with Node 0 and Node 1, you can have the following configuration:

```
1 add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 1
2 add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 0
3
4 add route 192.254.102.0 255.255.255.0 192.254.101.103
5 <!--NeedCopy-->
```

> **Note**
>
> In an L3 cluster setup, only spotted SNIP configuration is supported.

Configuring layer 3 clustering

September 14, 2021

Understanding the L3 cluster

The demand to expand the high availability deployment and increase the scalability of the client traffic across different networks guided to establish the L3 cluster. The L3 cluster lets you group Citrix ADC appliances across individual subnets (L2 cluster).

L3 cluster is also referred to as “cluster in Independent Network Configuration (INC) mode”. In L3 cluster deployment, the cluster nodes in the same network are grouped to form a Node group. L3 cluster uses GRE tunneling to steer the packets across networks. The heartbeat messages across the L3 clusters are routed.

This document includes the following details:

- Architecture

- Example

Architecture

The L3 cluster architecture comprises the following components:

- **Nodegroup**. The cluster nodes from each network (n1, n2) and (n3, n4), as depicted in the following figure, are grouped to form a Node group. These Node groups are terminated to the layer 3 switch on either side of the network.
  - The cluster communicates with the client through the physical connections between the cluster node and the client-side connecting device. The logical grouping of these physical connections is called the client data plane.
  - The cluster communicates with the server through the physical connections between the cluster node and the server side connecting device. The logical grouping of these physical connections is called the server data plane.
- **Backplane Switch.** Cluster nodes within the same network communicate with each other by using the cluster backplane. The backplane is a set of interfaces in which one interface of each node is connected to a common switch, which is called the cluster backplane switch.
- **GRE Tunnel.** The packets between nodes in a L3 cluster are exchanged over an unencrypted GRE tunnel that uses the NSIP addresses of the source and destination nodes for routing. The steering mechanism changes for nodes belonging to the different network. The packets are steered through a GRE tunnel to the node on the other subnet, instead of rewriting the MAC.



Example

Consider an example of an L3 cluster deployment consisting of the following:

- Three Citrix ADC appliances (n1, n2, and n3) nodes are grouped into Nodegroup1.
- Similarly, the nodes n4 and n5 are grouped in Nodegroup2. In the third network, there are two node groups. Nodegroup3 includes n6 and n7 and Nodegroup4 includes n8 and n9.
- The Citrix ADC appliances that belong to the same network are combined to form a node group.



Points to consider before configuring the L3 cluster

Consider the following points before configuring the L3 cluster on a Citrix ADC appliance:

- The backplane is not mandatory while configuring L3 subnets. If the backplane is not specified, the node does not go to the backplane fail state.

  > **Note**
  >
  > If you have more than one node in the same L2 network, it is mandatory to define the backplane interface. If the backplane interface is not mentioned, the nodes go to the backplane fail state.

- L2 features and striped SNIPs are not supported in the L3 cluster.
- The external traffic distribution in the L3 cluster supports only Equal Cost Multiple Path (ECMP).
- The ICMP errors and fragmentation are not processed when steering is disabled in an L3 cluster deployment:
- The networking entities (`route`, `route6`, `pbr`, and `pbr6`) must be bound to the configuration node group.

- VLAN, RNAT, and IP tunnel cannot be bound to a configuration node group.
- Configuration node group must always have property STRICT “YES.
- The cluster nodes must not be added to a config node group via “add cluster node” command.
- The `add cluster instance -INC enabled` command clears the networking entities (route, route6, PBR, pb6, RNAT, IP tunnel, ip6tunnel).
- The `clear config extended+` command does not clear the entities (route, route6, PBR, pb6, RNAT, IP tunnel, ip6tunnel) in an L3 cluster.

Configuring the L3 cluster

In an L3 cluster configuration, the cluster command has different attributes to configure that is based on nodes, and node groups. The L3 cluster configuration also includes an IPv6 profile apart from IPv4 profiles.

Configuring an L3 cluster on a Citrix ADC appliance consists of the following tasks:

- Create a cluster instance
- Create a node group in L3 cluster
- Add a Citrix ADC appliance to the cluster and group with node group
- Add cluster IP address to the node
- Enable the cluster instance
- Save the configuration
- Add a node to an existing node group
- Create a node group in L3 cluster
- Group new nodes to the newly created node group
- Join the node to the cluster

Configuring the following by using the CLI

- **To create a cluster instance**

  ```
  add cluster instance <clid> -inc (<ENABLED|DISABLED>)[-processLocal <
  ENABLED | DISABLED]
  ```

- **To create a nodegroup in L3 cluster**

  ```
  add cluster nodegroup <name>
  ```

- **To add a Citrix ADC appliance to the cluster and to associate with nodegroup**

  ```
  add cluster node <nodeid> <nodeip> -backplane <interface_name> node
  group <ng>
  ```


- **To add the cluster IP address on this node**

  ```
  add ns ip <IPAddress> <netmask> -type clip
  ```

- **Enable the cluster instance**

  ```
  enable cluster instance <clId>
  ```

- **Save the configuration**

  ```
  save ns config
  ```

- **Warm reboot the appliance**

  ```
  reboot -warm
  ```

- **To add a new node to an existing nodegroup**

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng>
  ```

- **To create a new nodegroup in L3 cluster**

  ```
  add cluster nodegroup <ng>
  ```

- **To group new nodes to the newly created nodegroup**

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng>
  ```

- **To join the node to the cluster**

```
1     join cluster  - clip <ip_addr> -password <password>
2
3     add cluster instance 1  - inc ENABLED  - processLocal  ENABLED
4
5        Done
6 <!--NeedCopy-->
```

> **Note**
>
> The “inc” parameter must be ENABLED for an L3 cluster.

```
1     add cluster nodegroup ng1
2
3        Done
4
5     > add cluster node 0 1.1.1.1  - state ACTIVE -backplane 0/1/1  -
          nodegroup ng1
6
7        Done
8
9     > add ns ip 1.1.1.100 255.255.255.255  - type clip
10
```



```
11         Done
12
13      > enable cluster instance 1
14
15         Done
16
17      > save ns config
18
19         Done
20
21      > add cluster node 1 1.1.1.2  -state ACTIVE  -nodegroup ng1
22
23         Done
24
25      > add cluster nodegroup ng2
26
27         Done
28
29      > add cluster node 4 2.2.2.1  -state ACTIVE  -nodegroup ng2
30
31         Done
32
33      > add cluster node 5 2.2.2.2  -state ACTIVE  -nodegroup ng2
34
35         Done
36
37      > join cluster -clip 1.1.1.100 -password nsroot
38  <!--NeedCopy-->
```

Advertising cluster IP address of a L3 cluster

Configure the cluster IP address to be advertised to the upstream router to make the cluster configuration accessible from any subnet. The cluster IP address is advertised as a kernel route by the dynamic routing protocols configured on a node.

Advertising the cluster IP address consists of the following tasks:

- **Enable the host route option of the cluster IP address**. The host route option pushes the cluster IP address to a ZebOS routing table for kernel route redistribution through dynamic routing protocols.
- **Configuring a dynamic routing protocol on a node**. A dynamic routing protocol advertises the cluster IP address to the upstream router. For more information on configuring a dynamic routing protocol, see Configuring Dynamic Routes.

To enable the host route option of the cluster IP address by using the CLI

At the command prompt, type:

```
-  add nsip <IPAddress> <netmask> -hostRoute ENABLED

-  show nsip \<IPAddress\>

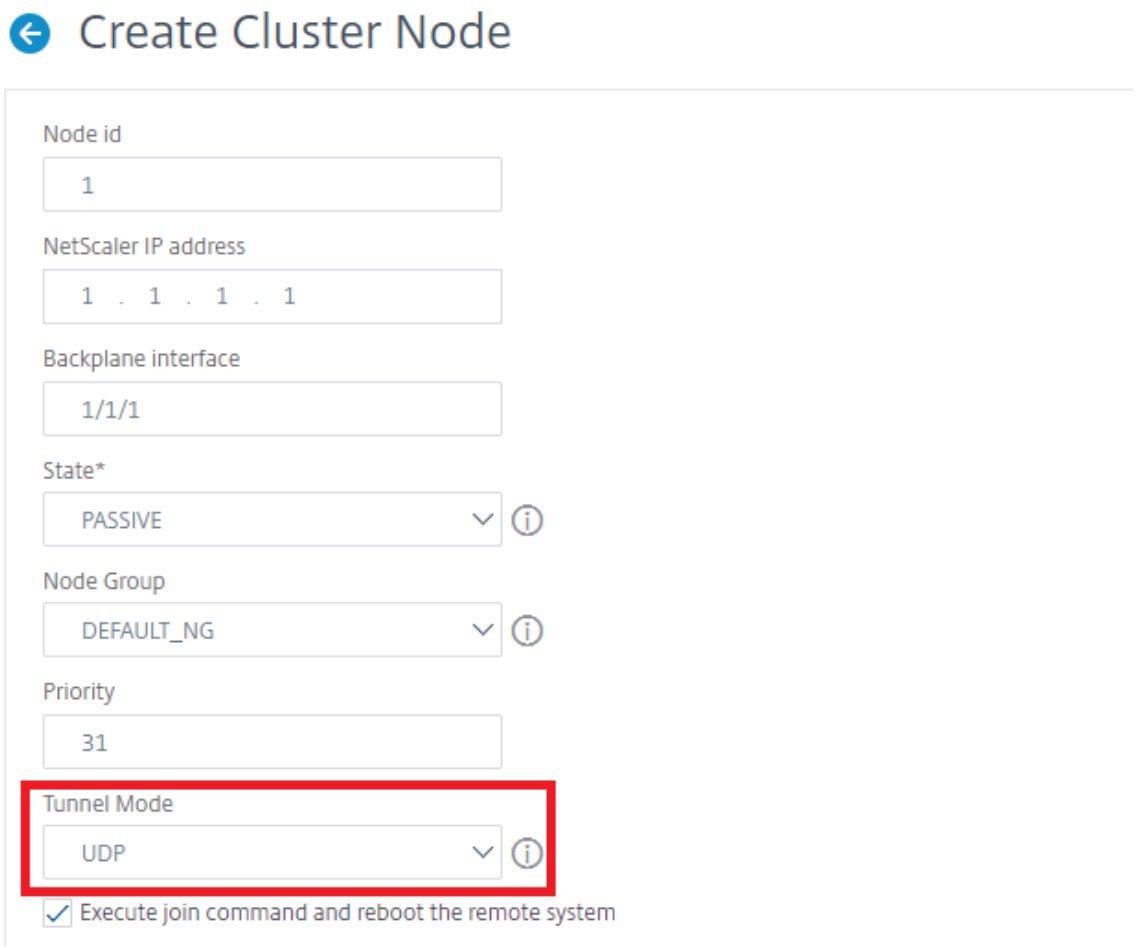
    > add ns ip 10.102.29.60 255.255.255.255 -hostRoute ENABLED

       Done
<!--NeedCopy-->
```

Spotted, partially striped configurations on L3 cluster

The spotted and partially striped configurations on the L3 cluster slightly differ from the L2 cluster. The configuration might differ from node to node as the nodes reside on different subnets. The network configurations can be node specific in the L3 cluster, hence you have to configure the spotted or partially striped configurations based on the below-mentioned parameters.

To configure spotted, partially striped configurations on a Citrix ADC appliance over the L3 cluster, perform the following tasks:

- Add a cluster owner group to an IPv4 static routing table
- Add a cluster owner group to an IPv6 static routing table
- Add a cluster owner group to an IPv4 policy based routing (PBR)
- Add a cluster owner group to an IPv6 PBR
- Add a VLAN
- Bind a VLAN to a specific owner group of cluster node group

Configuring the following by using the CLI

- **To add a cluster ownergroup to an IPv4 static route table of the Citrix ADC appliance**

  ```
  add route <network> <netmask> <gateway> -owner group <ng>
  ```

- **To add a cluster ownergroup to an IPv6 static route table of the Citrix ADC appliance**

  ```
  add route6 <network> -owner group <ng>
  ```

- **To add a cluster ownergroup to an IPv4 PBR**

  ```
  add pbr <name> <action> -owner group <ng>
  ```

- **To add a cluster ownergroup to an IPv6 PBR**

  ```
  add pbr6 <name> <action> -owner group <ng>
  ```


- **To add a VLAN**

  ```
  add vlan <id>
  ```

- **To bind a VLAN to a specific ownergroup of cluster nodegroup**

  ```
  bind vlan <id> -ifnum - [IPAddress <ip_addr | ipv6_addr> [-owner group
  <ng>]
  ```

  The following commands are sample examples of spotted and partially striped configurations which can be configured by using the CLI.

```
1     > add route 10.102.29.0 255.255.255.0 10.102.29.2  - ownergroup ng2
2
3         Done
4
5     > add route6 fe80::9404:60ff:fedd:a464/64  - ownergroup ng1
6
7         Done
8
9     > add pbr pbr1 allow  - ownergroup ng1
10
11        Done
12
13    > add pbr6 pbr2 allow  - ownergroup ng2
14
15        Done
16
17    > add vlan 2
18
19        Done
20
21    > bind vlan 2  - ifnum 1/2  - [IPAddress 10.102.29.80 | fe80::9404:60
         ff:fedd:a464/64-ownergroup ng1
22
23        Done
24 <!--NeedCopy-->
```

Configure node group

In an L3 cluster, to replicate the same set of configurations on more than one node group, the following commands are used:

Configuring the following by using the CLU

- **To add an IPv4 static route to the routing table of the Citrix ADC appliance**

```
add route <network> <netmask> <gateway> -ownerGroup <ng>
```

Sample Configuration:

```
add route 0 0 10.102.53.1 -ownerGroup ng1

add route 0 0 10.102.53.1 -ownerGroup ng2
<!--NeedCopy-->
```

You define a new node group 'all' to support the preceding configuration, and have to configure the following commands:

Configuring the following by using the CLI

- **To add a new nodegroup to cluster with strict parameter**

```
add cluster node group <name> -strict <YES | NO>
```

- **To bind a cluster node or an entity to the given nodegroup**

```
bind cluster nodegroup <name> -node <nodeid>
```

- **To add IPv4 static route to all ownergroup**

```
add route <network> <netmask> <gateway> -ownerGroup <ng>
```

Sample configuration:

```
add cluster nodegroup all -strict YES

bind cluster nodegroup all -node 1

bind cluster nodegroup all -node 2

add route 0 0 10.102.53.1 -ownerGroup all
<!--NeedCopy-->
```

Traffic distribution in a L3 cluster

In a cluster setup, external networks view the collection of Citrix ADC appliances as a single entity. So, the cluster must select a single node that must receive the traffic. In the L3 cluster this selection is done using the ECMP. The selected node is called the flow receiver.

> **Note**
>
> For an L3 cluster (nodes across different networks), only the ECMP traffic distribution can be used.

The flow receiver gets the traffic and then, using internal cluster logic determines the node that must process the traffic. This node is called the flow processor. The flow receiver steers the traffic to the flow processor over the backplane if the flow receiver and the flow processor are on the same network. The traffic is steered through the tunnel if the flow receiver and the flow processor are on different networks.

> **Note**
>
> - The flow receiver and flow processor must be nodes capable of serving traffic.
> - From NetScaler 11 onwards, you can disable steering on the cluster backplane. For more information, see Disabling steering on the cluster backplane.

The preceding figure shows a client request flowing through the cluster. The client sends a request to a virtual IP (VIP) address. A traffic distribution mechanism configured on the client data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the node that must process the traffic, and steers the request to that node (unless the flow receiver selects itself as the flow processor). If the flow processor and flow receiver are in the same node group, the packet is steered over the backplane. And if the flow processor and flow receiver are in different node groups, then the packet is steered through the tunnel over the routed path.

The flow processor establishes a connection with the server. The server processes the request and sends the response to the subnet IP (SNIP) address that sent the request to the server. Since in the L3 cluster the SNIP is always a spotted SNIP, the node that owns the SNIP address receives the response from the server.

Setting up a Citrix ADC cluster

September 14, 2021

Citrix ADC appliances that you want to add to the cluster must satisfy the criteria specified in Prerequisites for Cluster Nodes. Before actually setting up a cluster, you must be aware of cluster basics. For information, see Cluster Overview.

Forming a cluster requires you to set up inter-node communication, create the cluster (by adding the first Citrix ADC appliance), and then add the other cluster nodes. Each of these steps is explained with relevant details in subsequent topics.

> **Note**
>
> While there are some differences in setting up an L2 and L3 cluster, there are many similarities too. The subsequent topics explain the setup for both cluster types while highlighting the configurations that are specific to L3 clusters.

Setting up inter-node communication

September 14, 2021

The nodes in a cluster setup communicate with one another using the following inter-node communication mechanisms:

- Nodes that are within the network (same subnet) communicate with each other through the cluster backplane. The backplane must be explicitly set up. The following are the detailed steps.
- Across networks, steering of packets is done through a GRE tunnel and other node-to-node communication is routed across nodes as required.

> **Important**
>
> - From Release 11.0 all builds, a cluster can include nodes from different networks.
> - From Release 13.0 build 58.3, GRE steering is supported on Fortville NICs in an L3 cluster.

To set up the cluster backplane, do the following for every node

1. Identify the network interface that you want to use for the backplane.
2. Connect an Ethernet or optical cable from the selected network interface to the cluster backplane switch.

For example, to use interface 1/2 as the backplane interface for node 4, connect a cable from the 1/2 interface of node 4 to the backplane switch.

Important points to note when setting up the cluster backplane

- Do not use the appliance's management interface (0/x) as the backplane interface. In a cluster, the interface 0/1/x is read as:

  0 -> node ID 0
  1/x -> Citrix ADC interface

- Do not use the backplane interfaces for the client or server data planes.
- Citrix recommends using the link aggregate (LA) channel for the cluster backplane.

- In a two-node cluster, where the backplane is connected back-to-back, the cluster is operationally DOWN under any of the following conditions:
  - One of the nodes is rebooted.
  - Backplane interface of one of the nodes is disabled.

  Therefore, Citrix recommends that you dedicate a separate switch for the backplane, so that the other cluster node and traffic are not impacted. You cannot scale out the cluster with a back-to-back link. You might encounter a downtime in the production environment when you scale out the cluster nodes.
- Backplane interfaces of all nodes of a cluster must be connected to the same switch and bound to the same L2 VLAN.
- If you have multiple clusters with the same cluster instance ID, make sure that the backplane interfaces of each cluster are bound to a different VLAN.
- The backplane interface is always monitored, regardless of the HA monitoring settings of that interface.
- The state of MAC spoofing on the different virtualization platforms can affect the steering mechanism on the cluster backplane. Therefore, make sure the appropriate state is configured:
  - XenServer - Disable MAC spoofing
  - Hyper-V - Enable MAC spoofing
  - VMware ESX - Enable MAC spoofing (also make sure “Forged Transmits” is enabled)
- The MTU for the cluster backplane is automatically updated. However, if jumbo frames are configured on the cluster, the MTU of the cluster backplane must be explicitly configured. The value must be set to 78 + X, where X is the maximum MTU of the client and server data planes. For example, if the MTU of a server data plane is 7500 and of the client data plane is 8922. The MTU of a cluster backplane must be set to 78 + 8922 = 9000. To set this MTU, use the following command:

  ```
  > set interface <backplane_interface> -mtu <value>
  ```
- The MTU for the interfaces of the backplane switch must be specified to be greater than or equal to 1,578 bytes. It is applicable if the cluster has features like MBF, L2 policies, ACLs, routing in CLAG deployments, and vPath.

UDP based tunnel support for L2 and L3 cluster

Starting from Citrix ADC release 13.0 build 36.x, Citrix ADC L2 and L3 cluster can steer the traffic using UDP based tunneling. It is defined for the inter-node communications of two nodes in a cluster. By using the “tunnel mode” parameter, you can set GRE or UDP tunnel mode from the add and set cluster node command.

In an L3 cluster deployment, packets between Citrix ADC nodes are exchanged over an unencrypted GRE tunnel that uses the NSIP addresses of the source and destination nodes for routing. When this exchange occurs over the internet, in the absence of an IPsec tunnel, the NSIPs is exposed on the internet, and might result in security issues.

> **Important**
>
> Citrix recommends customers to establish their own IPsec solution when using a L3 cluster.

The following table helps you to categorize the tunnel support based on different deployments.

Steering Types	AWS	Microsoft Azure	On -premises
MAC	Not supported	Not supported	Supported
GRE tunnel	Supported	Not supported	Supported
UDP tunnel	Supported	Supported	Supported

> **Important**
>
> In a L3 cluster, the tunnel mode is set to GRE by default.

Configuring UDP based tunnel

You can add a cluster node by setting the parameters of node ID and mention the state. Configure the backplane by providing the interface name, and select the tunnel mode of your choice (GRE or UDP).

CLI procedures

To enable the UDP tunnel mode by using the CLI.

At the command prompt, type:

- `add cluster node <nodeId>@ [-state <state>] [-backplane <interface_name>] [-tunnelmode <tunnelmode>]`
- `set cluster node <nodeId>@ [-state <state>] [-tunnelmode <tunnelmode>]`

> **Note**
>
> Possible values for tunnel mode are NONE, GRE, UDP.

Example

- `add cluster node 1 -state ACTIVE -backplane 1/1/1 -tunnelmode UDP`
- `set cluster node 1 -state ACTIVE -tunnelmode UDP`

GUI procedures

To enable the UDP tunnel mode by using the GUI.

1. Navigate to **System > Cluster > Nodes**.
2. In the **Cluster Nodes** page, click **Add**.
3. In the **Create Cluster Node**, set the **Tunnel Mode** parameter to UDP and click **Create**.



4. Click **Close**.

Creating a Citrix ADC cluster

September 14, 2021

To create a cluster, start by taking one of the Citrix ADC appliances that you want to add to the cluster. On this node, you must create the cluster instance and define the cluster IP address. This node is the first cluster node and is called the cluster configuration coordinator (CCO). All configurations that are

performed on the cluster IP address are stored on this node and then propagated to the other cluster nodes.

The responsibility of CCO in a cluster is not fixed to a specific node. It can change over time depending on the following factors:

- The priority of the node. The node with the highest priority (lowest priority number) is made the CCO. Therefore, if a node with a priority number lower than the existing CCO is added, the new node takes over as the CCO.

  > **Note**
  >
  > Node priority can be configured from NetScaler 10.1 onwards.

- If the current CCO goes down, the node with the next lowest priority number takes over as the CCO. If the priority is not set or if there are multiple nodes with the lowest priority number, the CCO is selected from one of the available nodes.

> **Note**
>
> The configurations of the appliance (including SNIP addresses and VLANs) are cleared by implicitly running the `clear ns config extended` command. However, the default VLAN and NSVLAN are not cleared from the appliance. Therefore, if you want the NSVLAN on the cluster, make sure it is created before the appliance is added to the cluster. For an L3 cluster (cluster nodes on different networks), networking configurations are not cleared from the appliance.
>
> **Important**
>
> HA Monitor (HAMON) on a cluster setup is used to monitor the health of an interface on each node. The HAMON parameter must be enabled on each node to monitor the state of the interface. If the operational state of the HAMON enabled interface goes down due to any reason, the respective cluster node is marked as unhealthy (NOT UP) and that node cannot serve traffic.

To create a cluster by using the command line interface

1. Log on to an appliance (for example, appliance with NSIP address 10.102.29.60) that you want to add to the cluster.
2. Add a cluster instance.

   ```
   add cluster instance <clId> -quorumType <NONE | MAJORITY> -inc <ENABLED
    | DISABLED> -backplanebasedview <ENABLED | DISABLED><!--NeedCopy-->
   ```

   > **Note**
   >
   > - The cluster instance ID must be unique within a LAN.
   > - The `-quorumType` parameter must be set to MAJORITY and not NONE in the following

scenarios:
  - Topologies which do not have redundant links between cluster nodes. These topologies might be prone to network partition due to a single point of failure.
  - During any cluster operations such as node addition or removal.
- For an L3 cluster, make sure the `-inc` parameter is set to ENABLED. The `-inc` parameter must be disabled for an L2 cluster.
- When the `-backplanebasedview` parameter is enabled, the operational view (set of nodes that serve traffic) is decided based on heartbeats received only on the backplane interface. By default, this parameter is disabled. When this parameter is disabled, a node does not depend on the heartbeat reception only on the backplane.

3. [Only for an L3 cluster] Create a node group. In the next step, the newly added cluster node must be associated with this node group.

> **Note**
>
> This node group includes all or a subset of the Citrix ADC appliances that belong to the same network.

```
add cluster nodegroup <name><!--NeedCopy-->
```

4. Add the Citrix ADC appliance to the cluster.

"'add cluster node -state -backplane -nodegroup

```
> **Note** For an L3 cluster:
>
>-  The node group parameter must be set to the name of the node group that is created.
>-  The backplane parameter is mandatory for nodes that are associated with a node group that has more than one node, so that the nodes within the network can communicate with each other.</span>

Example:

Adding a node for an L2 cluster (all cluster nodes are in the same network).
```

add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1

```
  Adding a node for an L3 cluster which includes a single node from each network. Here, you do not have to set the backplane .
```

add cluster node 0 10.102.29.60 -state PASSIVE -nodegroup ng1


```
Adding a node for an L3 cluster which includes multiple nodes
   from each network. Here, you have to set the backplane so
   that nodes within a network can communicate with each other.
```

add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1 -nodegroup ng1
“‘

5. Add the cluster IP address (for example, 10.102.29.61) on this node.

```
add ns ip <IPAddress> <netmask> -type clip
<!--NeedCopy-->
```

Example

```
> add ns ip 10.102.29.61 255.255.255.255 -type clip
<!--NeedCopy-->
```

6. Enable the cluster instance.

```
enable cluster instance <clId><!--NeedCopy-->
```

7. Save the configuration.

```
save ns config<!--NeedCopy-->
```

8. Warm reboot the appliance.

```
reboot -warm<!--NeedCopy-->
```

Verify the cluster configurations by using the show cluster instance command. Verify that the output of the command displays the NSIP address of the appliance as a node of the cluster.

9. After the node is UP, login to the CLIP and change RPC credentials for both cluster IP address and Node IP address. For more information about changing an RPC node password, see Change an RPC node password.

To create a cluster by using the GUI

1. Log on to an appliance (for example, an appliance with NSIP address 10.102.29.60) that you intend to add to the cluster.
2. Navigate to **System > Cluster.**
3. In the details pane, click the **Manage Cluster** link.
4. In the Cluster Configuration dialog box, set the parameters required to create a cluster. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click **Create.**
6. In the Configure cluster instance dialog box, select the Enable cluster instance check box.

7. In the Cluster Nodes pane, select the node and click **Open.**
8. In the Configure Cluster Node dialog box, set the State.
9. Click **OK**, and then click **Save.**
10. Warm reboot the appliance.
11. After the node is UP, login to the CLIP and change RPC credentials for both cluster IP address and Node IP address. For more information about changing an RPC node password, see Change an RPC node password.

Strict mode support for sync status of the cluster

You can now configure a cluster node to view errors when applying the configuration. A new parameter, “syncStatusStrictMode” is introduced in both the add and set cluster instance command to track the status of each node in a cluster. By default, the “syncStatusStrictMode” parameter is disabled.

To enable the strict mode by using the CLI

At the command prompt, type:

```
set cluster instance <clID> [-syncStatusStrictMode (ENABLED | DISABLED)]
```

Example:

```
set cluster instance 1 -syncStatusStrictMode ENABLED
```

To view the status of strict mode by using the CLI

```
>show cluster instance
1) Cluster ID: 1
    Dead Interval: 3 secs
    Hello Interval: 200 msecs
    Preemption: DISABLED
    Propagation: ENABLED
    Quorum Type: MAJORITY
    INC State: DISABLED
    Process Local: DISABLED
    Retain Connections: NO
    Heterogeneous: NO
    Backplane based view: DISABLED
    Cluster sync strict mode: ENABLED
    Cluster Status: ENABLED(admin), ENABLED(operational), UP

  WARNING(s):
    (1) - There are no spotted SNIPs configured on the cluster.
       Spotted SNIPs can help improve cluster performance
```



```
18
19          Member Nodes:
20          Node ID     Node IP      Health     Admin State  Operational
               State
21          -------     -------      ------     -----------
               -----------------
22      1)      1           192.0.2.20      UP          ACTIVE       ACTIVE(
           Configuration Coordinator)
23      2)      2           192.0.2.21      UP          ACTIVE       ACTIVE
24      3)      3           192.0.2.19*     UP          ACTIVE       ACTIVE
25  <!--NeedCopy-->
```

To view the sync failure reason of a cluster node by using the GUI

1. Navigate to **System > Cluster > Cluster Nodes**.
2. In the **Cluster Nodes** page, scroll to the extreme right to view the details of the synchronization failure reason of the cluster nodes.

Adding a node to the cluster

September 14, 2021

You can seamlessly scale the size of a cluster to include a maximum of 32 nodes. When a Citrix ADC appliance is added to the cluster, the configurations from that appliance are cleared (by internally running the clear ns config -extended command). The SNIP addresses, **MTU** settings of the backplane interface, and all VLAN configurations (except the default VLAN and NSVLAN) are also cleared from the appliance.

The cluster configurations are then synchronized on this node. There can be an intermittent drop in traffic while the synchronization is in progress.

> **Important**
>
> Before you add a Citrix ADC appliance to a cluster:
>
> - Set up the backplane interface for the node. Check preceding topic.
> - Check if the licenses that are available on the appliance match that are available on the configuration coordinator. The appliance is added only if the licenses match.
> - If you want the NSVLAN on the cluster, make sure that the NSVLAN is created on the appliance before it is added to the cluster.
> - Citrix recommends that you add the node as a passive node. Then, after joining the node to

the cluster, complete the node specific configuration from the cluster IP address. Run the force cluster sync command if the cluster has only spotted IP addresses. And which has L3 VLAN binding, or has static routes.

- When an appliance with a preconfigured link aggregate (LA) channel is added to a cluster, the LA channel continues to exist in the cluster environment. The LA channel is renamed from LA/x to nodeId/LA/x, where LA/x is the LA channel identifier.

To add a node to the cluster by using the CLI

Note

When you add a node to a cluster setup, if the node has a default static route, it gets added to the cluster coordinator node (CCO). If this default static route points to an incorrect gateway, it might result in downtime of the services. Hence, verify the default static route of the new node, before adding it to the cluster setup.

1. Log on to the cluster IP address, at the command prompt, do the following:
   - Add the appliance (for example, 10.102.29.70) to the cluster.

   Note

   For an L3 cluster:

   - The node group parameter must be set to a node group that has nodes of the same network.
   - If this node belongs to the same network as the first node that was added, then configure the node group that was used for that node.
   - If this node belongs to a different network, then create a node group and bind this node to the node group.
   - The backplane parameter is mandatory for nodes that are associated with a node group that has more than one node, so that the nodes within the network can communicate with each other.

```
add cluster node <nodeId> <IPAddress> -state <state> -backplane <
   interface_name> -nodegroup <name>

Example:

add cluster node 1 10.102.29.70 -state PASSIVE -backplane 1/1/1
<!--NeedCopy-->
```

   - Save the configuration.

```
save ns config
<!--NeedCopy-->
```


2. Log on to the newly added node (for example, 10.102.29.70) and join the node to the cluster.

```
join cluster -clip <ip_addr> -password <password>

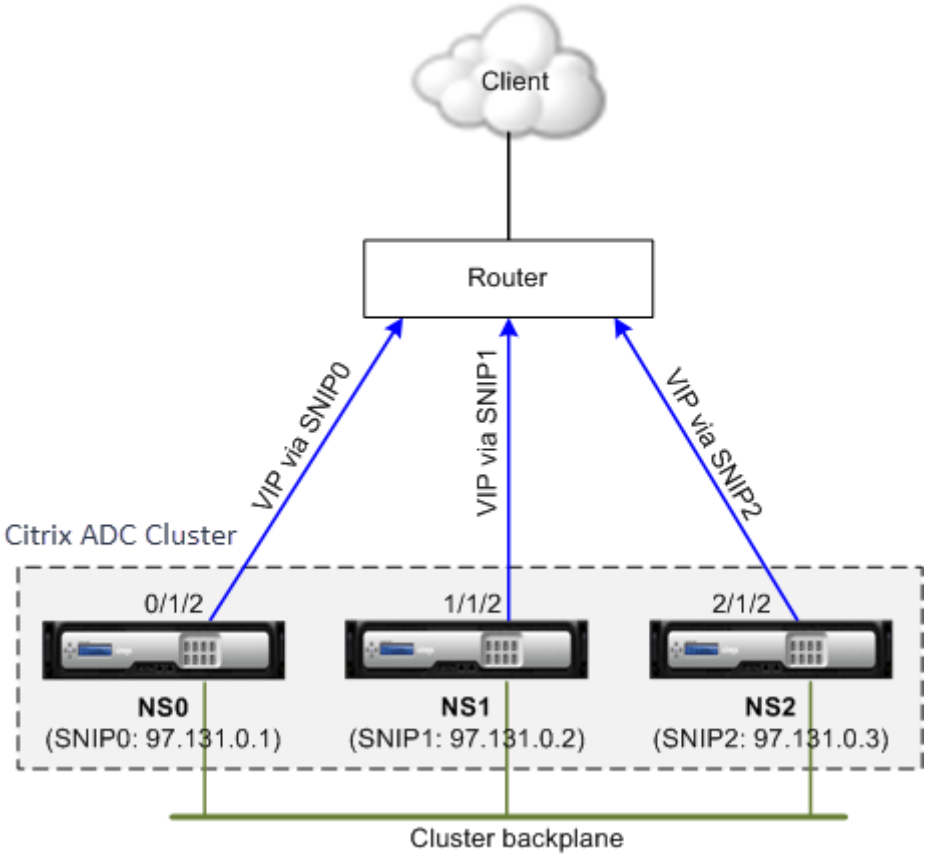
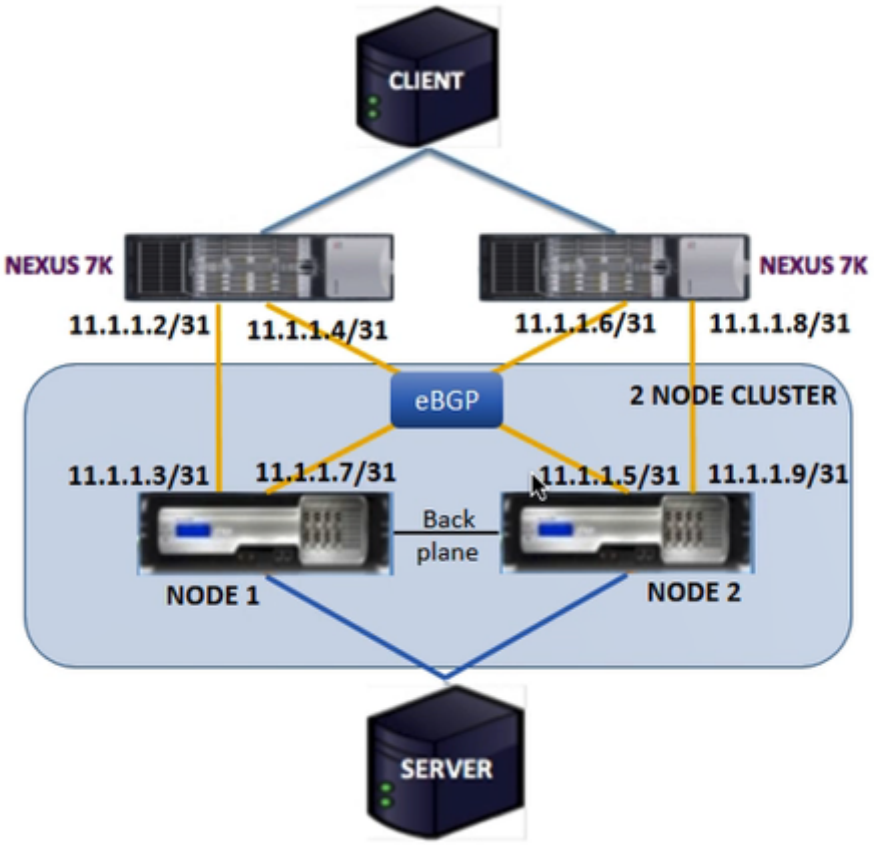
Example:

join cluster -clip 10.102.29.61 -password nsroot
<!--NeedCopy-->
```

3. Configure the following commands on the CLIP.

   - Bind VLAN to an interface

```
bind vlan <id> -ifnum <interface_name>
<!--NeedCopy-->
```

     Example:

```
bind vlan 1 -ifnum 2/1/2
<!--NeedCopy-->
```

   - Add spotted IP address to the newly added node

```
add ns ip <IpAddress> <netmask> -ownerNode <positive_interger
    >
<!--NeedCopy-->
```

     Example:

```
add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2
<!--NeedCopy-->
```

   - Verify VLAN on NSIP

```
show vlan <id>
<!--NeedCopy-->
```

     Example:

```
show vlan 1
<!--NeedCopy-->
```

4. Perform the following configurations:

   - If the node is added to a cluster that has only spotted IPs, the configurations are synchronized before the spotted IP addresses are assigned to that node. In such cases, L3 VLAN bindings can be lost. To avoid this loss, either add a striped IP or add the L3 VLAN bindings.

- Define the required spotted configurations.
- Set the MTU for the backplane interface.

5. Save the configuration.

```
save ns config
<!--NeedCopy-->
```

6. Warm reboot the appliance.

```
reboot -warm
<!--NeedCopy-->
```

7. After the node is UP and sync is successful, change RPC credentials for the node from the cluster IP address. For more information about changing an RPC node password, see Change an RPC node password.

```
set rpcNode <node-NSIP> -password <passwd>

Example:

set rpcNode 192.0.2.4 -password mypassword
<!--NeedCopy-->
```

8. Set the cluster node to Active.

```
set cluster node <nodeID> -state active.

Example:

set cluster node 1 -state active
<!--NeedCopy-->
```

To add a node to the cluster by using the GUI

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes**.
3. In the details pane, click **Add** to add the new node (for example, 10.102.29.70).
4. In the **Create Cluster Node** dialog box, configure the new node. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click **Create**. When prompted to perform a warm reboot, click **Yes.**
6. After the node is UP and sync is successful, change RPC credentials for the node from the cluster IP address. For more information about changing an RPC node password, see Change an RPC node password.

7. Navigate to **System > Cluster > Nodes > Edit**.
8. Modify the State to **ACTIVE** and confirm.

To join a previously added node to the cluster by using the GUI

If you have used the CLI to add a node to the cluster, but have not joined the node to the cluster, you can use the following procedure.

> **Note**
>
> When a node joins the cluster, it takes over its share of traffic from the cluster and hence an existing connection can get terminated.

1. Log on to the node that you want to join to the cluster (for example, 10.102.29.70).
2. Navigate to **System > Cluster.**
3. In the details pane, under Get Started, click the Join Cluster link.
4. In the Join to existing cluster dialog box, set the cluster IP address and the `nsroot` password of the configuration coordinator. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click **OK.**

Viewing the details of a cluster

September 14, 2021

You can view the details of the cluster instance and the cluster nodes by logging on to the cluster IP address.

To view details of a cluster instance by using the CLI

Log on to the cluster IP address and, at the command prompt, type:

```
show cluster instance <clId>
```

> **Note**
>
> When the preceding command is run from the NSIP address of the non-CCO node, the command displays the status of the cluster on this node.

To view details of a cluster node by using the CLI

Log on to the cluster IP address and, at the command prompt, type:


```
show cluster node <nodeId>
```

To view details of a cluster instance by using the GUI

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**
3. In the details pane, under **Get Started**, click the **Manage Cluster** link to view the details of the cluster.

To view details of a cluster node by using the GUI

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**
3. In the details pane, click the node for which you want to view the details.

Distributing traffic across cluster nodes

September 14, 2021

After you have created the Citrix ADC cluster and performed the required configurations, you must deploy Equal Cost Multiple Path (ECMP) or cluster Link Aggregation (LA) on the client data plane (for client traffic) or server data plane (for server traffic). These mechanisms distribute external traffic across the cluster nodes.

Policy-based backplane steering

The policy-based backplane steering (PBS) is a mechanism in cluster deployment, which steers the traffic across cluster nodes based on the hash method defined for the flow. The flow is defined by a combination of L2 and L3 parameters similar to the Access Control List (ACL).

The PBS supports both IPv4 and IPv6 traffic. In the case of IPv6 deployments, the steering supports an extra option `[dfdprefix <positive_integer>]`. It provides the flexibility to choose the same flow processor for the same IP prefix. The prefix option is supported for source IP or destination IP hash methods only.

> **Note**
>
> If the PBS mechanism is not used to steer the traffic, the traffic is steered through the default method.

To configure the new ACL attributes, at the CLI, type the following commands:

CLI commands for IPv4

- add ns acl <aclname> <aclaction> [-type (classic | dfd)] [-dfdhash <dfdhash>]
- set ns acl <aclname> <aclaction> [-dfdhash <dfdhash>
- show ns acl [<aclname>][-type (classic | DFD)]
- apply ns acls [-type (classic | DFD)]
- clear ns acls [-type (classic | DFD)]
- renumber ns acls [-type (classic | DFD)]

CLI commands for IPv6

- add ns acl6 <acl6name> <acl6action> [-type (classic | dfd)][-dfdhash <dfdhash>][-dfdprefix <positive_interger>]
- set ns acl6 <acl6name> <acl6action> [-dfdhash <dfdhash>][-dfdprefix <positive_interger>]
- show ns acl6 [<acl6name>][-type (classic | DFD)]
- apply ns acls6 [-type (classic | DFD)]
- clear ns acls6 [-type (classic | DFD)]
- renumber ns acls6 [-type (classic | DFD)]

The following are the different types of hash methods that you can specify to steer the packet to the flow processor:

- SIP-SPORT-DIP-DPORT
- SIP
- DIP
- SIP-DIP
- SIP-SPORT

Limitations

1. The traffic flow distribution across the cluster nodes is not ensured, as the flow processor is decided by the admin configured rules.
2. L2 mode is not supported.
3. The node groups and striped SNIPs are not supported, as there is no deployment scenarios.
4. MPTCP is not supported.
5. Support only for TCP, UDP, and ICMP traffic.
6. Cluster over L3 mode is not supported.
7. Process local at service level is not supported.

Using the Equal Cost Multiple Path (ECMP)

September 14, 2021

By using the Equal Cost Multiple Path (ECMP) mechanism on a cluster deployment, active cluster nodes advertise the virtual server IP addresses. The cluster node which receives the advertised traffic steers the traffic to the node that must process the traffic. There can be redundant steering in spotted and partially striped virtual servers. Therefore, from NetScaler 11 onwards, spotted and partially striped virtual server IP addresses advertise the owner nodes, which reduce the redundant steering.

You must have detailed knowledge of routing protocols to use ECMP. For more information, see Configuring Dynamic Routes. For more information on routing in a cluster, see Routing in a Cluster.

To use ECMP, you must first perform the following:

- Enable the required routing protocol (OSPF, RIP, BGP, or ISIS) on the cluster IP address.
- Bind the interfaces and the spotted IP address (with dynamic routing enabled) to a VLAN.
- Configure the selected routing protocol and redistribute the kernel routes on the ZebOS by using the VTYSH shell.

Perform similar configurations on the cluster IP address and on the external connecting device.

> **Note**
>
> - Make sure that the licenses on the cluster support dynamic routing, otherwise ECMP does not work.
> - ECMP is not supported for wildcard virtual servers since RHI needs a VIP address to advertise to a router and wildcard virtual servers. As they do not have associated VIP addresses.

Figure 1. ECMP topology



When you use the ECMP mechanism for traffic distribution on a cluster deployment, the active cluster nodes advertise the virtual server IP addresses to the upstream router. The ECMP router can reach the VIP address via SNIP0, SNIP1, or SNIP2. The traffic flow in the Figure 1 is described as follows:

1. The client sends a request to the VIP hosted on the cluster.
2. The upstream router, based on the learned routes of the VIP, forwards the packet to any one of the nodes. Let's say NS1. The node NS1 is the flow receiver.
3. The flow receiver (NS1) determines the node that must process the traffic, which is called the flow processor. For example, Node NS2 is the flow processor.
4. The flow receiver (NS1) with SNIP1 (97.131.0.2) steers the request to the flow processor (NS2) with SNIP2 (97.131.0.3).
5. The flow processor (NS2) establishes a connection with the server.
6. The server processes the request and sends the response to the SNIP address that sent the request to the server.

Notes:

- Only ACTIVE nodes advertise VIP routes.
- INACTIVE nodes do not advertise VIP routes.
- All ACTIVE nodes advertise striped VIPs.

- Only ACTIVE owner nodes advertise spotted or partially striped VIPs.

To configure ECMP on the cluster by using the command line interface

1. Log on to the cluster IP address.
2. Enable the routing protocol.

```
enable ns feature <feature>
```

Example: To enable the OSPF routing protocol.

```
enable ns feature ospf
```

3. Add a VLAN.

```
add vlan <id>
```

Example

```
add vlan 97
```

4. Bind the interfaces of the cluster nodes to the VLAN.

```
bind vlan <id> -ifnum <interface_name>
```

Example

```
bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
```

5. Add a spotted SNIP address for each node and enable dynamic routing on it.

```
add ns ip <SNIP> <netmask> -ownerNode <positive_integer> -
   dynamicRouting ENABLED
```

Example

```
add ns ip 97.131.0.1 255.0.0.0 -ownerNode 0 -dynamicRouting
   ENABLED -type SNIP
add ns ip 97.131.0.2 255.0.0.0 -ownerNode 1 -dynamicRouting
   ENABLED -type SNIP
add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2 -dynamicRouting
   ENABLED -type SNIP
```


6. Bind one of the spotted SNIP addresses to the VLAN. When you bind one spotted SNIP address to a VLAN, all other spotted SNIP addresses defined on the cluster in that subnet are automatically bound to the VLAN.

```
bind vlan <id> -IPAddress <SNIP> <netmask>
```

Example

```
bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
```

> **Note**
>
> You can use NSIP addresses of the cluster nodes instead of adding SNIP addresses. If so, you do not have to perform steps 3–6.

7. Configure the routing protocol on ZebOS using the VTYSH shell.

Example:

To configure an OSPF routing protocol on node IDs 0, 1, and 2.

```
vtysh
! interface vlan97 !
 router ospf   owner-node 0
ospf router-id 97.131.0.1   exit-owner-node
owner-node 1    ospf router-id 97.131.0.2
exit-owner-node
owner-node 2
ospf router-id 97.131.0.3   exit-owner-node   redistribute kernel
   network 97.0.0.0/8 area 0  !
```

> **Note**
>
> For VIP addresses to be advertised, RHI setting is done by using the vserverRHILevel parameter as follows:

```
add ns ip <IPAddress> <netmask> -type VIP -vserverRHILevel <
   vserverRHILevel>
```

For OSPF specific RHI settings, there are more settings that can be done as follows:

```
add ns ip <IPAddress> <netmask> -type VIP -ospfLSAType ( TYPE1 |
   TYPE5 ) -ospfArea <positive_integer>
```

Use the add ns ip6 command to perform the preceding commands on IPv6 addresses.

8. Configure ECMP on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

```
1 //For OSPF (IPv4 addresses) Global config: Configure terminal
      feature ospf    Interface config:  Configure terminal
      interface Vlan10      no shutdown      ip address 97.131.0.5/8
         Configure terminal  router ospf 1  network 97.0.0.0/8 area
      0.0.0.0  --------------------------------
2
3 //For OSPFv3 (IPv6 addresses) Global config:  Configure terminal
      feature ospfv3    Configure terminal  interface Vlan10      no
      shutdown      ipv6 address use-link-local-only      ipv6 router
       ospfv3 1 area 0.0.0.0    Configure terminal  router ospfv3 1
```

Router monitoring cluster nodes in ECMP deployment

In a cluster setup, on an owner node that has a spotted SNIP address configuration, you can now disable the ownerDownResponse option. By default, the option is enabled, allowing the node to respond to an ICMP/ARP/ICMP6/ND6 request coming from the upstream router. You can now disable this option to allow the router to monitor if a cluster node is active or inactive. When the router sends a request, if the option is disabled, it identifies the owner node to be inactive and unavailable for traffic distribution.

To configure ECMP for static routes traffic distribution by using the command line interface

```
1 add ns ip <ipddress> <netmask> -ownernode <node-id>  - ownerDownResponse
      disable
```

Use Case: ECMP with BGP routing

September 14, 2021

To configure ECMP with BGP routing protocol, perform the following steps:

1. Log on to the cluster IP address.
2. Enable BGP routing protocol.

```
1 > enable ns feature bgp
```


3. Add VLAN and bind the required interfaces.

```
> add vlan 985
> bind vlan 985 -ifnum 0/0/1 1/0/1
```

4. Add the spotted IP address and bind them to the VLAN.

```
> add ns ip 10.100.26.14 255.255.255.0 -ownerNode 1 -
   dynamicRouting ENABLED
> add ns ip 10.100.26.15 255.255.255.0 -ownerNode 2 -
   dynamicRouting ENABLED
> bind vlan 985 -ipAddress 10.100.26.10 255.255.255.0
```

5. Configure BGP routing protocol on ZebOS using the VTYSH shell.

```
> vtysh conf t router bgp 65535 neighbor 10.100.26.1 remote-as
   65535
```

6. Configure BGP on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

```
> router bgp 65535 no synchronization
  bgp log-neighbor-changes neighbor 10.100.26.14 remote-as 65535
     neighbor 10.100.26.15 remote-as 65535 no auto-summary
  dont-capability-negotiate
  dont-capability-negotiate
  no dynamic-capability
```

Configuration of cluster ECMP by using Cisco Nexus 7000 switch with routing Protocol

September 14, 2021

With ECMP over a cluster setup, a Citrix ADC appliance is able to handle the traffic through a routing protocol. The ECMP mechanism helps in advertising the virtual server IP addresses through all active cluster nodes.

To use ECMP, you must first enable the BGP protocol on the cluster IP address. Bind the interfaces and the spotted IP address (with dynamic routing enabled) to a VLAN. Configure the selected routing protocol and redistribute the kernel routes on the ZebOS by using the VTYSH shell.

Use Case: Cluster ECMP by using Cisco Nexus 7000 switch with routing Protocol

Consider an example of a cluster deployment with a Cisco Nexus 7000 switch:

- Two Citrix ADC appliances (Node 1 and Node 2), connected to the Nexus switch (upstream).
- Two Cisco Nexus 7000 switch.
- Client and server (drawing HTTP traffic through the Nexus switch). With Hot Standby Router Protocol (HSRP) enabled on the client-side.



Prerequisites

Consider the following points before configuring cluster nodes on a Citrix ADC appliance.

1. All appliances must be of the same platform type.
2. Border Gateway Protocol (BGP) must be enabled on the cluster nodes.

Configuring by using the CLI on a Citrix ADC appliance

1. Log on to an appliance (for example, appliance with NSIP address 1.1.1.1)
2. To add a cluster node.


```
1 add cluster node 0 1.1.1.2 -state ACTIVE -backplane 0/10/8
```

3. To add the cluster IP address

```
1 add ns ip 1.1.1.10 255.255.255.254 -type clip
```

4. Save the configuration

```
1 save ns config
```

5. Warm reboot the appliance

```
1 reboot -warm
```

6. To add node 1 using CLIP

```
1 add cluster node 1 2.2.2.2 -state ACTIVE -backplane 1/10/8
```

7. To join a node to the cluster

```
1 join cluster -clip 1.1.1.10 -password nsroot
```

8. Perform the following configuration on CLIP
   - `enable ns feature bgp ospf DYNAMICROUTING`
   - `add ns ip 11.1.1.3 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0`
   - `add ns ip 11.1.1.7 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0`
   - `add ns ip 11.1.1.5 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`
   - `add ns ip 11.1.1.9 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`

On the Cisco Nexus router (11.1.1.2/31 and 11.1.1.4/31), you must perform the following configurations by using the command line:

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`


```
> interface vlan100
  no shutdown
  ip address 50.1.1.1/8
   hsrp 50
    ip 50.50.50.50

> interface Ethernet 4/15
   ip address 11.1.1.2/31
   no shutdown

> interface Ethernet 4/19
   ip address 11.1.1.4/31
   no shutdown

> interface Ethernet 4/22
   switchport
    switchport access vlan 100
```

On the Cisco Nexus router (11.1.1.6/31 and 11.1.1.8/31), you must perform the following configurations by using the command line:

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`

```
> interface vlan100
  no shutdown
  no ip redirects
  ip address 50.1.1.2/8
   hsrp 50
   ip 50.50.50.50

> interface Ethernet 4/13
  ip address 11.1.1.6/31
  no shutdown

> interface Ethernet 4/15
  ip address 11.1.1.8/31
  no shutdown

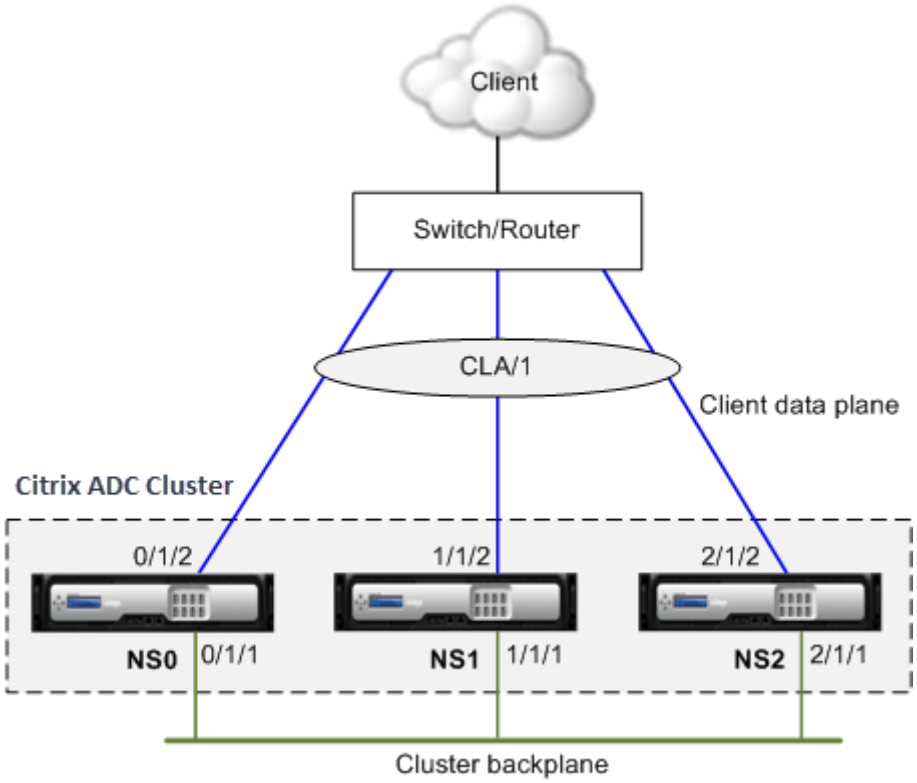
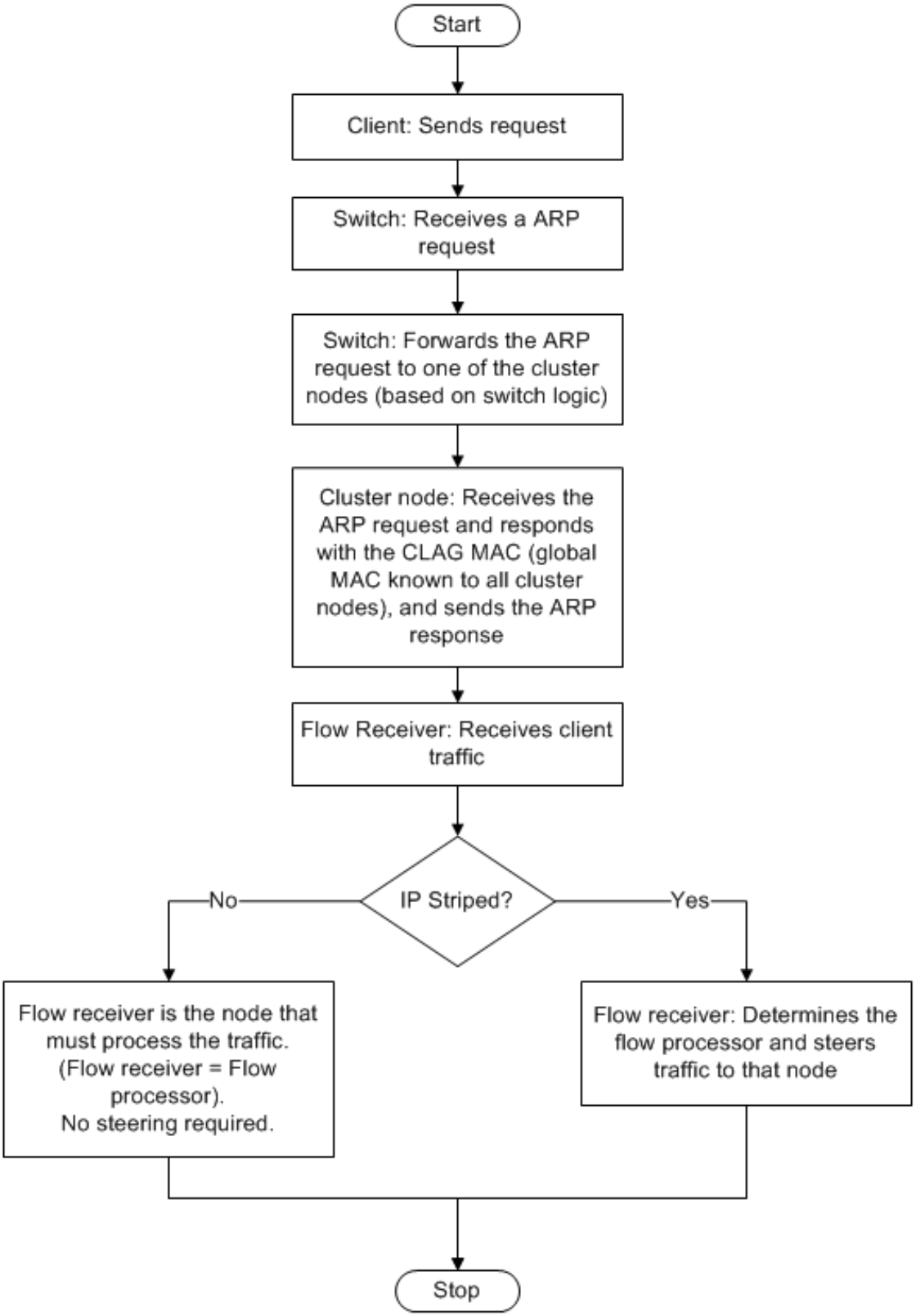
> interface Ethernet 4/22
  switchport
```



```
18       switchport access vlan 100
```

For the BGP protocol, you must perform the following configurations on CLIP of the Citrix ADC appliance:

```
1  >  vtysh
2  ns# router bgp 1
3   redistribute kernel
4   owner-node 0
5   neighbor 11.1.1.2 remote-as 2
6   neighbor 11.1.1.2 as-origination-interval 1
7   neighbor 11.1.1.2 advertisement-interval 0
8   neighbor 11.1.1.6 remote-as 2
9   neighbor 11.1.1.6 as-origination-interval 1
10  neighbor 11.1.1.6 advertisement-interval 0
11 owner-node 1
12  neighbor 11.1.1.4 remote-as 2
13  neighbor 11.1.1.4 as-origination-interval 1
14  neighbor 11.1.1.4 advertisement-interval 0
15  neighbor 11.1.1.8 remote-as 2
16  neighbor 11.1.1.8 as-origination-interval 1
17  neighbor 11.1.1.8 advertisement-interval 0
18 exit-owner-node
```

Perform the following configurations on the Cisco Nexus router (11.1.1.3 and 11.1.1.5)

```
1  > ip access-list acl1
2    10 permit ip 50.0.0.0/8 any
3    route-map test permit
4    match ip address acl1
5   router bgp 2
6    address-family ipv4 unicast
7     redistribute direct route-map test
8     maximum-paths 2
9   neighbor 11.1.1.3 remote-as 1
10   address-family ipv4 unicast
11   neighbor 11.1.1.5 remote-as 1
12  address-family ipv4 unicast
```

Perform the following configurations on the Cisco Nexus router (11.1.1.7 and 11.1.1.9)

```
1  > ip access-list acl1
2     10 permit ip 50.0.0.0/8 any
3    route-map test permit 1
4     match ip address acl1
```



```
  router bgp 2
   address-family ipv4 unicast
   redistribute direct route-map test
   maximum-paths 2
  neighbor 11.1.1.7 remote-as 1
   address-family ipv4 unicast
  neighbor 11.1.1.9 remote-as 1
   address-family ipv4 unicast
```

For the OSPF protocol, you must perform the following configurations on CLIP of the Citrix ADC appliance:

```
> vtysh
ns# router osfp 1
redistribute kernel
owner-node 0
 network 15.1.1.2/31 area 0
 network 15.1.1.6/31 area 0
exit-owner-node

owner-node 1
 network 15.1.1.4/31 area 0
 network 15.1.1.8/31 area 0
exit-owner-node

route-map map2 permit 1
set metric 10
```

On the Cisco Nexus router (11.1.1.2/31 and 11.1.1.4/31), you must perform the following configurations by using the command line:

```
> route-map- map2 permit 1
   set metric 10

   interface Ethernet4/15
     ip address 15.1.1.2/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   interface Ethernet4/19
     ip address 15.1.1.4/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

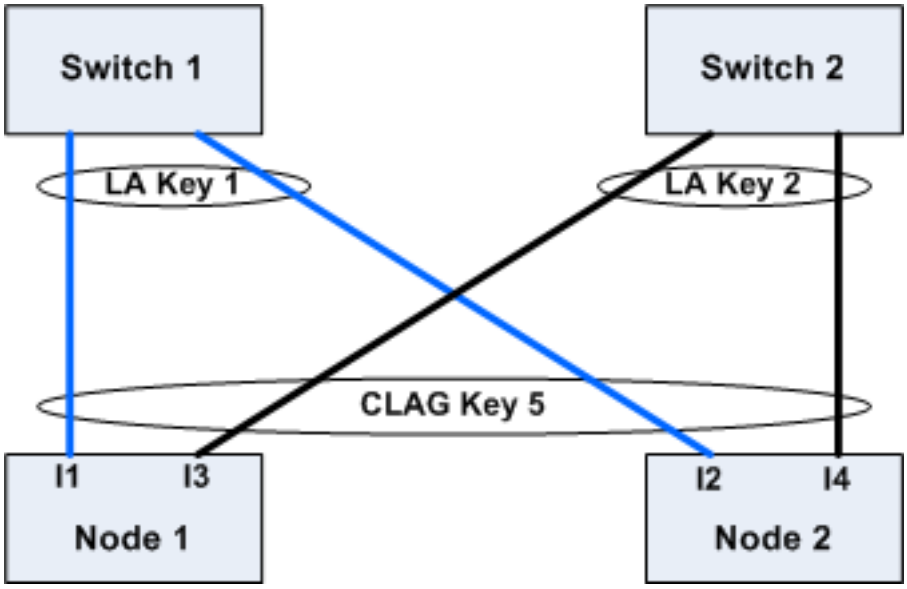
   router ospf 1
```



```
15      router-id 1.1.1.1
16      redistribute direct route-map map2
```

On the Cisco Nexus router (11.1.1.7/31 and 11.1.1.9/31), you must perform the following configurations by using the command line:

```
1  > route-map- map2 permit 1
2      set metric 10
3
4     interface Ethernet4/13
5       ip address 15.1.1.6/31
6       ip router ospf 1 area 0.0.0.0
7       no shutdown
8
9      interface Ethernet4/15
10      ip address 15.1.1.8/31
11      ip router ospf 1 area 0.0.0.0
12      no shutdown
13
14     router ospf 1
15       router-id 1.1.1.2
16       redistribute direct route-map map2
```

Using cluster link aggregation

September 14, 2021

Cluster link aggregation is a group of interfaces of cluster nodes. It is an extension of Citrix ADC link aggregation. The only difference is that, while link aggregation requires the interfaces to be from the same device, in cluster link aggregation, the interfaces are from different nodes of the cluster. For more information about link aggregation, see Configuring Link Aggregation.

> **Important**
>
> - Cluster link aggregation is supported for a cluster of hardware (MPX) appliances.
> - Cluster link aggregation is supported for a cluster of virtual (VPX) appliances that are deployed on ESX and KVM hypervisors, with the following restrictions:
> - Dedicated interfaces must be used. It means that the interfaces must not be shared with other virtual machines.
> - When a node becomes INACTIVE, the corresponding cluster LA interface is marked as power

DOWN, so that the data traffic is not sent to an INACTIVE node.

- When a node becomes ACTIVE, the corresponding cluster LA interface is marked as power ON.
- If the cluster link aggregation member interfaces are manually disabled or if cluster link aggregation itself is manually disabled, then interface power down capability is achieved only by the LACP timeout mechanism.
- Jumbo MTU is not supported on LACP cluster link aggregation.

Note: Cluster link aggregation is not supported on VPX appliances that are deployed on XenServer, AWS, and Hyper-V.

- Starting from 12. 0 release, cluster link aggregation is supported on Citrix ADC SDX appliances.
- The number of interfaces that can be bound to cluster LA is 16 (from each node). The maximum number of interfaces in cluster LA can be (16 * n), where n is the number of nodes in a cluster. The total number of interfaces in cluster LA depends on the number of interfaces for every port channel on the upstream switch.
- If a Citrix ADC appliance uses Intel Fortville interfaces, the switchover of a cluster node to passive mode might cause a few seconds of outage with CLAG. The issue occurs because LACP is enabled for CLAG to function properly, and the outage time depends on NIC LACP timers.

For example, consider a three-node cluster where all three nodes are connected to the upstream switch. A cluster LA channel (CLA/1) is formed by binding interfaces 0/1/2, 1/1/2, and 2/1/2.

Figure 1. Cluster Link Aggregation topology



A cluster LA channel has the following attributes:

- Each channel has a unique MAC agreed upon by cluster nodes.
- The channel can bind both local and remote nodes' interfaces.
- A maximum of four cluster LA channels are supported in a cluster.
- Backplane interfaces cannot be part of a cluster LA channel.
- When an interface is bound to a cluster LA channel, the channel parameters have precedence over the network interface parameters. A network interface can be bound to one channel only.
- Management access to a cluster node, must not be configured on a cluster LA channel (for example, CLA/1) or its member interfaces. This is because when the node is INACTIVE, the corresponding cluster LA interface is marked as power down, and therefore looses management access.

Figure 2. Traffic distribution flow using cluster LA



Backup and restore support of cluster LA on Citrix ADC MPX

You can backup and restore the cluster setup of LA on Citrix ADC MPX. The cluster LA MAC address is independent of the physical interface MAC address of the cluster nodes, and can change after the backup and restore process. The cluster LA can serve the traffic after a cluster restore process is complete. For more information about backup and restore, see Backup and restore of cluster setup

Static cluster link aggregation

September 14, 2021

You must configure a static cluster LA channel on the cluster IP address and on the external connecting device. If possible, configure the upstream switch to distribute traffic based on IP address or port instead of MAC address.

To configure a static cluster LA channel by using the CLI

1. Log on to the cluster IP address.

   > **Note**
   >
   > Make sure that you configure the cluster LA channel on the cluster IP address before configuring link aggregation on the external switch. Otherwise, the switch forwards traffic to the cluster even though the cluster LA channel is not configured. It can lead to loss of traffic.

2. Create a cluster LA channel.

   ```
   add channel <id> -speed <speed>
   ```

   Example

   ```
   add channel CLA/1 -speed 1000
   ```

   > **Note**
   >
   > You must not specify the speed as AUTO. Rather, you must explicitly specify the speed as 10, 100, 1000, or 10000. Only interfaces that have the speed matching the `<speed>` attribute in the cluster LA channel are added to the active distribution list.

3. Bind the required interfaces to the cluster LA channel. Make sure that the interfaces are not used for the cluster backplane.

   ```
   bind channel <id> <ifnum>
   ```


Example

```
bind channel CLA/1 0/1/2 1/1/2 2/1/2
```

4. Verify the configurations.

```
show channel <id>
```

Example

```
show channel CLA/1
```

> **Note**
>
> You can bind the cluster LA channel to a VLAN by using the `bind vlan` command. The interfaces of the channel are automatically bound to the VLAN.

5. Configure static LA on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1). Similar configurations must be performed on other switches.

```
Global config:
Configure terminal

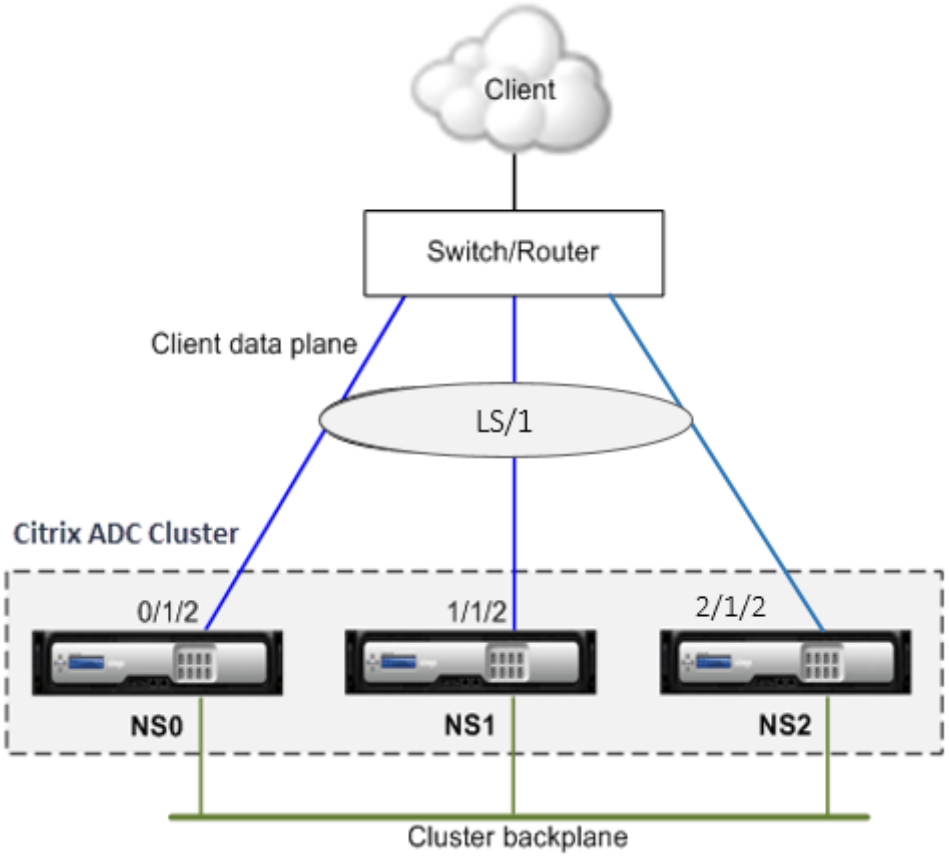
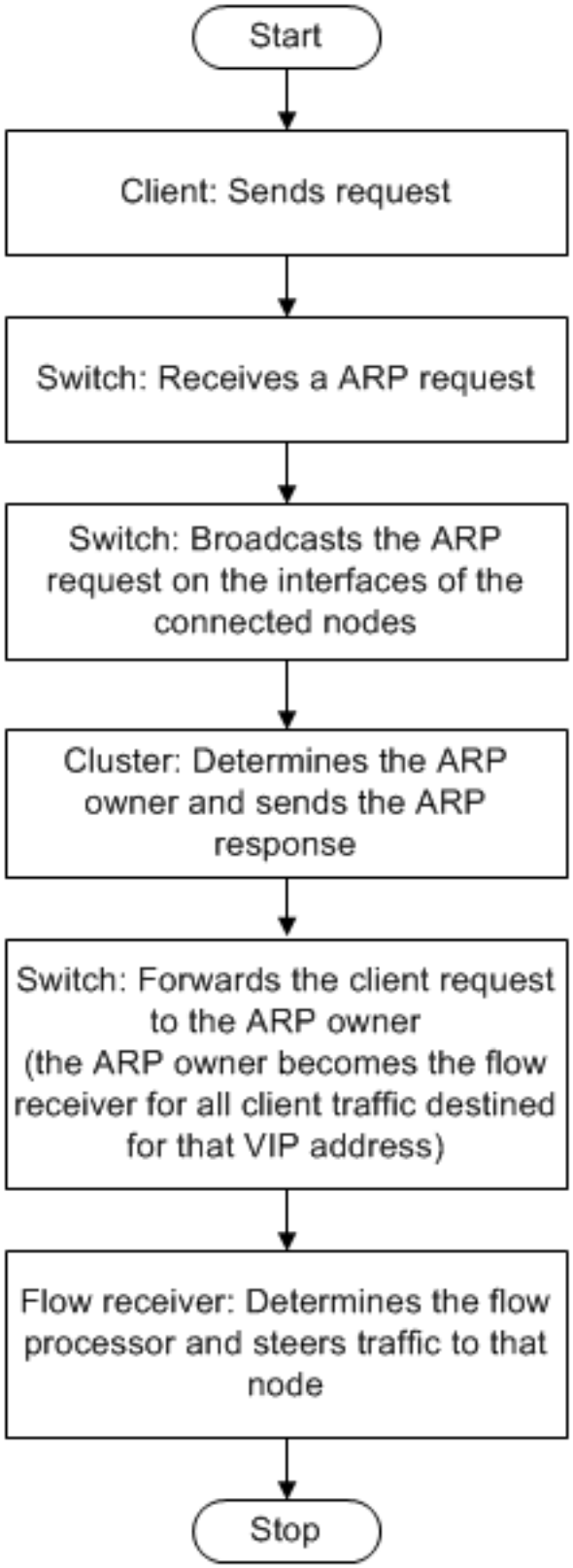
Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown
```

Dynamic cluster link aggregation

September 14, 2021

Dynamic cluster LA channel uses the Link Aggregation Control Protocol (LACP).

You must perform similar configurations on the cluster IP address and on the external connecting device. If possible, configure the upstream switch to distribute traffic based on IP address or port instead of MAC address.

> **Points to remember**
>
> - Enable LACP (by specifying the LACP mode as either ACTIVE or PASSIVE).

```
>**Note**
>
> Make sure the LACP mode is not set as PASSIVE on both the Citrix ADC
    cluster and the external connecting device.
```

> - Specify the same LACP key on each interface that you want to be the part of the channel. For creating a cluster LA channel, the LACP key can have a value from 5 through 8. For example, if you set the LACP key on interfaces 0/1/2, 1/1/2, and 2/1/2 to 5, CLA/1 is created. The interfaces 0/1/2, 1/1/2, and 2/1/2 are automatically bound to CLA/1. Similarly, if you set the LACP key to 6, the CLA/2 channel is created.
> - Specify the LAG type as Cluster.

To configure a dynamic cluster LA channel by using the CLI

On the cluster IP address, for each interface that you want to add to the cluster LA channel, type:

```
set interface <id> -lacpMode <lacpMode> -lacpKey <positive_integer> -
lagType CLUSTER<!--NeedCopy-->
```

Example:

To configure a cluster LA channel CLA/1 of 3 interfaces.

```
> set interface 0/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
```

> **Note**
>
> Optionally, you can enable Link Redundancy in a Cluster with LACP.

Similarly, configure dynamic LA on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1). Similar configurations must be performed on other switches.

```
Global config:
Configure terminal
feature lacp
```



```
Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown
```

Link redundancy in a cluster with LACP

September 14, 2021

A Citrix ADC cluster provides link redundancy for LACP to ensure that all nodes have the same partner key.

To understand the need for link redundancy, let us consider the example of the following cluster setup along with the accompanying cases (with attention to case 3):



In this setup, interfaces I1, I2, I3, and I4 are bound to the LACP channel with KEY 5. On the partner side, I1 and I2 are connected to Switch 1 to form a single LA channel with KEY 1. Similarly, I3 and I4 are connected to Switch 2 to form a single LA channel with KEY 2.

Now let us consider the following cases to understand the need for link redundancy:

- **Case 1: Switch 1 is up and Switch 2 is down**

  In this case, cluster LA on both the nodes would stop receiving LACPDUs from Key2 and would start receiving LACPDUs from Key1. On both the nodes, cluster LA is connected to KEY 1 and I1 and I2 is UP and the channel on both the nodes would be UP.

- **Case 2: Switch1 goes down and Switch2 becomes UP**

  In this case, cluster LA on both the nodes would stop receiving LACPDUs from Key1 and would start receiving LACPDUs from Key2. On both the nodes, cluster LA is connected to Key2 and I3 and I4 is UP and the channel on both the nodes would be UP.

- **Case 3: Both Switch1 and Switch2 are UP**

  In this case, it is possible that cluster LA on node1 chooses Key1 as its partner and cluster LA on node2 chooses Key2 as its partner. It means that I1 on node1 and I4 on node2 are receiving traffic which is undesirable. It can happen because the LACP state machine is node-level and chooses its partners on first-come first-serve basis.

  To solve these concerns, link redundancy of dynamic cluster LA is supported. To configure link redundancy on a channel or interface, you must enable it and optionally specify the threshold throughput as follows:

  ```
  set channel CLA/1 -linkRedundancy ON -lrMinThroughput <positive_integer
  >
  ```

  The throughput of the partner channels is checked against the configured threshold throughput. The partner channel that satisfies the threshold throughput is selected in a first-in-first-out (FIFO) manner. If none of the partner channel meets the threshold, or if threshold throughput is not configured, the partner channel with the maximum number of links is selected.

  > **Note**
  >
  > The threshold throughput can be configured from NetScaler 11 onwards.

Using USIP mode in cluster

September 14, 2021

In use source IP (USIP) mode, the cluster, or Citrix ADC appliance forwards each packet to the appropriate back-end server with the client IP address.

USIP mode traffic distribution

The USIP mode behavior differs traffic distribution across client data plane and server data plane in ECMP and CLAG deployment. The following section provides more information about USIP mode behavior. For more information related to CLAG on USIP mode, see Using cluster link aggregation.

USIP mode

The cluster uses the client IP to open the server-side connection. The source port may or might not be preserved based on the `useproxyport` setting.

USIP `useproxyport` scenarios

The USIP `useproxyport` is ON for the traffic flow, the source port is selected in a way that the reverse traffic hashes to the flow processor. It ensures single steering on the server side.

The USIP `useproxyport` is OFF for the traffic flow, the source port is preserved and hence there is double steering on the server side.

> **Important**
>
> - When USIP is ON, the client IP is used in back end server connection, and traffic distribution for response is needed across cluster nodes. You can use ECMP or CLAG deployment for traffic distribution on the server side. In the absence of traffic distribution on the server side, the whole return traffic might land on a single cluster node, resulting in congestion.
> - The `set rsskeytype -rsskey symmetric` command is used to reduce double steering to single steering of traffic in the `useproxyport` off deployments. Where the 4-tuple for the connection remains the same for the server and client side. For example, wildcard MAC mode virtual server.

Limitations

The USIP does not work when the process local is disabled.

USIP mode deployment

The following figure depicts a USIP mode deployment in a cluster setup.

Configure the following using CLI

1. Enable the routing protocol.

```
1 enable ns feature <feature>
```


Example:

```
enable ns feature ospf
```

2. Add a spotted SNIP address for each node and enable dynamic routing on it.

```
add ns ip <SNIP> <netmask> -dynamicRouting ( ENABLED | DISABLED )
    -ownerNode <positive_interger>  -ownerdownResponse ( YES | NO
    )
```

Example

```
-  add ns ip 192.0.2.1 255.255.255.0 -dynamicRouting ENABLED  -
   ownerNode 0  -ownerDownResponse NO
-  add ns ip 192.0.2.2 255.255.255.0 -dynamicRouting ENABLED  -
   ownerNode 1  -ownerDownResponse NO
-  add ns ip 192.0.2.3 255.255.255.0 -dynamicRouting ENABLED  -
   ownerNode 2  -ownerDownResponse NO
```

3. Add a VLAN.

```
add vlan <id>
```

Example

```
add vlan 300
```

4. Bind the interfaces of the cluster nodes to the VLAN.

```
bind vlan <id> -ifnum <interface_name>
```

Example

```
bind vlan 300 -ifnum 0/1/2 1/1/2 2/1/2
```

5. Bind one of the spotted SNIP addresses to the VLAN. When you bind one spotted SNIP address to a VLAN, all other spotted SNIP addresses defined on the cluster in that subnet are automatically bound to the VLAN.

```
bind vlan <id> -IPAddress <ip_addr | ipv6_addr> -netmask
```

Example

```
bind vlan 300  -IPAddress 192.0.2.1 255.255.255.0
```


6. Configure the routing protocol on ZebOS using the VTYSH shell. Configure OSPF routing protocol on node IDs 0, 1, and 2.

```
vtysh
configure terminal
ns block-sec-rtadv
router ospf
owner -node 0
router-id 192.0.2.1
exit-owner-node
owner-node 1
router-id 192.0.2.2
exit-owner-node
owner-node 2
router-id 192.0.2.3
exit-owner-node
network 192.0.2.0/24 area 0

default-information originate always
```

7. Perform the following configurations on the Cisco 3750 router by using the CLI.

```
Configure terminal
feature ospf
interface vlan300
no shutdown
ip address 192.0.2.100/24
Configure terminal
router ospf 1
router-id 192.0.2.100
network 192.0.2.0 0.0.0.255 area 0
```

Notes

- Traffic distribution on client and server need not be the same. For example, you can configure ECMP on the client side and CLAG on the server side or the opposite way.
- Plan for extra capacity of the backplane as there is more steering overhead in the USIP deployment.
- The Configuration related to CLAG and Monitor Static Route (MSR) must remain the same on the server side.
- Traffic steering is more in the USIP mode deployments.

Managing the Citrix ADC cluster

September 14, 2021

After you have created a cluster and configured the required traffic distribution mechanism, the cluster is able to serve traffic. During the lifetime of the cluster, you can perform the following cluster tasks:

- Configuring node groups
- Disabling nodes of a cluster
- Discovering Citrix ADC appliances
- Viewing statistics
- Synchronizing cluster configurations, and cluster files
- Synchronizing the time across the nodes
- Upgrading or downgrading the software of cluster nodes

Configuring linksets

September 14, 2021

Linkset is a group of interfaces of cluster nodes that belong to the same broadcast domain. In linksets, each node has the information about which interfaces of other nodes are connected to the same broadcast domain.

> **Note**
>
> Linksets are a mandatory configuration in the following scenarios:
>
> - For deployments that require MAC-Based Forwarding (MBF).
> - For "-m MAC" mode that is enabled at the virtual server along with MBF mode enabled globally.
> - To improve the manageability of ACL and L2 policies involving interfaces. You define a linkset of the interfaces and add ACL and L2 policies based on linksets.

In a cluster setup, the following features use MBF internally.

- Forwarding session
- L2Conn
- MAC mode virtual server
- Transparent monitor
- LLB

Linksets must be configured only through the cluster IP address.

Consider an example with a three node cluster. In the following figure, the interfaces 0/1/2, 1/1/2, and 2/1/2 are in the same broadcast domain and therefore can be configured as linkset (LS/1).

Figure 1. Linksets topology



Figure 2. Traffic distribution flow using linksets



To configure a linkset by using the CLI

1. Log on to the cluster IP address.
2. Create a linkset.

   “‘add linkset

```
**Example**

```add linkset LS/1<!--NeedCopy-->
```

3. Bind the required interfaces to the linkset. Make sure the interfaces are not used for the cluster backplane.

   “‘bind linkset -ifnum ...

```
**Example**

```bind linkset LS/1 -ifnum 0/1/2 1/1/2 2/1/2<!--NeedCopy-->
```

4. Verify the linkset configurations.

   “‘show linkset

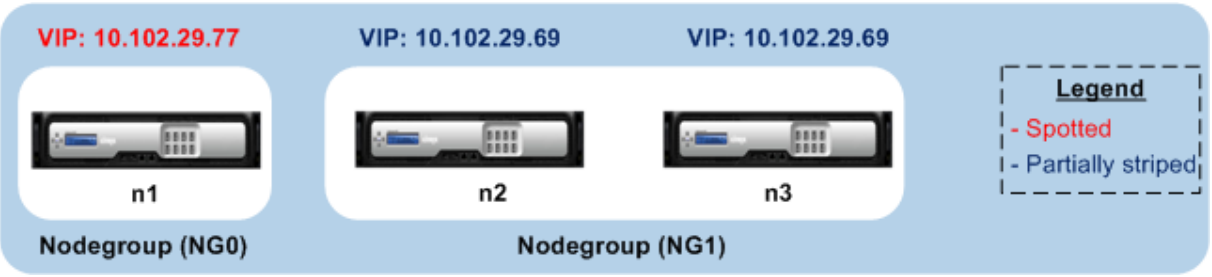
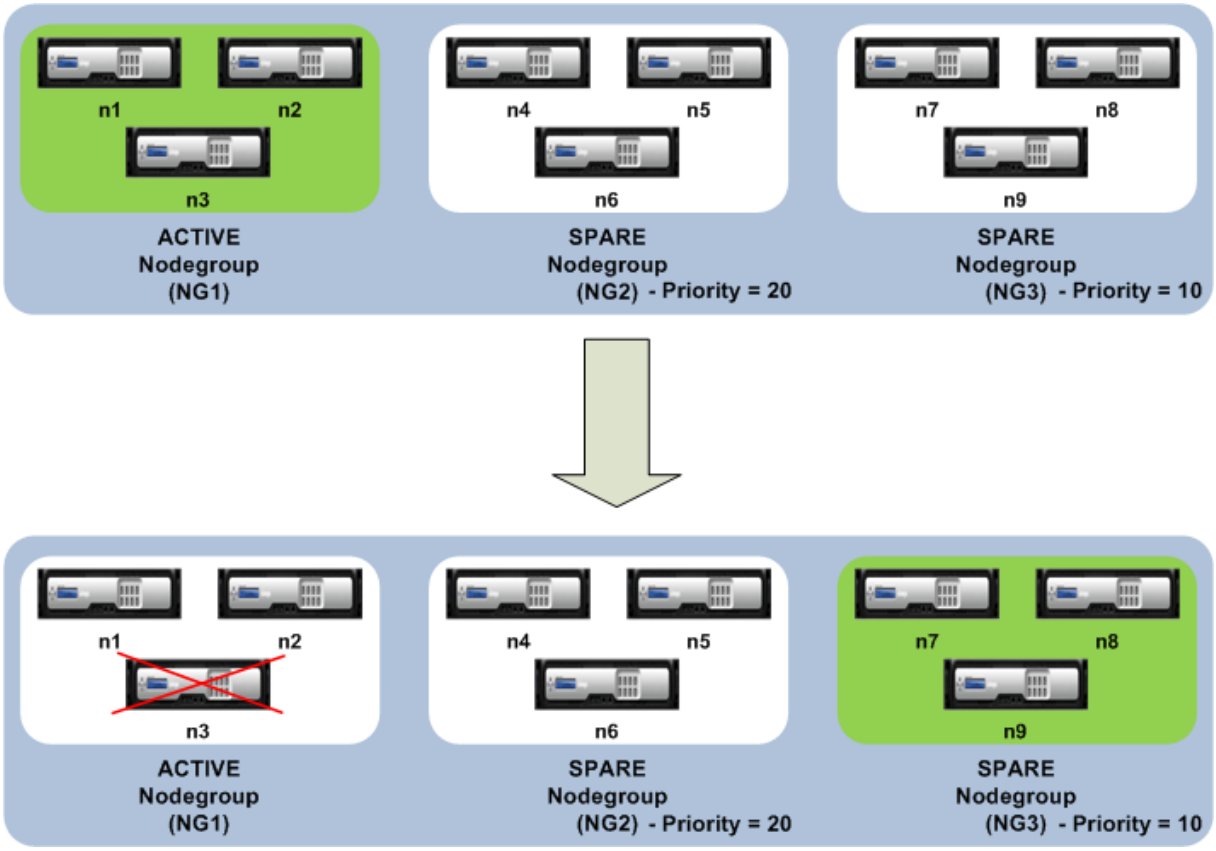
```
**Example**

```show linkset LS/1<!--NeedCopy-->
```

> **Note**
>
> You can bind the linkset to a VLAN by using the `bind vlan` command. The interfaces of the linkset are automatically bound to the VLAN.

To configure a linkset by using the GUI

1. Log on to the cluster IP address.
2. Navigate to **System > Network > Linksets**.
3. In the details pane, click **Add**.
4. In the **Create Linkset** dialog box:
   - Specify the name of the linkset by setting the Linkset parameter.
   - Specify the Interfaces to be added to the linkset and click Add. Repeat this step for each interface you want to add to the linkset.
5. Click **Create**, and then click **Close**.


```

# Node groups for spotted and partially striped configurations

September 14, 2021

By virtue of the default cluster behavior, all configurations performed on the cluster IP address are available on all nodes of the cluster. However, there might be cases where you need some configurations to be available only on specific cluster nodes.

You can achieve this requirement by defining a node group that includes the specific cluster nodes, and then binding the configuration to that node group. It ensures that the configuration is active only on those cluster nodes. These configurations are called partially striped or spotted (if active only one a single node). For more information, see Striped, Partially Striped, and Spotted Configurations.

For example, consider a cluster with three nodes. You create a node group NG0 that includes node n1 and another node group NG1 that includes n2 and n3. Bind load balancing virtual servers 0.77 to NG0 and load balancing virtual server 0.69 to NG1.

It means that virtual server 0.77 is active only on n1 and therefore only n1 receives the traffic that is directed to 0.77. Similarly, virtual server 0.69 is active only on nodes n2 and n3 and therefore only n2 and n3 receives traffic that is directed to 0.69.

Figure 1. Citrix ADC cluster with node groups configured for spotted and partial-striped configurations




The entities or configurations that you can bind to a node group are:

- Load balancing, content switching, cache redirection, authentication, authorization, and auditing virtual servers

  > **Note**
  >
  > FTP load balancing virtual servers cannot be bound to node groups.

- VPN virtual server (Supported from NetScaler 10.5 Build 50.10 onwards)
- Global Server Load Balancing (GSLB) sites and other GSLB entities (Supported from NetScaler 10.5 Build 52.11 onwards)
- Limit identifiers and stream identifiers

 

# Behavior of node groups

September 14, 2021

Due to the interoperability of node groups with different Citrix ADC features and entities, there are some behavioral aspects to be noted. Nodes in a node group can also be backed up. Read on for more information.

## General behavior of a cluster node group

- A node group that has entities bound to it cannot be removed.
- A cluster node that belongs to a node group with entities bound to it, cannot be removed.
- A cluster instance that has node groups with entities bound to it, cannot be removed.
- You cannot add an entity that has a dependency on another entity. It must not be a part of the node group. If you must do so, first remove the dependency. Then, add both the entities to the node group and reassociate the entities.

  **Examples:**

  - Assume you have a virtual server, VS1, whose backup is virtual server VS2. To add VS1 to a node group, first make sure that VS2 is removed as the backup server of VS1. Then, bind each server individually to the node group, and then configure VS2 as the backup for VS1.
  - Assume you have a content switching virtual server, CSVS1, whose target load balancing virtual server is LBVS1. To add CSVS1 to a node group, first remove LBVS1 as the target. Then, bind each server individually to the node group, and then configure LBVS1 as the target.
  - Assume you have a load balancing virtual server, LBVS1, that has a policy which invokes another load balancing virtual server, LBVS2. To add either one of the virtual servers, first remove the association. Then, bind each server individually to the node group, and then reassociate the virtual servers.
- You cannot bind an entity to a node group. It has no nodes and that has the strict option enabled. Therefore, you cannot unbind the last node of a node group that has entities bound to it and that has the strict option enabled.
- The strict option cannot be modified for a node group that has no nodes but has entities bound to it.

## Backing up nodes in a node group

By default, a node group is designed to provide back up nodes for members of a node group. If a node group member goes down, a cluster node that is not a member of the node group dynamically

 

replaces the failed node. This node is called the replacement node.

> **Note**
>
> For a single-member node group, a backup node is automatically preselected when an entity is bound to the node group.

When the original member of the node group comes up, the replacement node, by default, is replaced by the original member node.

From NetScaler 10.5 Build 50.10 onwards, however, the Citrix ADC allows you to change this replacement behavior. When you enable the sticky option, the replacement node is retained even after the original member node comes up. The original node takes over only when the replacement node goes down.

You can also disable the backup functionality. To do it, you must enable the strict option. In this scenario, when a node group member goes down, no other cluster node is picked up as a backup node. The original node continues being part of the node group when it comes up. This option ensures that entities bound to a node group are active only on node group members.

> **Note**
>
> The strict and sticky option can be set only when creating a node group.

## Configuring node groups for spotted and partially striped configurations

September 14, 2021

To configure a node group for spotted and partially striped configurations you must first create a node group and then bind the required nodes to the node group. You then associate the required entities to that node group. The entities that are bound to the node group are of the following:

- **Spotted** - If bound to a node group that has a single node.
- **Partially Striped** - If bound to a node group that has more than one node.

**Some points to remember:**

- GSLB is supported on a cluster only when GSLB sites are bound to node groups that have a single cluster node. For more information, see Setting Up GSLB in a Cluster.
- Citrix Gateway is supported on a cluster only when the VPN virtual servers are bound to node groups that have a single cluster node. The sticky option must be enabled on the node group.
- For versions prior to NetScaler 11, the application firewall is supported only on individual cluster nodes (spotted configuration). Application firewall profiles can be associated only with virtual

 

servers that are bound to node groups that have a single cluster node. It means that application you are not allowed to do the following:
  - Bind application firewall profiles to striped or partially striped virtual servers.
  - Bind the policy to a global bind point or to user-defined policy labels.
  - Unbind, from a node group, a virtual server that has application firewall profiles.
- NetScaler 11 introduced application firewall support for striped and partially striped configurations. For more information, see Application Firewall Support for Cluster Configurations.

Check Citrix ADC Features Supported in a Cluster to see the NetScaler versions from which GSLB, Citrix Gateway, and application firewall are supported in a cluster.

### To configure a node group by using the command line interface

1. Log on to the cluster IP address.
2. Create a node group. Type:

   ```
   add cluster nodegroup <name> -strict (YES | NO)<!--NeedCopy-->
   ```

   Example

   ```
   1 add cluster nodegroup NG0 -strict YES
   ```

3. Bind the required nodes to the node group. Type the following command for each member of the node group:

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

   Example

   To bind nodes with IDs 1, 5, and 6.

   ```
   1 > bind cluster nodegroup NG0 -node 1
   2 > bind cluster nodegroup NG0 -node 5
   3 > bind cluster nodegroup NG0 -node 6
   ```

4. Bind the entity to the node group. Type the following command once for every entity that you want to bind:

   ```
   bind cluster nodegroup <name> (-vServer <string> | -identifierName <
   string> | -gslbSite <string> -service <string>)<!--NeedCopy-->
   ```

   > **Note**
   >
   > The gslbSite and service parameters are available from NetScaler 10.5 onwards.

   Example

   To bind virtual servers VS1 and VS2 and rate limit identifier named identifier1.


```
1 > bind cluster nodegroup NG0 -vServer VS1
2 > bind cluster nodegroup NG0 -vServer VS2
3 > bind cluster nodegroup NG0 -identifierName identifier1
```

5. Verify the configurations by viewing the details of the node group. Type:

   ```
   show cluster nodegroup <name><!--NeedCopy-->
   ```

   Example

   ```
   1 > show cluster nodegroup NG0
   ```

To configure a node group by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Node Groups.**
3. In the details pane, click **Add.**
4. In the **Create Node Group** dialog box, configure the node group:
   a) Under **Cluster Nodes**, click the **Add** button.
      - The Available list displays the nodes that you can bind to the node group and the Configured list displays the nodes that are bound to the node group.
      - Click the **+** sign in the Available list to bind the node. Similarly, click the **-** sign in the Configured list to unbind the node.
   b) Under **Virtual Servers**, select the tab corresponding to the type of virtual server that you want to bind to the node group. Click the **Add** button.
      - The Available list displays the virtual servers that you can bind to the node group and the Configured list displays the virtual servers that are bound to the node group.
      - Click the **+** sign in the Available list to bind the virtual server. Similarly, click the - sign in the Configured list to unbind the virtual server.

Configuring redundancy for node groups

September 14, 2021

> **Note**
>
> Supported from NetScaler 10.5 Build 52.1115.e onwards.

Node groups can be configured such that when one node group goes down, another node group can take over and process traffic. For example, when a node group NG1 goes down, NG2 takes over.

> **Note**
>
> This functionality can be used to configure data center redundancy where each node group is configured as a data center.

To achieve this use case, cluster nodes must be logically grouped into node groups, where some node groups must be configured as ACTIVE and others as SPARE. The active node group with the highest priority (that is, the lowest priority number) is made operationally active and therefore serves traffic. When a node from this operationally active node group goes down, the node count of this node group is compared with the node count of the other active node groups in order of their priority. If a node group has a higher or equal node count, that node group is made operationally active. Else, the spare node groups are checked.

> **Note**
>
> - Only one state-specific node group can be active at a given point in time.
> - A cluster node inherits the state of the node group. So, if a node with "SPARE" state is added to the node group with state as "ACTIVE", the node automatically behaves as an active node.
> - The pre-emption parameter that is defined for the cluster instance decides whether the initial active node group takes the control when it comes up again.
> - A spare node group can take up a node group and host active traffic when an active node group goes down.

The following figure shows a node group setup that has node group redundancy defined. NG1 is initially the active node group. When it loses one of the nodes, the spare node group (NG3) with the highest priority starts serving traffic.

Figure 1. Citrix ADC cluster with node group redundancy configured.



Configuring redundancy for node groups

1. Log on to the cluster IP address.
2. Create the active node group and bind the required cluster nodes.

```
1 > add cluster nodegroup NG1 -state ACTIVE
2 > bind cluster nodegroup NG1 -node n1
3 > bind cluster nodegroup NG1 -node n2
4 > bind cluster nodegroup NG1 -node n3
```

3. Create the spare node group and bind the requisite nodes.

```
1 > add cluster nodegroup NG2 -state SPARE -priority 20
2 > bind cluster nodegroup NG2 -node n4
3 > bind cluster nodegroup NG2 -node n5
4 > bind cluster nodegroup NG2 -node n6
```

4. Create another spare node group and bind the requisite nodes.

```
1 > add cluster nodegroup NG3 -state SPARE -priority 10
2 > bind cluster nodegroup NG3 -node n7
```



```
3 > bind cluster nodegroup NG3 -node n8
4 > bind cluster nodegroup NG3 -node n9
```

Disabling steering on the cluster backplane

September 14, 2021

> **Note**
>
> Supported from NetScaler 11 onwards.

The default behavior of a Citrix ADC cluster is to direct the traffic that it receives (flow receiver) to another node (flow processor). The flow processor must then process the traffic. This process of directing the traffic from flow receiver to flow processor occurs over the cluster backplane and is called steering.

If necessary, you can disable steering so that the process becomes local to the flow receiver and therefore makes the flow receiver as the flow processor. Such a configuration setup can come handy when you have a high latency link.

> **Note**
>
> This configuration is applicable only for striped virtual servers.

- For partially striped virtual servers, if the flow receiver is a non-owner node, the traffic is steered to an owner node. If however, the flow receiver is an owner node, then steering is disabled.
- For spotted virtual servers, the flow receiver is the flow processor, and hence there is no need for steering.

Some points to remember when disabling the steering mechanism:

- Striped SNIPs are not supported as steering is disabled.
- MPTCP and FTP do not work.
- L2 mode must be disabled.
- If USIP is enabled, traffic might not reach back to the same node as the steering is disabled.
- Traffic that is directed to the cluster IP address is steered to the configuration coordinator.
- When a node joins or leaves a cluster, it is possible that more than 1/N connections are affected. It is because a change in the nodes available, might cause the routes to be rehashed. As a result, the traffic is routed to another node and due to the non-availability of steering, the traffic is not processed.

Steering can be disabled at the individual virtual server level or at the global level. The global configuration takes precedence over the virtual server setting.

- Disabling backplane steering for all striped virtual servers

  Configured at cluster instance level. Traffic meant for any striped virtual server is not steered on the cluster backplane.

  ```
  add cluster instance \<clId\> -processLocal ENABLED<!--NeedCopy-->
  ```

- Disabling backplane steering for a specific striped virtual server

  Configured on a striped virtual server. Traffic meant for the virtual server is not steered on the cluster backplane.

  ```
  add lb vserver <name> <serviceType> -processLocal ENABLED<!--NeedCopy
  -->
  ```

Synchronizing cluster configurations

September 14, 2021

Citrix ADC configurations that are available on the configuration coordinator are synchronized to the other nodes of the cluster when:

- A node joins the cluster
- A node rejoins the cluster
- A new command is run through the cluster IP address

Also, you can forcefully synchronize the configurations that are available on the configuration coordinator (full synchronization) to a specific cluster node. Make sure you synchronize one cluster node at a time, otherwise the cluster can get affected.

To synchronize cluster configurations by using the CLI

At the command prompt of the appliance on which you want to synchronize the configurations, type:

```
1 force cluster sync
```

To synchronize cluster configurations by using the GUI

1. Log on to the appliance on which you want to synchronize the configurations.
2. Navigate to **System > Cluster.**
3. In the details pane, under **Utilities**, click Force cluster sync.
4. Click **OK.**

Synchronizing time across cluster nodes

September 14, 2021

The cluster uses a Precision Time Protocol (PTP) to synchronize the time across cluster nodes. PTP uses multicast packets to synchronize the time. If there are some issues in time synchronization, you must disable PTP and configure Network Time Protocol (NTP) on the cluster.

To enable/disable PTP by using the command line interface

At the command prompt of the cluster IP address, type:

```
set ptp -state disable
```

To enable/disable PTP by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**
3. In the details pane, under **Utilities**, click **Configure PTP Settings.**
4. In the **Enable/Disable PTP** dialog box, select whether you want to enable or disable PTP.
5. Click **OK.**

Synchronizing cluster files

September 14, 2021

The files available on the configuration coordinator are called cluster files. These files are automatically synchronized on the other cluster nodes when the node is added to the cluster and periodically, during the lifetime of the cluster. Also, you can manually synchronize the cluster files.

Important: Removing any certificate or key files in a cluster environment restricts further configuration on the ADC appliance. Add the files back at the same location to make any configuration changes.

The directories and files from the configuration coordinator that are synchronized are:

- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/bookmark/
- /nsconfig/dns/
- /nsconfig/htmlinjection/
- /netscaler/htmlinjection/ens/

- /nsconfig/monitors/
- /nsconfig/nstemplates/
- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/likewise/db/
- /var/download/
- /var/wi/tomcat/webapps/
- /var/wi/tomcat/conf/Catalina/localhost/
- /var/wi/java_home/lib/security/cacerts
- /var/wi/java_home/jre/lib/security/cacerts
- /nsconfig/license/
- /nsconfig/rc.conf

> **Tip**
>
> Files (certificates and key files) that are copied to the cluster configuration coordinator manually (or through the shell) are not automatically available on the other cluster nodes. Run the "sync cluster files" command from the cluster IP address before running a command that depends on these files.

To synchronize cluster files by using the command line interface

At the command prompt of the cluster IP address, type:

```
sync cluster files <mode>
```

To synchronize cluster files by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**

3. In the details pane, under **Utilities**, click Synchronize cluster files.
4. In the **Synchronize** cluster files dialog box, select the files to be synchronized in the Mode drop-down list.
5. Click **OK.**

Viewing the statistics of a cluster

September 14, 2021

You can view the statistics of a cluster instance and cluster nodes to evaluate the performance or to troubleshoot the operation of the cluster.

To view the statistics of a cluster instance by using the command line interface

At the command prompt of the cluster IP address, type:

```
stat cluster instance <clId>
```

To view the statistics of a cluster node by using the command line interface

At the command prompt of the cluster IP address, type:

```
stat cluster node <nodeid>
```

> **Note**
>
> The `stat cluster node <nodeid>` command displays the cluster level statistics when you run the command from the cluster IP address. However, when you run from the NSIP address of a cluster node, the command displays node level statistics.

To view the statistics of a cluster instance by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**
3. In the details pane, in the center of the page, click **Statistics.**

To view the statistics of a cluster node by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**

3. In the details pane, select a node and click **Statistics** to view the statistics of the node. To view the statistics of all the nodes, click **Statistics** without selecting a specific node.

Discovering Citrix ADC appliances

September 14, 2021

You can discover the appliances present in the same subnet as the current node. The required discovered appliances can then be selectively added to the cluster. This operation can be performed to either create a cluster or to add nodes to an existing cluster.

> **Note**
>
> - The discover operation can be performed only through the configuration utility.
> - This operation cannot discover Citrix ADC appliances from different networks.
> - When performing this operation to add nodes to an existing cluster, the L3 VLAN configurations are cleared from the node. You make sure to define these configurations once the appliance is added to the cluster.

To discover appliances by using the GUI

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**
3. In the details pane, at the bottom of the page, click **Discover NetScalers.**
4. In the **Discover NetScalers** dialog box, set the following parameters:
   - **IP address range** - Specify the range of IP addresses within which you want to discover appliances. For example, you can search for all NSIP addresses between 10.102.29.4 to 10.102.29.15 by specifying this option as 10.102.29.4 - 15.
   - **Backplane interface** - Specify the interfaces to be used as the backplane interface. It is an optional parameter. If you do not specify this parameter, you must update it after the node is added to the cluster.
5. Click **OK.**
6. Select the appliances that you want to add to the cluster.
7. Click **OK.**

Disabling a cluster node

September 14, 2021

You can temporarily remove a node from a cluster by disabling the cluster instance on that node. A disabled node is not synchronized with the cluster configurations. When the node is enabled again, the cluster configurations are automatically synchronized on it. For more information, see Synchronization across cluster nodes.

A disabled node cannot serve traffic and all existing connections on this node are terminated.

> **Note**
>
> If the configurations of a disabled non-configuration coordinator node are modified (through the NSIP address of the node), the configurations are not automatically synchronized on that node. You can manually synchronize the configurations as described in Synchronizing Cluster Configurations.

To disable a cluster node by using the command line interface

At the command prompt of the node that you want to disable, type:

```
disable cluster instance <clId>
```

> **Note**
>
> To disable the cluster, run the
> disable cluster instance command on the cluster IP address.

To disable a cluster node by using the configuration utility

1. On the node that you want to disable, navigate to **System > Cluster**, and click **Manage Cluster.**
2. In the **Configure** cluster instance dialog box, unselect the **Enable** cluster instance check box.

   > **Note**
   >
   > To disable the cluster instance on all the nodes, perform the preceding procedure on the cluster IP address.

Removing a cluster node

September 14, 2021

When a node is removed from the cluster, the cluster configurations are cleared from the node (by internally running the clear ns config -extended command). The SNIP addresses, **MTU** settings of the backplane interface, and all VLAN configurations (except the default VLAN and NSVLAN) are also cleared from the appliance.

> **Note**
>
> - If the deleted node was the cluster configuration coordinator (CCO), another node is automatically selected as the CCO, and the cluster IP address is assigned to that node. All the current cluster IP address sessions are invalid and you have to start a new session.
> - To delete the whole cluster, you must remove each node individually. When you remove the last node, the cluster IP addresses are deleted.
> - When an active node is removed, the traffic serving capability of the cluster is reduced by one node. Existing connections on this node are terminated.

To remove a cluster node by using the CLI

For NetScaler 10.1 and later versions

1. Log on to the cluster IP address and at the command prompt, type:

```
rm cluster node <nodeId>

save ns config
```

2. Log on to the removed node, NSIP address, and at the command prompt, type:

```
save ns config
```

> **Note**
>
> If the cluster IP address is unreachable from the node, run the rm cluster instance command on the NSIP address of that node itself.

For NetScaler 10

1. Log on to the node that you want to remove from the cluster and remove the reference to the cluster instance.

```
rm cluster instance <clId>

save ns config
```

2. Log on to the cluster IP address and remove the node from which you removed the cluster instance.

```
rm cluster node <nodeId>

save ns config
```


Make sure you do not run the `rm cluster node` command from the local node. It results in inconsistent configurations between the CCO and the node.

To remove a cluster node by using the GUI

On the cluster IP address, navigate to **System > Cluster > Nodes**, select the node you want to remove and click **Remove.**

Removing a node from a cluster deployed using cluster link aggregation

September 14, 2021

To remove a node from a cluster that uses cluster link aggregation as the traffic distribution mechanism, you must make sure that the node is made passive so that it does not receive any traffic and then, on the upstream switch, remove the corresponding interface from the channel.

For detailed information on cluster link aggregation, see Using Cluster Link Aggregation.

To remove a node from a cluster that uses cluster link aggregation as the traffic distribution mechanism

1. Log on to the cluster IP address.
2. Set the state of the cluster node that you want to remove to PASSIVE.

```
set cluster node <nodeId> -state PASSIVE
```

3. On the upstream switch, remove the corresponding interface from the channel by using switch-specific commands.

> **Note**
>
> You do not have to manually remove the nodes interface on the cluster link aggregation channel. It is automatically removed when the node is deleted in the next step.

4. Remove the node from the cluster.

```
rm cluster node <nodeId>
```

Detecting jumbo probe on a cluster

September 14, 2021

If a Jumbo frame is enabled on a cluster interface, the backplane interface must be large enough to support the all packets in the Jumbo frame. It is achieved by setting the Maximum Transmission Unit (MTU) of the backplane as:

Backplane_MTU = maximum (all cluster interface MTUs) + 78

To verify the preceding configuration, you must send a jumbo probe (of the preceding computational size) to all the peer nodes of a Cluster setup. If the probe does not succeed, the appliance displays a warning message in the output of the “show cluster instance” command.

In the command interface mode, type the following command:

```
> show cluster instance
  Cluster ID: 1
  Dead Interval: 3 secs
  Hello Interval: 200 msecs
  Preemption: DISABLED
  Propagation: ENABLED
  Quorum Type: MAJORITY
  INC State: DISABLED
  Process Local: DISABLED
  Cluster Status: ENABLED(admin),    ENABLED(operational), UP
```

Warning

The MTU for a backplane interface must be large enough to handle all packets in the frame. It must be equal to <MTU_VAL>. If the recommended value is not user configurable, you must review the MTU value of jumbo interfaces.

Sl. no	Member Nodes	Health	Admin State	Operation State
1	Node ID: 1; Node IP: 10.102.53.167	UP	Active	ACTIVE (Configuration Coordinator)
2	Node ID: 2; Node IP: 10.102.53.168	UP	Active	Active

Route monitoring for dynamic routes in cluster

September 14, 2021

You can use a route monitor to make a cluster node dependent on the internal routing table whether

it contains or does not contain a dynamically learned route. A route monitor on each node checks the internal routing table to ensure there is a route entry for reaching a particular network is always present. If the route entry is not present, the state of the route monitor changes to DOWN.

In a cluster deployment, if the client-side or server side-link of a node goes down, traffic is steered to this node through the peer nodes for processing. The steering of traffic is implemented by configuring dynamic routing and adding static ARP entries, pointing to the special MAC address of each node, on all the nodes. If there are many nodes in a cluster deployment, adding and managing static ARP entries with special MAC addresses on all the nodes is a cumbersome task. Now, nodes implicitly use special MAC addresses for steering packets. Therefore, static ARP entries pointing to special MAC addresses are no longer required to be added to the cluster nodes.

To bind a cluster node using the CLI

At the command prompt, type:

```
bind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
unbind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
```

Consider a scenario where Node 1 is bound to route monitor 1.1.1.0 255.255.255.0. When a dynamic route fails, Node 1 become INACTIVE. The health status is available in the `show cluster node` command by node id as per following.

```
Node ID: 1
IP:  10.102.169.96
Backplane:  1/1/2
Health: NOT UP
Reason(s): Route Monitor(s) of the node have failed
Route Monitor -  Network: 1.1.1.0   Netmask: 255.255.255.0   State:
    DOWN
```

Monitoring cluster setup using SNMP MIB with SNMP link

September 14, 2021

SNMP MIB is device specific information that is configured on the SNMP agent for identifying a Citrix ADC appliance. It can identify information such as, appliance name, administrator, and location. In a cluster setup, you can now configure the SNMP MIB in any node by including the “ownerNode” pa-

rameter in the set SNMP MIB command. Without this parameter, the set SNMP MIB command applies only to the Cluster Coordinator (CCO) node.

To display the MIB configuration for a cluster node other than the CCO, include the “ownerNode” parameter in the show SNMP MIB command.

Configuring SNMP MIB on CLIP

To configure and view MIB configuration on CLIP by using the command line interface.

```
set snmp mib [-contact <string>] [-name <string>] [-location <string>]
      [-customID <string>] [-ownerNode <positive_integer>]
Done
show snmp mib [-ownerNode <positive_integer>]

 > set mib -contact John -name NS59 -location San Jose -customID 123 -
    ownerNode 3
Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun  7
           2016, 10:27:29
        sysUpTime:   124300
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  John
        sysName:     NS59
        sysLocation: San Jose
        sysServices: 72
        Custom ID: 123
 Done

> unset mib -contact -name -location -customID -ownerNode 3
 Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun 7
           2016, 10:27:29
        sysUpTime:   146023
        sysObjectID: .1.3.6.1.4.1.5951.1.1
```



```
        sysContact:  WebMaster (default)
        sysName:     NetScaler
        sysLocation: POP (default)
        sysServices: 72
        Custom ID: Default
Done
```

Cluster SNMP trap messages

In cluster setup, the SNMP trap alarm configurations must be done from the CLIP. The commands are propagated to each of the nodes.

For more information on configuring SNMP, see Configuring the Citrix ADC to generate SNMP traps.

The following are the cluster specific traps that are available:

```
>sh snmp alarm | grep cluster
    CLUSTER-BACKPLANE-HB-MISSING N/A N/A 86400   ENABLED - ENABLED
    CLUSTER-CCO-CHANGE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-NODE-HEALTH N/A N/A 86400 ENABLED - ENABLED
    CLUSTER-NODE-QUORUM N/A N/A 86400 ENABLED - ENABLED
    CLUSTER-OVS-CHANGE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-PROP-FAILURE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-SYNC-FAILURE N/A N/A N/A ENABLED - ENABLED
    CLUSTER-SYNC-PARTIAL-SUCCESS N/A N/A N/A ENABLED - ENABLED
    CLUSTER-VERSION-MISMATCH N/A N/A 86400 ENABLED - ENABLED
```

Monitoring command propagation failures in a cluster deployment

September 14, 2021

In a cluster deployment, you can use the new command “show prop status” for faster monitoring and troubleshooting of issues. The issues related to command-propagation failure on non-CCO nodes. This command displays up to 20 of the most recent command propagation failures on all non-CCO nodes. You can use either the Citrix ADC appliance CLI or GUI to perform this operation. You can access them through the CLIP address or through the NSIP address of any node in the cluster deployment.

Graceful shutdown of nodes

September 14, 2021

In a cluster setup, some of the existing connections (1/Nth connections, where N is the cluster size) at the cluster level or specific virtual server level are lost. This behavior is observed if a node leaves or joins the system. To address the loss, you must gracefully handle the existing connections. Graceful handling is done by configuring the “retain connections on cluster” option in the CLIP address and specifying a timeout interval in the node’s NSIP.

Graceful handling of connections is applicable in two scenarios:

1. Cluster upgrade
2. New node addition

Graceful handling of Nodes in cluster upgrade

To upgrade a cluster, you must upgrade one node at a time. Before upgrading a node, you must set it to passive state and then set it to active state after the upgrade. To avoid terminating existing connections when upgrading the node, shut it down gracefully with a configured timeout interval. Otherwise, 1/Nth (where N is the cluster size) of the cluster’s connections are terminated.

> **Note**
>
> If existing sessions are not completed within the configured timeout interval, they get terminated after the grace time.

Following are the steps to gracefully handle nodes in a cluster upgrade scenario:

1. Consider a cluster setup of five nodes (n0, n1, n2, n3, n4).
2. Before you shut down a node, you must configure the “retainConnectionsOnCluster” option. It helps to retain all existing connections of this node at the cluster level or virtual server level for a specific time interval.

   Example

   On CLIP

   “‘set cluster instance –retainConnectionsOnCluster YES

   ```
   OR

   ```set lb vserver <vserver name>  - retainConnectionsOnCluster Yes
      <!--NeedCopy-->
   ```

3. Now, log on to the NSIP address of node n3 and set the node n3 to PASSIVE with a timeout internal.

   Example

   “‘set cluster node n3 –state PASSIVE –delay 60


   ```

```
1 ```saveconfig<!--NeedCopy-->
```

4. After the grace period expires, close all connections, shut down n3 and reboot the Citrix ADC appliance.
5. Upgrade the appliance. Then, with the CLI connected to the appliance's NSIP address, set the node to ACTIVE.

   Example

   "'set cluster node n3 –state ACTIVE

```
1 ```saveconfig<!--NeedCopy-->
```

6. Repeat steps 4–6 for all nodes in the cluster.
7. After all nodes are upgraded and set to ACTIVE, reset the retainConnectionsOnCluster option from the CLIP address.

   Example

   "'set cluster instance -retainConnectionsOnCluster NO

```
1 OR
2
3 ```set lb vserver <vserver name>  - retainConnectionsOnCluster NO
     <!--NeedCopy-->
```

> **Note**
>
> If there is a version mismatch when upgrading a cluster, cluster propagation is automatically disabled and no commands are allowed on the CLIP.

Graceful handling of nodes during a new node addition

The graceful handling of nodes describes how a new node can be added to the existing Citrix ADC cluster. Consider you have a Citrix ADC cluster that is already serving traffic. And you want to add an extra appliance as a node to the cluster without terminating its existing connections. To accomplish the preceding scenario, set the option to retain existing connections either at a Global level or at a specific virtual server level. Once done, save the configuration. Now set the option to retain connections to NO, to allow existing connections from other nodes to be reassigned to the new node.

Following are the steps to gracefully handle nodes if a node newly added:

1. You save the existing configuration that has the “retainConnectionsOnCluster” option enabled. By doing so, you can retain all existing connections of this node at the cluster level or virtual server level for a specific time interval.

   On CLIP

   ```
   1 set cluster instance x - retainConnectionsOnCluster YES
   ```

   OR

   ```
   1 set lb vserver xxxx -retainConnectionsOnCluster Yes
   ```

2. Add a node ‘n5’ to the cluster setup.
3. Disable “the retainConnectionOnCluster” option to “NO” for distributing existing connections from other nodes to the newly added node n5.

   On CLIP

   ```
   1 set cluster instance x - retainConnectionsOnCluster NO
   ```

   OR

   ```
   1 set lb vserver xxxx -retainConnectionsOnCluster NO
   ```

> **Note**
>
> The backplane steering depends on the type of traffic distribution mechanism (ECMP, CLAG, and USIP) on a cluster setup. The increase in backplane steering is based on the traffic type.

Configuring graceful shutdown of nodes in a cluster

To configure graceful shutdown of nodes in a cluster, do the following:

1. Configure the “retainConnectionsonCluster” option at Global (cluster) level.
2. Configure the “retainConnectionsonCluster” option at the virtual server level.
3. Set the node (leaving the system) to the passive state with a graceful timeout interval specified in the node’s NSIP address.
4. Monitor the existing connections to make sure all transactions are completed within the grace period.

To retain existing connections at the global (cluster) level by using the CLI

You can retain existing connections either at a global level or at a specific virtual server level. This option is configured to retain all existing connections at the global level. By default, this option is disabled.

At the command prompt type:

```
- set cluster instance <clusterID> -retainConnectionsOnCluster YES

- set cluster instance 60 - retainConnectionsOnCluster YES
```

To retain existing connections of a specific virtual server in the cluster by using the CLI

This option is configured to retain existing connections specific to a load balancing virtual server. To retain those connections, we enable this option at the virtual server level. By default, this option is disabled.

At the command prompt, type:

```
- set lb vserver <clusterID> -retainConnectionsOnCluster Yes

- set lb vserver v1 -retainConnectionsOnCluster Yes
```

To set a cluster node to passive state by using the CLI

To set a cluster node to passive state with a graceful timeout interval. This setting is performed in the node's NSIP as propagation is disabled during cluster upgrade.

At the command prompt, type:

```
- set cluster node <clusterID> -state passive
-backplane <interface_name>@
-priority <positive_integer>
-delay <mins>

- set cluster node 4 -state PASSIVE -delay 60

- set cluster instance 60 - retainConnectionsOnCluster YES
- set lb vserver v1 -retainConnectionsOnCluster Yes
- set cluster node 4 -state PASSIVE -delay 60
```

> **Note**
>
> You might observe the following behavior on a cluster node when it is set to passive with a delay option configured from a CLIP:
>
> - After the timeout, the node shows as passive from the NSIP of the node.

> - The **show cluster instance** command on CLIP displays the node as active from the CLIP. Whereas the **show cluster node** command on the CLIP displays the node as passive.

To configure graceful shutdown of nodes by using the GUI

1. Navigate to **Configuration > System > Cluster** and click **Manage Cluster.**
2. On the **Manage Cluster** page, select **Retain Connections on Cluster** option.
3. Click **OK**, and then click **Done**.

Graceful shutdown of services

September 14, 2021

Starting with NetScaler 12.1 build 49.xx, Citrix ADC clusters support graceful shutdown of services. To gracefully shut down the services, you can perform one of the following tasks.

- Explicitly disable the service, and
  - Set a delay (in seconds).
  - Enable graceful shutdown.
- Add a TROFS code or string to the monitor.

For more details, see Graceful shutdown of services.

To configure graceful shutdown for a service by using the CLI

Disable with graceful option only:

At the command prompt, type:

```
disable service <name> [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

Example

```
disable service svc1 -graceFul YES
Done
sh service svc1
                svc1 (10.102.225.11:80) - HTTP
                State: GOING OUT OF SERVICE   Graceful (number of
                   active clients: 1)
```



```
6                   Last state change was at Wed Jul 25 10:46:29 2018
7                   Time since last state change: 0 days, 00:00:02.680
8                   ... ... ... .
9                   ... ... ... .
10                  Traffic Domain: 0
11
12 1)            Monitor Name: tcp-default
13                                  State: UP              Weight: 1
                                                        Passive: 0
14                                  Probes: 26             Failed [Total: 0
                                        Current: 0]
15                                  Last response: Success - TCP syn+ack
                                        received.
16                                  Response Time: 0.0 millisec
17 <!--NeedCopy-->
```

Disable with timeout and graceful option:

At the command prompt, type:

```
1 disable service <name> [<delay>] [-graceFul (YES|NO)]
2
3 show service <name>
4 <!--NeedCopy-->
```

Example

```
1 disable service svc1 2000 -graceFul YES
2
3 Done
4 > sh service svc1
5                   svc1 (10.102.225.11:80) - HTTP
6                   State: GOING OUT OF SERVICE (Graceful (number of active
                        clients: 1), Out Of Service in 1998 seconds)
7                   Last state change was at Wed Jul 25 10:49:08 2018
8                   Time since last state change: 0 days, 00:00:01.710
9                   ... ... ... .
10                  ... ... ... .
11                  Traffic Domain: 0
12
13 1)            Monitor Name: tcp-default
14                                  State: UP              Weight: 1
                                                        Passive: 0
15                                  Probes: 57             Failed [Total: 0
                                        Current: 0]
```



```
16                                  Last response: Success - TCP syn+ack
                                        received.
17                                  Response Time: 0.0 millisec
18 Done
19 <!--NeedCopy-->
```

Disable service group with timeout and graceful option:

At the command prompt, type:

```
1 disable serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-delay
2 <secs>] [-graceFul ( YES | NO )]
3 Show service group <serviceGroupName>
4 <!--NeedCopy-->
```

Example:

```
1  disable servicegroup sg -delay 2000 -graceFul yes
2  sh servicegroup sg
3                   sg - HTTP
4                   State: DISABLED                Effective State: OUT OF
                        SERVICE Monitor Threshold : 0
5                   Max Conn: 0       Max Req: 0          Max Bandwidth: 0
                        kbits
6                   Use Source IP: NO
7                   Client Keepalive(CKA): NO
8                    … … … .
9                    … … …
10
11
12                  1)   200.200.10.21:80       Server Name: server3
                        Server ID: None Weight: 1
13                                  State:    GOING OUT OF SERVICE (learnt
                                        from node:2 )     Graceful (number
                                        of active clients: 6), Out Of
                                        Service in 1993 seconds
14                                  Last state change was at Mon Aug 13
                                        15:15:11 2018
15                                   … … …
16
17                  2)   200.200.10.22:80       Server Name: server4
                        Server ID: None Weight: 1
18                                  State:    GOING OUT OF SERVICE (learnt
                                        from node:2 )     Graceful (number
                                        of active clients: 7), Out Of
                                        Service in 1993 seconds
```



```
                                    Last state change was at Mon Aug 13
                                       15:15:11 2018
<!--NeedCopy-->
```

> **Note**
> CLIP displays the aggregated value of all active clients connections from all cluster nodes.

To configure graceful shutdown for a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Open the service, and from the Action list, click **Disable**. Enter a wait time, and select Graceful.

To configure a TROFS code or string in a monitor by using the CLI

At the command prompt, type one of the following commands:

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>
add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>
add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
<!--NeedCopy-->
```

To configure a TROFS code or string in a monitor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. On the Monitors pane, click Add, and do one of the following steps:
   - Select Type as HTTP, and specify a TROFS Code.
   - Select Type as HTTP-ECV or TCP-ECV, and specify a TROFS String.

IPv6 ready logo support for clusters

September 14, 2021

You can test clustered appliances for IPv6 Ready Logo certification. Modified commands for testing IPv6 core protocols, such as for ND test cases, Router Solicitation processing, and sending route advertisement and router redirection messages are available in a clustered setup. Following are the IPv6 functionalities available for testing the IPv6 core protocols.

Following are the modified functionalities available to pass IPv6 core protocols, such as ND test cases, Router Solicitation processing and sending Route advertisement and router redirection messaging in the IPv6readylogo phase2 test suite.

- Link Local SNIPs
- Address Resolution and Neighbor Unreachability
- Router and Prefix Discovery
- Router Redirection
- DoDAD

With these modified commands, the following configurations are supported in a clustered appliance.

Supportable configurations for testing IPv6 core protocols

For a clustered setup to pass IPv6 Ready Logo test cases, you can run the following configurations on the Cluster Management IP address (CLIP).

- global IP6 configuration
- basic IPv6 configuration
- more IPv6 configurations

Global configuration

A global IPv6 configuration enables you to set the global IPv6 parameters (such as relearning, routeRedirection, ndBasereachTime, nRetransmissionTime, `natprefix`, td, and doodad) to run the basic IPv6 configuration.

At the command prompt, type the following:

```
set ipv6 [-ralearning ( ENABLED | DISABLED )] [-routerRedirection (
    ENABLED | DISABLED )] [-ndBasereachTime<positive_integer>][-
    ndRetransmissionTime <positive_integer>] [-natprefix <ipv6_addr|*>[-
    td<positive_integer>]] [-doDAD ( ENABLED | DISABLED )]
```

Basic IPv6 configuration

The basic IPv6 configuration enables you to create an IPv6 address and bind to a VLAN interface. You can perform the following configurations to test the IPv6 core protocols.

To add a VLAN to the clustered setup by using the CLI

At the command prompt, type:

```
add vlan <id>
```

To add another VLAN to the clustered setup by using the CLI

At the command prompt, type:


```
add vlan <id>
```

To bind an interface to a VLAN by using the CLI

At the command prompt, type:

```
bind vlan <id> -ifnum <interface_name>
```

To bind an interface to a VLAN by using the CLI

This command adds the global prefix as on-link prefix into RA information for subsequent Router Advertisements. At the command prompt, type:

```
bind vlan <id> -ifnum <interface_name>
```

To add the IPv6 SNIP address on a VLAN by using the CLI

At the command prompt, type the following:

```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

To add the IPv6 address on VLAN by using the CLI

At the command prompt, type the following:

```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

To bind IPv6 address to VLAN by using the CLI

At the command prompt, type the following:

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

To bind the IPv6 address to VLAN by using the CLI

At the command prompt, type the following:

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

To display the link local IPv6 address attached to VLAN by using the CLI

At the command prompt, type the following:

```
sh VLAN
```


Example 1

```
add vlan 2
add vlan 3
bind vlan 2 -ifnum 1/2
bind vlan 3 -ifnum 1/3
add ip6 fe80::9404:60ff:fedd:a464/64 -vlan 2 -scope link-local -type
    SNIP
add ip6 fe80::c0ee:7bff:fede:263f/64 -vlan 3 -scope link-local -type
    SNIP
add ip6 3ffe:501:ffff:100:9404:60ff:fedd:a464/64 -vlan 2
add ip6 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64 -vlan 3
bind vlan 2 -ipAddress 3ffe:501:ffff:100:9404:60ff:fedd:a464/64
bind vlan 3 -ipAddress 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64
```

Example 2

```
sh vlan
1)      VLAN ID: 2      VLAN Alias Name:
        Interfaces : 1/6
        IPs :
             3ffe:501:ffff:100:2e0:edff:fe15:ea2a/64
3)      VLAN ID: 3      VLAN Alias Name:
        Link-local IPv6 addr: fe80::9404:60ff:fedd:a464/64
        Interfaces : 1/5
        IPs :
             3ffe:501:ffff:101:2e0:edff:fe15:ea2b/64
 Done
```

More IPv6 cluster configuration

To test IPv6 core protocols, you can use the following new or modified IPv6 configurations.

To set VLAN specific Router Advertisement parameters by using the CLI

At the command prompt, type:

```
set nd6RAvariables -vlan <positive_integer> [-ceaseRouterAdv ( YES | NO
    )] [-sendRouterAdv ( YES | NO )] [-srcLinkLayerAddrOption ( YES | NO
     )][-onlyUnicastRtAdvResponse ( YES | NO )] [-managedAddrConfig (
    YES | NO)] [-otherAddrConfig ( YES | NO )] [-currHopLimit <
    positive_integer>][-maxRtAdvInterval <positive_integer>] [-
    minRtAdvInterval<positive_integer>] [-linkMTU <positive_integer>] [-
```



```
    reachableTime<positive_integer>] [-retransTime <positive_integer>]
    [-defaultLifeTime<integer>]
```

To set an on-link global prefix's configurable parameters by using the CLI

At the command prompt, type:

```
set onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-
    autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-
    decrementPrefixLifeTimes ( YES | NO )] [-prefixValideLifeTime   <
    positive_integer>] [-prefixPreferredLifeTime <positive_integer>]
```

To add configurable parameters to an on-link global prefix by using the CLI

At the command prompt, type:

```
add onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-
    autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-
    decrementPrefixLifeTimes ( YES | NO )]-prefixValideLifeTime <
    positive_integer>][-prefixPreferredLifeTime <positive_integer>]
```

To set an on-link link to the IPv6 prefix's configurable parameters by using the CLI

At the command prompt, type the following:

```
help set onLinkIPv6Prefix
```

To bind an on-link link to the IPv6 prefix's configurable parameters by using the CLI

At the command prompt, type:

```
help bind nd6RAvariables
```

To show nd6RAvariables by using the CLI

At the command prompt, type:

```
help sh nd6RAvariables
```

Example

```
> sh nd6RAvariables
  1) Vlan : 1
     SendAdvert     : NO   CeaseAdv       : NO        SourceLLAddress:
        YES
     UnicastOnly    : NO   ManagedFlag    : NO        OtherConfigFlag:
        NO
```



```
5       CurHopLimit     : 64    MaxRtrAdvInterv: 600        MinRtrAdvInterv:
            198
6       LinkMTU         : 0     ReachableTime  : 0          RetransTimer    :
            0
7       DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay     :
            0
8
9     2) Vlan : 2
10      SendAdvert      : NO    CeaseAdv       : NO         SourceLLAddress:
            YES
11      UnicastOnly     : NO    ManagedFlag    : NO         OtherConfigFlag:
            NO
12      CurHopLimit     : 64    MaxRtrAdvInterv: 600        MinRtrAdvInterv:
            198
13      LinkMTU         : 0     ReachableTime  : 0          RetransTimer    :
            0
14      DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay     :
            0
15  Done
16  >
17  > sh nd6Ravariables  - vlan 2
18    1) Vlan : 2
19      SendAdvert      : NO    CeaseAdv       : NO         SourceLLAddress:
            YES
20      UnicastOnly     : NO    ManagedFlag    : NO         OtherConfigFlag:
            NO
21      CurHopLimit     : 64    MaxRtrAdvInterv: 600        MinRtrAdvInterv:
            198
22      LinkMTU         : 0     ReachableTime  : 0          RetransTimer    :
            0
23      DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay     :
            0
24      Prefix :
25  3ffe:501:ffff:100::/64
26  Done
```

Managing cluster heartbeat messages

September 14, 2021

Managing heartbeat messages in a cluster is similar to managing them in a high availability (HA) configuration. Nodes can send and receive heartbeat messages to and from each other on all interfaces

that are enabled. To avoid increased traffic resulting from heartbeat messages, you can now disable the heartbeat option on node interfaces. However, the heartbeat option on the backplane interface cannot be disabled, because it is required for maintaining connectivity among the cluster nodes.

For more information about managing heart messages, see Managing High Availability Heartbeat Messages on a NetScaler Appliance.

To manage heartbeat messages on a node interface by using the Citrix ADC CLI

At the command prompt, type:

```
set interface <ID> [-HAHeartBeat (ON | OFF)]
Show interface <ID>
```

Configuring owner node response status

September 14, 2021

You can configure the ownerDownResponse option on a node which has a spotted SNIP address. By default, the option is enabled. It allows the spotted IP address to respond to PING or ARP requests (coming from the upstream router) when the node is inactive. If you disable the option, the IP address cannot respond to the router requests when the owner node is inactive.

To know how this feature is used for monitoring static routes in ECMP deployment, see Using Equal Cost Multiple Path (ECMP) topic.

To set owner node response status by using the Citrix ADC CLI

At the command prompt, type:

```
add ns ip <IPAddress> [-ownerNode <positive_integer>] [-
    ownerDownResponse (YES | NO )] [-td <positive_integer>]
```

Example

```
add ns ip 2.2.2.2 255.255.255.0 -ownernode 6  - ownerdownResponse YES
```


To set owner node response status by using the Citrix ADC GUI

1. Navigate to **System > Network > IPs** and click **Add** to create a spotted SNIP address.
2. On the **Create IP Address** page, select, or clear the **ownerDownResponse** check box.

To edit owner node response status by using the Citrix ADC GUI

Navigate to **System > Network > IPs**, select an IP address, and click **Edit** to select or clear the **ownerDownResponse** check box.

Monitor static route (MSR) support for inactive nodes in a spotted cluster configuration

September 14, 2021

In a cluster set up with the MSR option enabled on the route, only active nodes can probe to a static route. It can reach a network while inactive and spare nodes have no link to the route and cannot probe to it. You can now configure an inactive or spare node to send PING and ARP probe to IPv4 route and send ping6 and nd6 probe to IPv6 route. You can perform this only in a spotted cluster configuration in which the SNIP address is active and owned exclusively only by one node.

VRRP interface binding in a single node active cluster

September 14, 2021

When you migrate a high availability (HA) setup to a cluster setup, all configurations must be compatible and must be supportable in the cluster. To achieve this, you can now configure virtual router IDs (VRIDs and VRID6s) on a node interface.

> **Important**
>
> Currently, only a single-node active cluster system supports VRIDs and VRID6s.

For instructions for configuring VRIDs and VRID6s, see Configuring Virtual MAC Addresses.

To configure a virtual router ID on a single-node active cluster, add the VRID or VRID6 and bind it to the cluster-node interface.

To add a VRID by using the Citrix ADC CLI

At the command prompt, type:


```
add vrID <ID>
```

To bind a VRID to the cluster-node interface by using the Citrix ADC CLI

At the command prompt, type:

```
Bind vrid <ID> -ifnum <interface_name> | -trackifNum <interface_name

Add vrID 100
Bind vrid 100  - ifnum 1/1  1/2
done
```

To add a VRID6 by using the Citrix ADC CLI

At the command prompt, type:

```
add vrID6 <ID>
```

To bind a VRID6 to a cluster node interface by using the CLI

At the command prompt, type:

```
bind vrid6 <ID> -ifnum <interface_name> | -trackifNum <interface_name>

Add vrID6 100
Bind vrid6 100  - ifnum 1/1  1/2
Done
```

Cluster setup and usage scenarios

September 14, 2021

This section explains some scenarios in which the Citrix ADC cluster can be set up, and configured for different features and network topologies. Provide feedback if you want any other scenarios to be documented.

Creating a two-node cluster

September 14, 2021

A two-node cluster is an exception to the rule that a cluster is functional only when a minimum of (n/2 +1) nodes, where n is the number of cluster nodes, are able to serve traffic. If the same formula is applied to a two-node cluster, the cluster would fail if one node went down (n/2 +1=2).

A two-node cluster is functional even if only one node is able to serve traffic.

Creating a two node cluster is the same as creating any other cluster. You add a one node as the configuration coordinator and the other node as the other cluster node.

> **Note**
>
> Incremental configuration synchronization is not supported in a two-node cluster. Only full synchronization is supported.

Migrating an HA setup to a cluster setup

September 14, 2021

Migrating an existing high availability (HA) setup to a cluster setup requires you to first remove the Citrix ADC appliances from the HA setup and create a backup of the HA configuration file. You can then use the two appliances to create a cluster and upload the backed-up configuration file to the cluster.

> **Note**
>
> - Before uploading the backed-up HA configuration file to the cluster, you must modify it to make it cluster compatible. Refer to the relevant step of the procedure.
> - Use the **batch -f <backup_filename>** command to upload the backed-up configuration file.

The preceding approach is a basic migration solution which results in downtime for the deployed application. As such, it must be used only in deployments where there is no consideration to application availability.

However, in most deployments, the availability of the application is of paramount importance. For such cases, you must use the approach where an HA setup can be migrated to a cluster setup without any resulting downtime. In this approach, an existing HA setup is migrated to a cluster setup by first removing the secondary appliance and using that appliance to create a single-node cluster. After the cluster becomes operational and serves traffic, the primary appliance of the HA setup is added to the cluster.

To convert a HA setup to cluster setup (without any downtime) by using the command line interface

Let us consider the example of a HA setup with primary appliance (NS1) - 10.102.97.131 and secondary appliance (NS2) - 10.102.97.132.

1. Make sure the configurations of the HA pair are stable.
2. Log on to any one of the HA appliances, go to the shell, and create a copy of the ns.conf file (for example, ns_backup.conf).
3. Log on to the secondary appliance, NS2, and clear the configurations. This operation removes NS2 from the HA setup and makes it a standalone appliance.

```
> clear ns config full
```

> **Note**
>
> - This step is required to make sure that NS2 does not start owning VIP addresses, now that it is a standalone appliance.
> - At this stage, the primary appliance, NS1, is still active and continues to serve traffic.

4. Create a cluster on NS2 (now no longer a secondary appliance) and configure it as a PASSIVE node.

```
> add cluster instance 1

> add cluster node 0 10.102.97.132 -state PASSIVE -backplane
   0/1/1

> add ns ip 10.102.97.133 255.255.255.255 -type CLIP

> enable cluster instance 1

> save ns config

> reboot -warm
```

5. Modify the backed-up configuration file as follows:
   - Remove the features that are not supported on a cluster. For the list of unsupported features, see Citrix ADC Features Supported by a Cluster. It is an optional step. If you do not perform this step, the execution of unsupported commands fails.
   - Remove the configurations that have interfaces, or update the interface names from the c/u convention to the n/c/u convention.

     Example


```
> add vlan 10 -ifnum 0/1
```

must be changed to

```
> add vlan 10 -ifnum 0/0/1 1/0/1
```

- The backup configuration file can have SNIP addresses. These addresses are striped on all the cluster nodes. It is recommended that you add spotted IP addresses for each node.

  Example

```
> add ns ip 1.1.1.1 255.255.255.0 -ownerNode 0

> add ns ip 1.1.1.2 255.255.255.0 -ownerNode 1
```

- Update the host name to specify the owner node.

  Example

```
> set ns hostname ns0 -ownerNode 0

> set ns hostname ns1 -ownerNode 1
```

- Change all other relevant networking configuration that depends on spotted IPs. For example, L3 VLAN, RNAT configuration which uses SNIPs as NATIP, INAT rules that refers to SNIPs/MIPs).

6. On the cluster, do the following:

   - Make the topological changes to the cluster by connecting the cluster backplane, the cluster link aggregation channel, and so on.
   - Apply configurations from the backed-up and modified configuration file to the configuration coordinator through the cluster IP address.

```
> batch -f ns_backup.conf
```

   - Configure external traffic distribution mechanisms like ECMP or cluster link aggregation.

7. Switch the traffic from the HA setup to the cluster.

   - Log on to the primary appliance, NS1, and disable all the interfaces on it.

```
> disable interface <interface_id>
```

   - Log on to the cluster IP address and configure NS2 as an ACTIVE node.


```
> set cluster node 0 -state ACTIVE
```

> **Note**
>
> There might be a small amount (in the order of seconds) of downtime between disabling the interfaces and making the cluster node active.

8. Log on to the primary appliance, NS1, and remove it from the HA setup.
   - Clear all the configurations. This operation removes NS1 from the HA setup and makes it a standalone appliance.

     ```
     > clear ns config full
     ```

   - Enable all the interfaces.

     ```
     > enable interface <interface_id>
     ```

9. Add NS1 to the cluster.
   - Log on to the cluster IP address and add NS1 to the cluster.

     ```
     > add cluster node 1 10.102.97.131 -state PASSIVE -backplane
        1/1/1
     ```

   - Log on to NS1 and join it to the cluster by sequentially running the following commands:

     ```
     > join cluster -clip 10.102.97.133 -password nsroot

     > save ns config

     > reboot -warm
     ```

10. Log on to NS1 and perform the required topological and configuration changes.
11. Log on to the cluster IP address and set NS1 as an ACTIVE node.

    ```
    > set cluster node 1 -state ACTIVE
    ```

Transitioning between a L2 and L3 Cluster

September 14, 2021

> **Note**
>
> Supported from NetScaler 11 onwards.

An L2 cluster is one where all the nodes are from the same network and an L3 cluster is one that can include nodes from different networks. You can seamlessly transition from one type of cluster to the other without any downtime for the applications that are deployed on the Citrix ADC.

Transitioning a cluster from L2 to L3

You can transition to an L3 cluster when you want the cluster to include nodes from other networks.

On the cluster IP address, do the following:

1. Create a node group.

   Example

   ```
   1 > add cluster nodegroup NG0
   ```

   This node group is used in the next step to group all the nodes from the existing L2 cluster.

2. Transition the L2 cluster to an L3 cluster.

   Example

   ```
   1 > set cluster instance 1 -inc ENABLED -nodegroup NG0
   ```

   This command achieves the dual purpose of transitioning to the L3 cluster and also adding all the nodes of the L2 cluster to the node group.

3. Now, you can add more nodes to the cluster as explained in Adding a Node to the Cluster.

Transitioning a cluster from L3 to L2

You can transition to an L2 cluster when you want to retain nodes that belong to a single network.

On the cluster IP address, do the following:

1. Remove the cluster nodes from the networks that you do not want to retain.

   Example

   ```
   1 > rm cluster node <nodeId>
   ```

2. Transition the L3 cluster to a L2 cluster.

   Example


```
1 > set cluster instance 1 -inc DISABLED
```

The cluster now includes nodes only of a single network.

Setting up GSLB in a cluster

September 14, 2021

> **Note**
>
> Supported from NetScaler 10.5 Build 52.11 onwards.

To set up GSLB in a cluster you must bind the different GSLB entities to a node group. The node group must have a single member node.

> **Notes**
>
> - If you have configured the static proximity GSLB method, make sure that the static proximity database is present on all the cluster nodes. It happens by default if the database file is available at the default location. However, if the database file is maintained in a directory other than /var/netscaler/locdb/, you must manually sync the file to all the cluster nodes.
> - The `show gslb domain` command is not supported in a cluster setup.

To set up GSLB in a cluster by using the CLI:

Log on to the cluster IP address and perform the following operations at the command prompt:

1. Configure the different GSLB entities. For information, see GSLB Configuration Entities.

   > **Note**
   >
   > When creating the GSLB site, make sure that you specify the cluster IP address and public cluster IP address. The public cluster IP address is needed only when the cluster is deployed behind a NAT device. While configuring a GSLB site, you must use the cluster IP address of the same site. These parameters are required to ensure the availability of the GSLB auto-sync functionality.

   ```
   add gslb site <siteName> <siteType> <siteIPAddress> -publicIP <ip_addr>
    -clip <ip_addr> <publicCLIP><!--NeedCopy-->
   ```

2. Create a cluster node group.

   ```
   add cluster nodegroup <name> <name>@ [-strict ( YES | NO )] [-sticky (
   YES | NO )] [-state <state>] [-priority <positive_integer>]<!--NeedCopy
   -->
   ```


> **Note**
>
> Enable the sticky option if you want to set up GSLB based for VPN users.

3. Bind a single cluster node to the node group.

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

4. Bind the local GSLB site to the node group.

   ```
   bind cluster nodegroup <name> -gslbSite <string><!--NeedCopy-->
   ```

   > **Note**
   >
   > Make sure that the IP address of the local GSLB site IP address is striped (available across all cluster nodes).

5. Bind the ADNS (or ADNS-TCP) service or the DNS (or DNS-TCP) load balancing virtual server to the node group.

   To bind the ADNS service:

   “‘bind cluster nodegroup -service

   ```
   **To bind the DNS load balancing virtual server:**

   ```bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

6. Bind the GSLB virtual server to the node group.

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

7. [Optional] To set up GSLB based on VPN users, bind the VPN virtual server to the GSLB node group.

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

8. Verify the configurations.

   ```
   show gslb runningConfig<!--NeedCopy-->
   ```

To set up GSLB in a cluster by using the GUI:

Log on to the cluster IP address and perform the following operations in the Configuration tab:

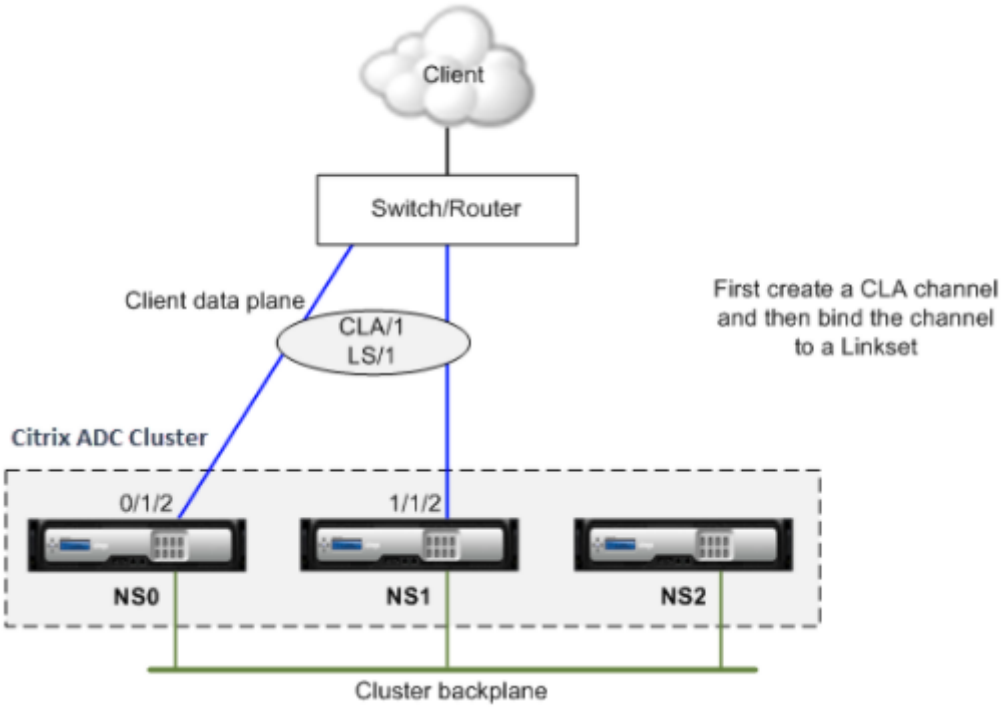
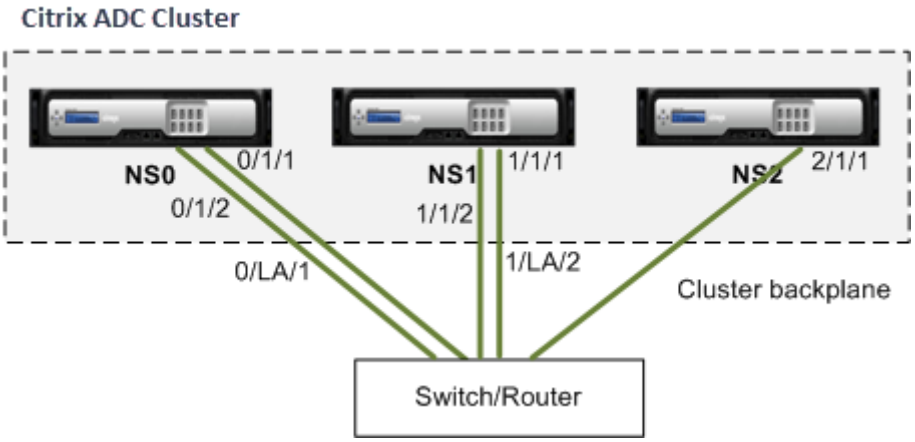
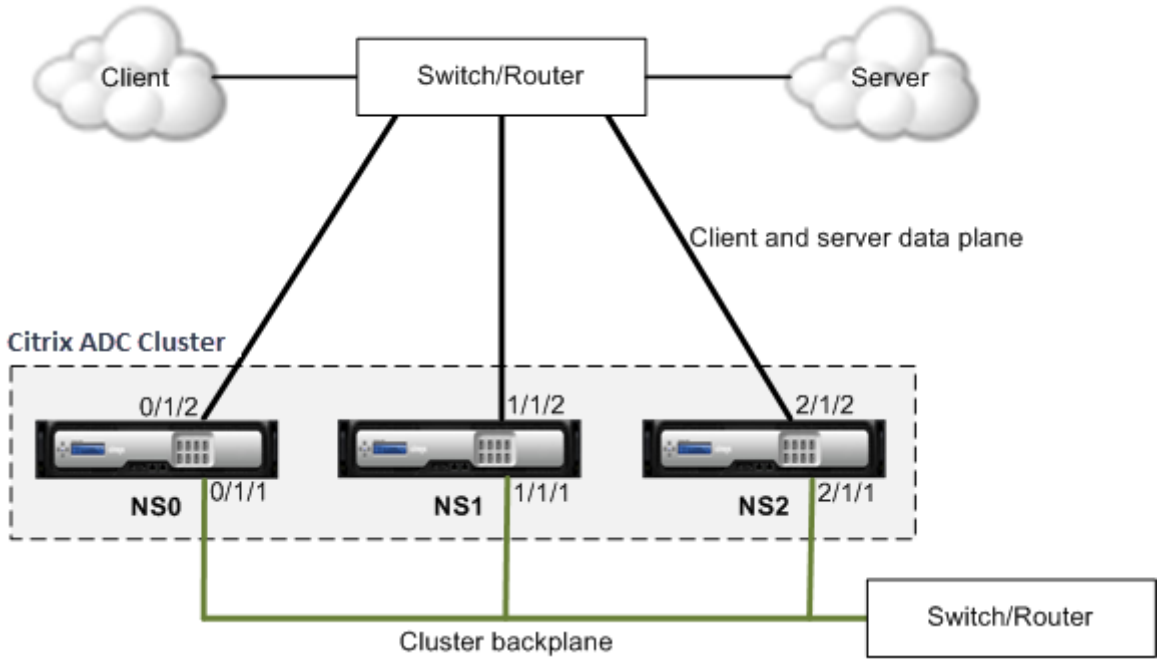
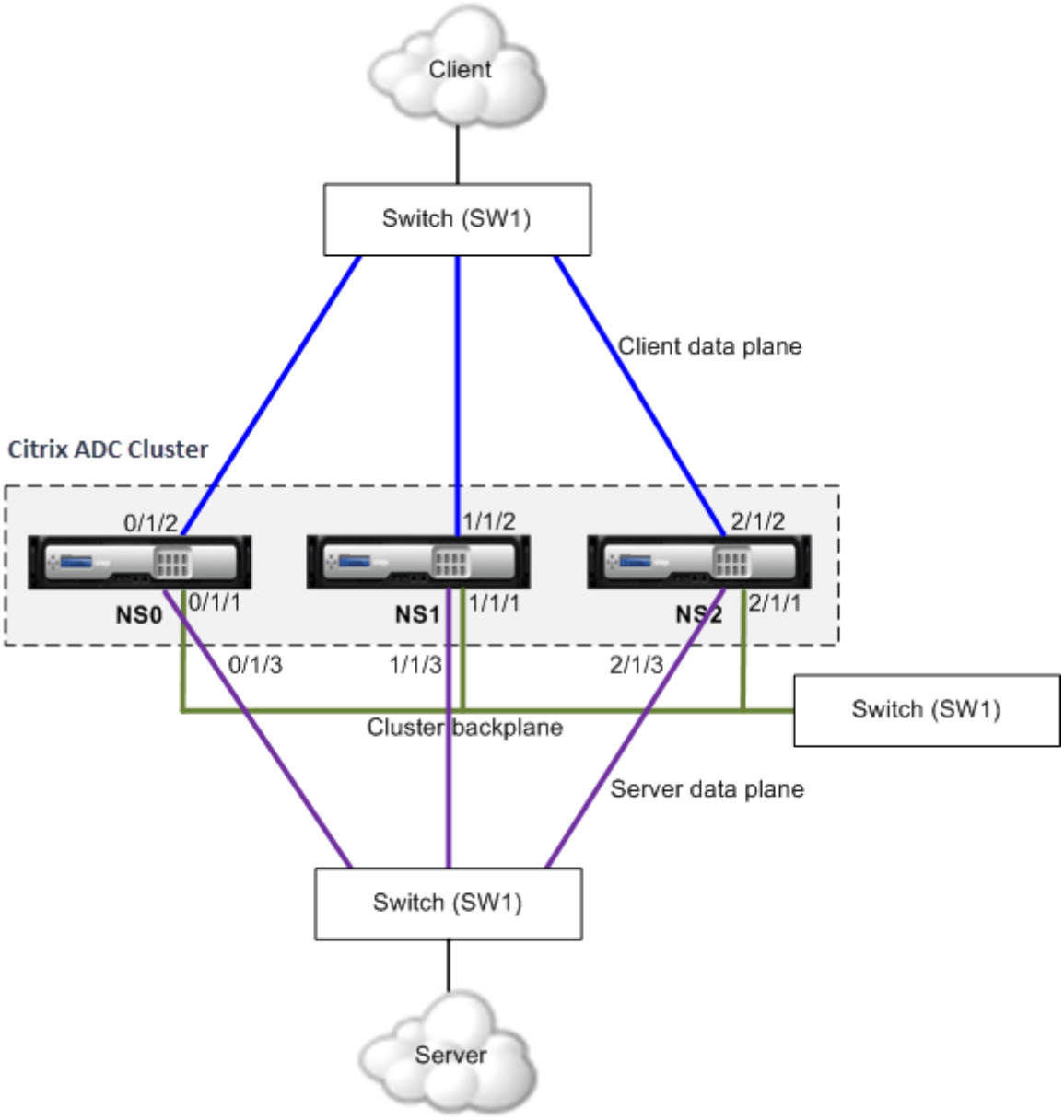
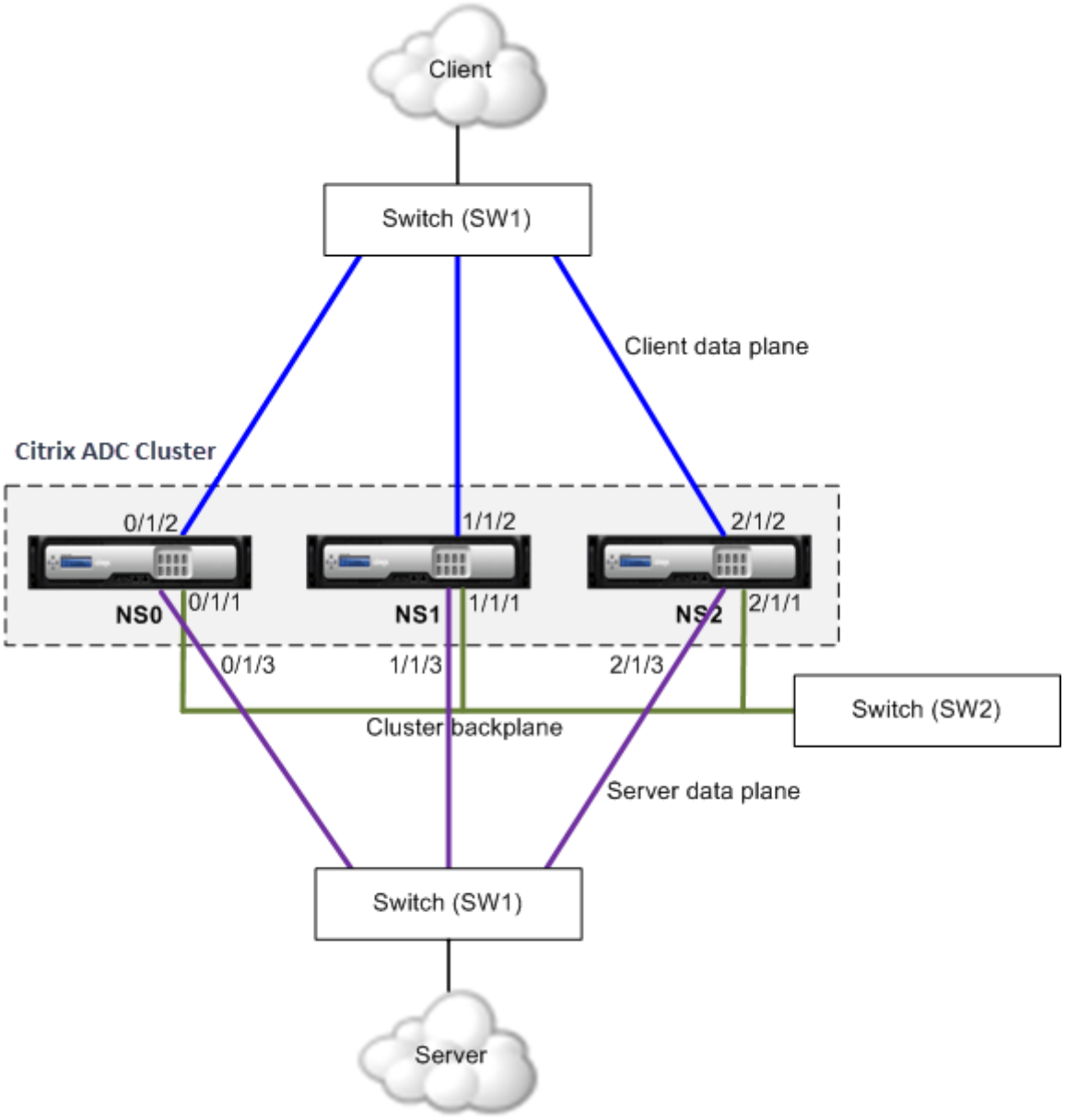
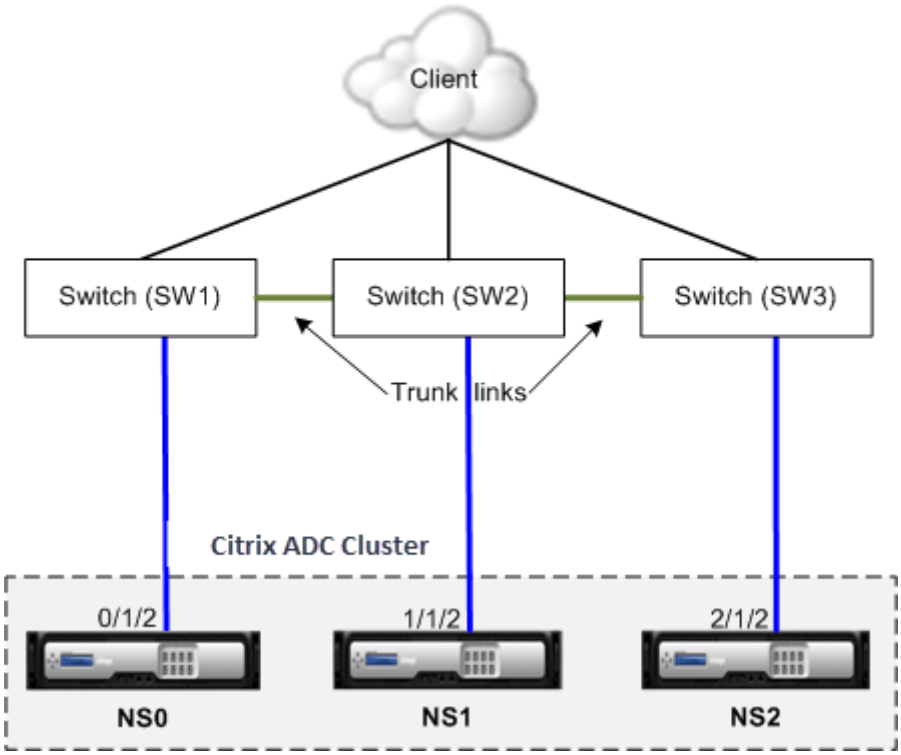
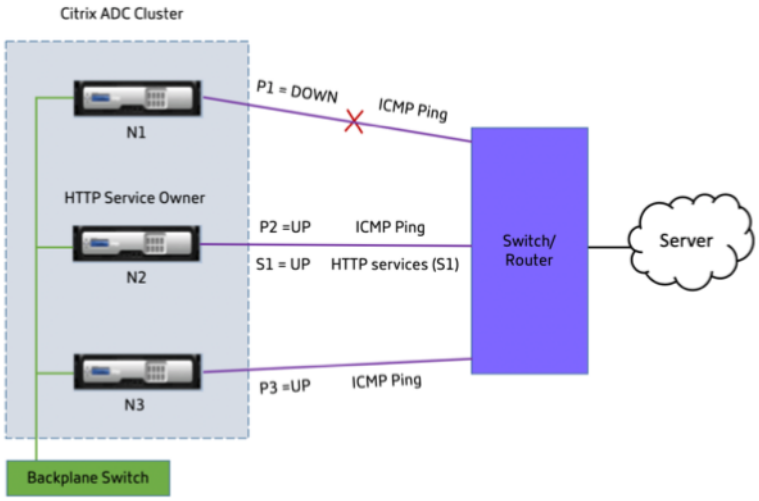
1. Configure the GSLB entities.

   Navigate to **Traffic Management > GSLB** to perform the required configurations.

2. Create a node group and perform other node group related configurations.

   Navigate to **System > Cluster > Node Groups** to perform the required configurations.

For the detailed configurations to be performed, see the description provided in the preceding CLI procedure.


   ```

### Support for GSLB parent-child topology in a cluster

Starting with NetScaler 12.1 build 49.xx, GSLB parent-child topology is supported in cluster.

For more information about the parent-child topology, see Parent-child topology deployment using the MEP protocol.

#### To set up GSLB parent-child topology in a cluster by using the CLI

**Parent site**

Perform the following configuration:

1. Create a cluster node group.

   ```
   add cluster nodegroup <name>
   ```

   Example:

   ```
   add cluster nodegroup parentng
   ```

2. Bind a single cluster node to the node group.

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   Example:

   ```
   bind cluster nodegroup parentng -node n2
   ```

3. Bind the local GSLB site to the node group.

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   Example:

   ```
   bind cluster nodegroup parentng -gslbSite site1
   ```

4. Bind the ADNS (or ADNS-TCP) service or the DNS (or DNS-TCP) load balancing virtual server to the node group.

   ```
   bind cluster nodegroup <name> -service <string>
   ```

   Example:

   ```
   bind cluster nodegroup parentng - service ADNS
   ```

5. Bind the GSLB virtual server to the node group.

   ```
   bind cluster nodegroup <name> -vServer <string>
   ```

   Example:

   ```
   bind cluster nodegroup parentng -vServce gslbvs1
   ```


Child site

Perform the following configuration:

1. Create a cluster node group.

   ```
   add cluster nodegroup <name>
   ```

   Example:

   ```
   add cluster nodegroup childng
   ```

2. Bind a single cluster node to the node group.

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   Example:

   ```
   bind cluster nodegroup childng -node -n3
   ```

3. Bind the local GSLB site to the node group.

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   Example:

   ```
   bind cluster nodegroup childng -gslbSite site1
   ```

> **Note**
>
> For parent and child sites to exchange aggregated statistics in metric-based load balancing methods, you must add local GSLB services on the child site. The metric-based load balancing methods are least connection, least bandwidth, and least packets.

To set up GSLB parent-child topology in a cluster by using the GUI

1. Configure the GSLB entities.

   Navigate to **Traffic Management** > **GSLB** to perform the required configurations.

2. Create a node group.

   Navigate to **System** > **Cluster** > **Node Groups** to perform the required configurations.

3. In the Node Group page, select the node group to which you want to bind a node, click **Edit**, and perform the following tasks. You can also perform these tasks when adding a node group.

   - Bind a node to the node group.

     In **Advance Settings**, click **Cluster Nodes** and perform the following tasks:

     - In **Cluster Nodes** section, click **No Cluster Node**.
     - In **Select Cluster Node**, click > and select the node that you want to bind to the node group. You can also add a cluster node.

- Bind the local GSLB site to the node group.

  In Advance Settings, click GSLB Sites and perform the following tasks:

  - In the **GSLB Sites** section, click No GSLB Site.
  - In the **Select GSLB Site**, click > and select the GSLB site that you want to bind to the node group. You can also add a GSLB site.

- Bind the GSLB virtual server to the node group.

  In **Advance Settings**, click **Virtual Servers** and perform the following task:

  - In the **Virtual Servers** pane, click +.
  - In **Choose Virtual Server**, select the server that you want to bind to the node group.

- Bind the ADNS (or ADNS-TCP) service or the DNS (or DNS-TCP) load balancing virtual server to the node group.

  In **Advance Settings**, click **Services** and perform the following tasks:

  - In **Services** section, click **No Service**.
  - In **Select Service**, select the service that you want to bind to the node group. You can also add a service.

> **Note**
>
> For child sites, you only have to bind the cluster node and local GSLB site to the node group.

Using cache redirection in a cluster

September 14, 2021

Cache redirection in a cluster works in the same way as it does on a standalone Citrix ADC appliance. The only difference is that the configurations are done on the cluster IP address. For more information on cache redirection, see Cache Redirection.

Points to remember when using cache redirection in transparent mode on a cluster:

- Before configuring cache redirection, make sure that you have connected all nodes to the external switch and that you have linksets configured. Otherwise, client requests are dropped.
- When MAC mode is enabled on a load balancing virtual server, make sure MBF mode is enabled on the cluster by using the enable ns mode MBF command. Otherwise, the requests are sent to origin server directly instead of being sent to the cache server.

Using L2 mode in a cluster setup

September 14, 2021

> **Note**
>
> Supported from NetScaler 10.5 and later releases.

To use L2 mode in a cluster setup, you must make sure of the following:

- Spotted IP addresses must be available on all the nodes as required.
- Linksets must be used to communicate with the external network.
- Asymmetric topologies or asymmetric cluster LA groups are not supported.
- Cluster LA group is recommended.
- Traffic is distributed between the cluster nodes only for deployments where services exist.

Using cluster LA channel with linksets

September 14, 2021

In an asymmetric cluster topology, some cluster nodes are not connected to the upstream network. In such a case, you must use linksets. To optimize the performance, you can bind the interfaces that are connected to the switch as a cluster LA channel and then bind the channel to a linkset.

To understand how a combination of cluster LA channel and linksets can be used, consider a three-node cluster for which the upstream switch has only two ports available. You can connect two of the cluster nodes to the switch and leave the other node unconnected.

> **Note**
>
> Similarly, you can also use a combination of ECMP and linksets in an asymmetric topology.

Figure 1. Linksets and cluster LA channel topology



To configure cluster LA channel and linksets by using the CLI

1. Log on to the cluster IP address.
2. Bind the connected interfaces to a cluster LA channel.

```
add channel CLA/1 -ifnum 0/1/2 1/1/2
```

3. Bind the cluster LA channel to the linkset.

```
add linkset LS/1 -ifnum CLA/1
```

Backplane on LA channel

September 14, 2021

In this deployment, LA channels are used for the cluster backplane.



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with the backplane interfaces as LA channels

1. Create a cluster of nodes NS0, NS1, and NS2.

   a) Log on to the first node that you want to add to the cluster and do the following:

   ```
   > create cluster instance 1
   > add cluster node 0 10.102.29.60 -state ACTIVE
   > enable cluster instance 1
   > add ns ip 10.102.29.61 255.255.255.255 -type CLIP
   > save ns config
   > reboot -warm
   ```

   b) Log on to the cluster IP address and do the following:

   ```
   > add cluster node 1 10.102.29.70 -state ACTIVE
   > add cluster node 2 10.102.29.80 -state ACTIVE
   ```

   c) Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

   ```
   > join cluster -clip 10.102.29.61 -password nsroot
   > save ns config
   > reboot -warm
   ```

   As seen in the preceding commands, the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. Log on to the cluster IP address and do the following:

a) Create the LA channels for nodes NS0 and NS1.

```
> add channel 0/LA/1 -ifnum 0/1/1 0/1/2
> add channel 1/LA/2 -ifnum 1/1/1 1/1/2
```

b) Configure the backplane for the cluster nodes.

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/1/1
```

Common interfaces for client and server and dedicated interfaces for backplane

September 14, 2021

It is a one-arm deployment of the Citrix ADC cluster. In this deployment, the client and server networks use the same interfaces to communicate with the cluster. The cluster backplane uses dedicated interfaces for inter-node communication.



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with a common interface for the client and server and a different interface for the cluster backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
2. Log on to the first node that you want to add to the cluster and do the following:

```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

3. Log on to the cluster IP address and do the following:

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

4. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

As seen in the preceding commands, the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

1. On the cluster IP address, create VLANs for the backplane interfaces and for the client and server interfaces.

   //For the backplane interfaces

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //For the interfaces that are connected to the client and server networks.

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

2. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco®

Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

//For the backplane interfaces. Repeat for each interface…

```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

//For the interfaces connected to the client and server networks. Repeat for each interface…

```
> interface Ethernet2/47
  switchport access vlan 200
  switchport mode access
  end
```

Common switch for client, server, and backplane

September 14, 2021

In this deployment, the client, server, and backplane use dedicated interfaces on the same switch to communicate with the Citrix ADC cluster.



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with a common switch for the client, server, and backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
2. Log on to the first node that you want to add to the cluster and do the following:

```
1 > create cluster instance 1
```



```
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

3. Log on to the cluster IP address and do the following:

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

4. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

As seen in the preceding commands, the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

1. On the cluster IP address, create VLANs for the backplane, client, and server interfaces.

   //For the backplane interfaces

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //For the client-side interfaces

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

   //For the server-side interfaces

```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

2. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

   //For the backplane interfaces. Repeat for each interface…


```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

//For the client interfaces. Repeat for each interface…

```
> interface Ethernet2/48
  switchport access vlan 200
  switchport mode access
  end
```

//For the server interfaces. Repeat for each interface…

```
> interface Ethernet2/49
  switchport access vlan 300
  switchport mode access
  end
```

Common switch for client and server and dedicated switch for backplane

September 14, 2021

In this deployment, the clients and servers use different interfaces on the same switch to communicate with the Citrix ADC cluster. The cluster backplane uses a dedicated switch for inter-node communication.



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with the same switch for the clients and servers and a different switch for the cluster backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
   - Log on to the first node that you want to add to the cluster and do the following:


```
> create cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

- Log on to the cluster IP address and do the following:

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

- Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot -warm
```

As seen in the preceding commands, the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. On the cluster IP address, create VLANs for the backplane, client, and server interfaces.

   //For the backplane interfaces

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //For the client-side interfaces

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

   //For the server-side interfaces

```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

3. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

   //For the backplane interfaces. Repeat for each interface…


```
> interface Ethernet2/47
> switchport access vlan 100
> switchport mode access
> end
```

//For the client interfaces. Repeat for each interface…

```
> interface Ethernet2/48
> switchport access vlan 200
> switchport mode access
> end
```

//For the server interfaces. Repeat for each interface…

```
> interface Ethernet2/49
> switchport access vlan 300
> switchport mode access
> end
```

Different switch for every node

September 14, 2021

In this deployment, each cluster node is connected to a different switch and trunk links are configured between the switches.



The cluster configurations are the same as the other deployments scenarios. Most of the client-side configurations is done on the client-side switches.

Sample cluster configurations

September 14, 2021

The following example can be used to configure a four-node cluster with ECMP, cluster LA, or Linksets.

1. Create the cluster.
   - Log on to the first node.
   - Add the cluster instance.

     ```
     1   > add cluster instance 1
     ```

   - Add the first node to the cluster.

     ```
     1   > add cluster node 0 10.102.33.184 -backplane 0/1/1
     ```

   - Enable the cluster instance.

     ```
     1   > enable cluster instance 1
     ```


- Add the cluster IP address.

```
1   > add ns ip 10.102.33.185 255.255.255.255 -type CLIP
```

- Save the configurations.

```
1   > save ns config
```

- Warm reboot the appliance.

```
1   > reboot -warm
```

2. Add the other three nodes to the cluster.
   - Log on to cluster IP address.
   - Add the second node to the cluster.

```
1   > add cluster node 1 10.102.33.187 -backplane 1/1/1
```

   - Add the third node to the cluster.

```
1   > add cluster node 2 10.102.33.188 -backplane 2/1/1
```

   - Add the fourth node to the cluster.

```
1   > add cluster node 3 10.102.33.189 -backplane 3/1/1
```

3. Join the added nodes to the cluster. This step is not applicable for the first node.
   - Log on to each newly added node.
   - Join the node to the cluster.

```
1   > join cluster -clip 10.102.33.185 -password nsroot
```

   - Save the configuration.

```
1   > save ns config
```

   - Warm reboot the appliance.

```
1   > reboot -warm
```

4. Configure the Citrix ADC cluster through the cluster IP address.

   // Enable load balancing feature


```
1      > enable ns feature lb
```

// Add a load balancing virtual server

```
1      > add lb vserver first_lbvserver http
2      ....
3      ....
```

5. Configure any one of the following (ECMP, cluster LA, or Linkset) traffic distribution mechanisms for the cluster.

   ECMP

   - Log on to the cluster IP address.
   - Enable the OSPF routing protocol.

     ```
     1   > enable ns feature ospf
     ```

   - Add a VLAN.

     ```
     1   > add vlan 97
     ```

   - Bind the interfaces of the cluster nodes to the VLAN.

     ```
     1   > bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
     ```

   - Add a spotted SNIP on each node and enable dynamic routing on it.

     ```
     1   > add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 -
           dynamicRouting ENABLED
     2   > add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 -
           dynamicRouting ENABLED
     3   > add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 -
           dynamicRouting ENABLED
     4   > add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 -
           dynamicRouting ENABLED
     ```

   - Bind one of the SNIP addresses to the VLAN.

     ```
     1   > bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
     ```

   - Configure the routing protocol on ZebOS by using the VTYSH shell.

     Static cluster LA

     - Log on to the cluster IP address.

- Add a cluster LA channel.

```
> add channel CLA/1 -speed 1000
```

- Bind the interfaces to the cluster LA channel.

```
> bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
```

- Perform an equivalent configuration on the switch.

  Dynamic cluster LA

  * Log on to the cluster IP address.
  * Add the interfaces to the cluster LA channel.

```
> set interface 0/1/5 -lacpmode active -lacpkey 5 -
    lagtype cluster
> set interface 1/1/5 -lacpmode active -lacpkey 5 -
    lagtype cluster
> set interface 2/1/5 -lacpmode active -lacpkey 5 -
    lagtype cluster
> set interface 3/1/5 -lacpmode active -lacpkey 5 -
    lagtype  cluster
```

  * Perform an equivalent configuration on the switch.

  Linksets. Assume that the node with nodeId 3 is not connected to the switch. You must configure a linkset so that the unconnected node can use the other node interfaces to communicate with the switch.

a) Log on to the cluster IP address.

b) Add a linkset.

```
> add linkset LS/1
```

c) Bind the connected interfaces to the linkset.

```
> bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6
```

6. Update the state of the cluster nodes to ACTIVE.

```
> set cluster node 0 -state ACTIVE
> set cluster node 1 -state ACTIVE
> set cluster node 2 -state ACTIVE
> set cluster node 3 -state ACTIVE
```


Using VRRP in a cluster setup

September 14, 2021

Virtual Router Redundancy Protocol (VRRP) is supported in a cluster setup for both IPv4 and IPv6. The two VRRP features supported in a cluster setup are Interface based VRRP and IP based VRRP.

IP based VRRP

In IP based VRRP, striped VIP addresses bound to the same VRID are configured on all nodes of a cluster setup. These VIP addresses are active on all the nodes

One of the cluster nodes acts as the VRID owner and sends out the VRRP advertisement to other nodes. If there is failure of the VRID owner node, another node in the cluster assumes the ownership of the VRID and starts sending VRRP advertisements. You can also assign a specific cluster node as the owner of the VRID.

> **Note**
>
> Citrix recommends you to use IP based method for VRRP deployment in cluster.

Configuring IP based VRRP for IPv4

Perform the following tasks on a cluster setup for configuring IP based VRRP for IPv4:

- **Add a VRID**. A VRID is an integer used by the Cluster setup to form a virtual MAC address. The generic VMAC address is in the form of 00:00:5e:00:02:<VRID>.
- **(Optional) Assign a node as the owner of the virtual MAC address**. You can set the owner node parameter (while adding or modifying VRID6) to the ID of the cluster node to assign it as the owner of the virtual MAC address. If the assigned owner node fails, one of the UP cluster nodes is dynamically elected as the owner of the virtual MAC address. You can set the owner node using the `set vrID <id> -ownerNode <positive_interger>` command.
- **Bind the VRID to the VIP address of the nodes**. Bind the created VRID to the striped VIP address.

To add a VRID by using the CLI

At the command prompt, type:

```
1 -  add vrid <ID> [-ownerNode <positive_integer>]
2 -  show vrid <ID>
```


To bind the VRID to VIP address by using the CLI

At the command prompt, type:

- `set ns ip <IPv4Address> -vrid <ID><!--NeedCopy-->`
- `show vrid <ID><!--NeedCopy-->`

To add a VRID by using the GUI

1. Navigate to **System > Network > VMAC** and, on the **VMAC** tab, click **Add**.
2. On the Create **VMAC** page, specify a value in the **Virtual Router ID** field and then click **Create**.

To bind the VRID to a VIP address by using the GUI

1. Navigate to **System > Network > IPs**, on the **IPV4s** tab, select a VIP address and click **Edit**.
2. Set the **Virtual Router ID** parameter while editing the VIP configuration.

```
1 > add vrid 90
2  Done
3 > set ns ip 192.0.2.90  - vrid 90
4  Done
```

Configuring IP based VRRP for IPv6

Perform the following tasks on a cluster setup for configuring IP based VRRP for IPv6:

- **Add a VRID6**. A VRID6 is an integer used by the Cluster setup to form a virtual MAC6 address. The generic VMAC6 address is in the form of 00:00:5e:00:02:<VRID6>.
- **(Optional) Assign a node as the owner of the virtual MAC6 address**. You can set the owner node parameter (while adding or modifying VRID6) to the ID of the cluster node to assign it as the owner of the virtual MAC6 address. If the assigned owner node fails, one of the UP cluster nodes is dynamically elected as the owner of the virtual MAC6 address.
- **Bind the VRID6 to the VIP6 address of the nodes**. Bind the created VRID6 to the striped VIP6 address.

To add a VRID6 by using the CLI

At the command prompt, type:

- `add vrid6 <ID> [-ownerNode <positive_integer>]<!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

To bind the VRID6 to VIP6 address by using the CLI

At the command prompt, type:

- `set ns ip6 <IPv6Address> -vrid6 <ID><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

To add a VRID6 by using the GUI

1. Navigate to **System > Network > VMAC** and, on the **VMAC6** tab, click **Add**.
2. On the **Create virtual MAC6** page, specify a value in the **Virtual Router ID** field and then click **Create**.

To bind the VRID6 to a VIP6 address by using the GUI

1. Navigate to **System > Network > IPs**, on the **IPV6s** tab, select a VIP address and click **Edit**.
2. Set the **Virtual Router ID** parameter while editing the VIP6 configuration.

```
1 > add vrid6 90
2  Done
3 > set ns ip6 2001:db8::5001  - vrid6 90
4  Done
```

Interface based VRRP

In the interface based VRRP feature, the same virtual MAC address is configured on both the nodes of the cluster. This virtual MAC address is used in GARP advertisements and ARP responses for the IP addresses configured on a node. This feature is useful in an active-spare two-node cluster setup that has external devices/routers that do not accept GARP advertisements.

> **Note**
>
> The interface based VRRP feature is applicable only to a two-node cluster with one node in active state and the other node serving as a spare.

With the same virtual MAC address on both cluster nodes, when the active node goes down and the spare node takes over as active, the MAC address for the IP addresses on the new active node remain unchanged and the ARP tables on the external devices/routers do not need to be updated.

Configuring interface based VRRP for IPv4

Perform the following tasks on a cluster setup to configure interface based VRRP for IPv4:

- **Add a VRID**. A VRID is an integer used by the Cluster setup to form a virtual MAC address.
- **Bind the VRID to node interfaces**. Bind the interfaces to the created VRID. The bound interfaces (in the current active node) use the virtual MAC address in GARP advertisements and ARP responses for its IPv4 addresses. You must associate the VRID to the interfaces of both nodes of the active-spare cluster setup. This is because unlike in a high availability setup, interface IDs differ in a cluster setup.

To add a VRID by using the CLI

At the command prompt, type:

```
- add vrid <ID>
- show vrid <ID>
```

To bind the VRID to an interface by using the CLI

At the command prompt, type:

```
- bind vrid <ID> -ifnum <interface_name>
- show vrid <ID>
```

To add a VRID and bind it to interfaces by using the GUI

1. Navigate to **System > Network > VMAC** and, on the **VMAC** tab, click **Add**.
2. On the **Create virtual MAC** page, specify a value in the **Virtual Router ID*** field, bind interfaces in the **Associate Interfaces** section, and then click **Create**.

```
> add vrid 300
 Done
> bind vrid 300 -ifnum 1/1/2 2/1/3
 Done
```

Configuring interface based VRRP for IPv6

Perform the following tasks on a cluster setup to configure interface based VRRP for IPv6:

- **Add a VRID6**. A VRID6 is an integer used by the Cluster setup to form a virtual MAC6 address. The generic VMAC6 address is in the form of 00:00:5e:00:01:<VRID6>.
- **Bind the VRID6 to node interfaces**. Bind the interfaces to the created VRID6. The bound interfaces (in the current active node) use the virtual MAC6 address in GARP advertisements and ARP responses for its IPv6 addresses. You must associate the VRID6 to the interfaces of both nodes

of the active-spare cluster setup. This is because unlike in a high availability setup, interface IDs differ in a cluster setup.

To add a VRID6 by using the CLI

At the command prompt, type:

```
1 -  add vrid6 <ID>
2 -  show vrid6 <ID>
```

To bind the VRID6 to an interface by using the CLI

At the command prompt, type:

- `bind vrid6 <ID> -ifnum <interface_name><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

To add a VRID6 and bind it to interfaces by using the GUI

1. Navigate **System > Network > VMAC** and, on the **VMAC6** tab, click **Add**.
2. On the **Create virtual MAC6** page, specify a value in the **Virtual Router ID** field, bind interfaces in the **Associate Interfaces** section, and then click **Create**.

```
1 > add vrid6 100
2  Done
3 > bind vrid6 100 -ifnum 0/1/1 1/1/2 2/1/3
4  Done
```

Monitoring services in a cluster using path monitoring

September 14, 2021

In a cluster setup, ownership for monitoring services is distributed among the nodes. Therefore, different nodes monitor different services. The node that monitors a service is called the service owner. Only the service owner probes the server to monitor the status of the services assigned to it. It further communicates the status of the services to all other nodes within the cluster. The drawback with distributed monitoring is that the network connectivity and link state between all nodes and the server is not determined. To overcome this drawback, you can use path monitoring.

> **Note**
>
> You cannot select a node to monitor a service. The selection of nodes to monitor a service is done through an internal mechanism. You can see the owner node to monitor services by using the `show service <service name>` and `show serviceGroup <service group name>` command.

Path monitoring checks the network connectivity and link state between a node and the service provided by the server. A node sends ICMP pings to verify whether the server is reachable or not.

How path monitoring works

Consider an example of a Citrix ADC cluster consisting of three nodes N1, N2, and N3. N2 is the service owner that monitors the state of HTTP services (S1). It advertises the service state to other nodes in the cluster. Path monitoring is enabled on all nodes in the cluster, for all the services. Each node sends only an ICMP ping to the server. The service owner sends both the HTTP service request and an ICMP ping. Each node reports its path monitoring state to the service owner.



The following two parameters determine the service state of a node:

- S = service state advertised by the service owner
- P = path monitoring state of each node

Whether a node can reach a server or not, determines the path monitoring state for that node.

The following table shows the service state set based on the path monitoring state, when the pathMonitorIndv parameter is enabled or disabled.

Parameter	Path monitoring state	Service state
pathMonitorIndv = NO; Is the default configuration.	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = DOWN
	P3 = UP	S1 = DOWN
pathMonitorIndv = YES	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = UP
	P3 = UP	S1 = UP

In this example, the service owner decides the service state for all the nodes based on the node whose path monitoring state is set to DOWN. If the path monitoring state for any of the nodes is DOWN, then the service owner sets the service state for all the nodes as DOWN. The service state for all the nodes is set to UP only if the path monitoring state for each of the node is UP.

You can use path monitoring for individual nodes by enabling the pathMonitorIndv parameter. This parameter enables the service owner to set the service state for each node based on the path monitoring state of that respective node.

> **Note**
>
> If the pathMonitorIndv parameter is set, some features like persistence might break.

Configuring path monitoring

Path monitoring is applicable for all the services and service groups. Path monitoring parameter is disabled by default.

To enable path monitoring for services/service groups by using the CLI

At the command prompt, type:

```
add service <service name> <IP address> <service type> <port> [-
    pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]

add servicegroup <servicegroup name> <service type> [-pathMonitor <YES
    | NO>] [-pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

Example:


```
add service s1 1.1.1.1 HTTP 80 -pathMonitor YES
add servicegroup sg_1 HTTP -pathMonitor YES

add service s1 1.1.1.1 HTTP 80 -pathMonitor YES -pathMonitorIndv YES
add servicegroup sg_1 HTTP -pathMonitor YES -pathMonitorIndv YES
<!--NeedCopy-->
```

You can also set the path monitoring parameter from the set command, as follows:

```
set service <service name> [-pathMonitor <YES | NO>] [-pathMonitorIndv
   <YES | NO>]
set servicegroup <servicegroup name> [-pathMonitor <YES | NO>] [-
   pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

Example:

```
set service s1 -pathMonitor YES
set servicegroup sg_1 -pathMonitor YES


set service s1 -pathMonitorIndv YES
set servicegroup sg_1 -pathMonitorIndv NO
<!--NeedCopy-->
```

To enable path monitoring for services/service groups by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.

   For service groups, navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.

2. In the **Services/Service Groups** pane, select a service/service group from the list, and then double-click to open it.
3. On the **Service Settings** tab, click **Edit**.
4. Select **Path Monitoring**.
5. Select **Individual Path Monitoring**, if you want to apply it, and then click **OK**.

> **Note**
>
> You can enable Individual Path Monitoring only if you enable Path Monitoring.

Backup and restore of cluster setup

September 14, 2021

You can back up the current state of a Citrix ADC cluster node. Later, you can use the backed-up files to restore the node to the same cluster state. As a precautionary measure, you must use this feature before performing an upgrade on the cluster nodes.

Back up a cluster Setup

You can take a basic or full backup depending on the following:

- Type of data to be backed up.
- Frequency at which you create a backup.
- **Basic back up.** Backs up only configuration files. You might want to perform this type of backup frequently, because the files it backs up change constantly. The files that are backed up are listed in the table.

Directory

Sub-Directory or Files

/nsconfig/

- ns.conf
- ZebOS.conf
- rc.netscaler
- snmpd.conf
- nsbefore.sh
- nsafter.sh
- inetd.conf
- ntp.conf
- syslog.conf
- newsyslog.conf
- crontab
- host.conf
- hosts
- ttys
- sshd_config
- httpd.conf
- monitrc
- rc.conf

- ssh_config
- local time
- issue
- issue.net

/var/

- download/*
- log/wicmd.log
- wi/tomcat/webapps/*
- wi/tomcat/logs/*
- wi/tomcat/conf/catalina/localhost/*
- nslw.bin/etc/krb.conf
- nslw.bin/etc/krb.keytab
- netscaler/locdb/*
- lib/likewise/db/*
- vpn/bookmark/*
- netscaler/crl
- nstemplates/*
- learnt_data/*

/netscaler/

- custom.html
- vsr.html
- **Full back up.** Apart from the files that are backed up by a basic backup, a full backup backs up some less frequently updated files. The files that are backed up when using the full backup option are listed in the table.

Directory

Sub-Directory or Files

/nsconfig/

- ssl/*
- license/*
- fips/*

/var/

- netscaler/ssl/*
- wi/java_home/jre/lib/security/cacerts/*
- wi/java_home/lib/security/cacerts/*

> **Important**
>
> The backup and restore do not work if CLAG is configured on an SDX cluster setup.

The backup is stored as a compressed TAR file in the /var/ns_sys_backup/ directory. To avoid issues due to non-availability of disk space, you can store a maximum of 50 backup files in this directory. You can use the rm system backup command to delete existing backup files so that you can create more backups.

When you perform the back-up operation on a CLIP of a cluster setup, back up files are created on each of the cluster nodes.

How to back up a cluster setup

To back up the cluster setup on CLIP by using the Citrix ADC CLI.

At the command prompt, do the following:

- Save the configuration.

  ```
  save ns config<!--NeedCopy-->
  ```

- Create the backup file (basic or full).

  "'create system backup [][–level (basic | full)][-comment]

  ```
  **Example**

  ```create system backup cluster-backup-1  -level basic<!--
     NeedCopy-->
  ```

The preceding command creates a back-up TAR file on each of the cluster node with the specified file name. For example, the CLUSTER-backup-1.tgz file is created on each of the cluster node.

> **Note**
>
> If the file name is not specified, back up TAR files are created on each of the cluster nodes with the following naming convention:
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp>.tgz<!--NeedCopy-->
> - backup_<level>_<nsip_address of the cluster node 1>_<date-timestamp>.tgz<!--NeedCopy-->
>
> For example, on a three node cluster setup,
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp>.tgz<!--NeedCopy--> is created on node0


  ```

- backup _<level>_<nsip_address of the cluster node 1>_<date-timestamp >.tgz<!--NeedCopy--> is created on node1
- backup_<level>_<nsip_address of the cluster node 2>_<date-timestamp >.tgz<!--NeedCopy--> is created on node2

- Verify the created backup files on CLIP.

  ```
  show system backup<!--NeedCopy-->
  ```

Restore a cluster setup

When a cluster node becomes faulty, you can replace this node with a new node. You can set the new node for a cluster, using a backup file of the faulty node.

For example, in a three-node cluster setup, if node1 becomes faulty, you can replace this faulty node with a new node as node1. Using the restore operation, you can restore one of the faulty node's backup files on the new node.

> **Note**
>
> The restore operation does not succeed if the backup file is renamed or if the contents of the file are modified.

How to restore a cluster node

To restore a cluster node by using the CLI

At the command prompt, do the following:

- Obtain a list of the backup files available on CLIP.

  ```
  show system backup<!--NeedCopy-->
  ```

- Copy the backup tar file to the /var/ns_sys_backup directory of the cluster node, which is to be restored.
- Add the backup tar file to the cluster node memory by running the following command on the cluster node.

  "'add system backup

  ```
  **Example**

  ```add system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
  ```

  > **Note**
  >
  > The command must be run on the cluster node to be restored.


  ```

- Restore the cluster node by specifying the backup file.

  “‘restore system backup

```
Example

```restore system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
```

> **Note**
>
> The command must be run on the cluster node to be restored.

- Reboot the cluster node.

  reboot

> **Note**
>
> The command must be run on the cluster node to be restored.

Upgrading or downgrading the Citrix ADC cluster

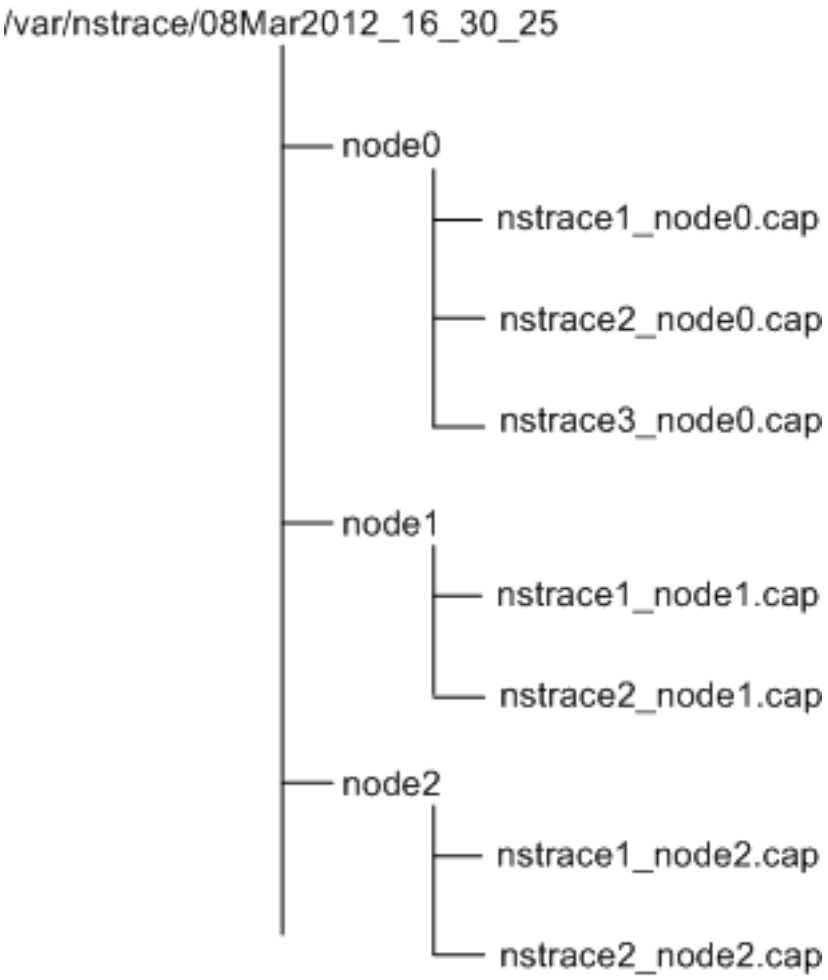
September 14, 2021

All the nodes of a Citrix ADC cluster must be running the same software version. Therefore, to upgrade or downgrade the cluster, you must upgrade or downgrade each Citrix ADC appliance of the cluster, one node at a time.

A node that is being upgraded or downgraded is not removed from the cluster. The node remains a part of the cluster and serves traffic uninterrupted, except for the downtime when the node reboots after it is upgraded or downgraded.

However, due to software version mismatch among the cluster nodes, configuration propagation is disabled on the cluster. Configuration propagation is enabled only after all the cluster nodes are of the same version. Since configuration propagation is disabled during upgrading on downgrading a cluster, you cannot perform any configurations through the cluster IP address during this time.

> **Important**
>
> - In a cluster setup with maximum connection (maxConn) global parameter set to a non-zero value, CLIP connections might fail if any of the following conditions is met:
>
```
- Upgrading the setup from Citrix ADC 13.0 76.x build to
  Citrix ADC 13.0 79.x build.
- Restarting the CCO node in a cluster setup running Citrix
  ADC 13.0 76.x build.
```


```

Workarounds:

```
- Before upgrading a cluster setup from Citrix ADC 13.0 76.x build to Citrix ADC 13.0 79.x build, maximum connection (maxConn) global parameter must be set to zero. After upgrading the setup, you can set the maxConn parameter to a desired value and then save the configuration.
- Citrix ADC 13.0 76.x build is not suitable for cluster setups. Citrix recommends not to use the Citrix ADC 13.0 76.x build for a cluster setup.
```

- In a cluster setup, a Citrix ADC appliance might crash, when:

```
- upgrading the setup from Citrix ADC 13.0 47.x or 13.0 52.x build to a later build, or
- upgrading the setup to Citrix ADC 13.0 47.x or 13.0 52.x build
```

Workaround: During the upgrade process, perform the following steps:

```
- Disable all cluster nodes and then upgrade each cluster node.
- Enable all cluster nodes after all the nodes are upgraded.
```

Points to note before upgrading or downgrading the cluster

- You cannot add cluster nodes while upgrading or downgrading the cluster software version.
- You can perform node-level configurations through the NSIP address of individual nodes. Make sure to perform the same configurations on all the nodes to maintain them in sync.
- You cannot run the `start nstrace` command from the cluster IP address when the cluster is being upgraded. However, you can get the trace of individual nodes by performing this operation on individual cluster nodes using their NSIP address.
- Citrix ADC 13.0 76.x build is not suitable for cluster setups. Citrix recommends not to use the Citrix ADC 13.0 76.x build for a cluster setup.
- Citrix ADC 13.0 47.x and 13.0 52.x builds are not suitable for a cluster setup. It is because the inter-node communications are not compatible in these builds.
- When a cluster is being upgraded, it is possible that the upgraded nodes have some additional features activated that are unavailable on the nodes that are not yet upgraded. It results in a license mismatch warning while the cluster is being upgraded. This warning is automatically resolved when all the cluster nodes are upgraded.

> **Important**
>
> - Citrix recommends that you wait for the previous node to become active before upgrading or downgrading the next node.
> - Citrix recommends that the cluster configuration node must be upgraded/downgraded last to avoid multiple disconnects of cluster IP sessions.

To upgrade or downgrade the software of the cluster nodes

1. Make sure the cluster is stable and the configurations are synchronized on all the nodes.
2. Access each node through its NSIP address and perform the following:
   - Upgrade or downgrade the cluster node. For detailed information about upgrading and downgrading the software of an appliance, see Upgrade and downgrade a NetScaler appliance.
   - Save the configurations.
   - Reboot the appliance.
3. Repeat step 2 for each of the other cluster nodes.

Operations supported on individual cluster nodes

September 14, 2021

As a rule, Citrix ADC appliances that are a part of a cluster cannot be individually configured from their NSIP address. However, there are some operations that are an exception to this rule. These operations, when run from the NSIP address, are not propagated to other cluster nodes.

The operations are:

cluster instance (set	rm	enable	disable)

-

cluster node (set	rm)

-

ns trace (start	show	stop)

-

interface (set	enable	disable)

-

route (add	rm	set	unset)

-

ARP (add	rm	send -all)

-
- force cluster sync
- sync cluster files
- disable NTP sync
- save ns config
- reboot
- shutdown

For example, when you run the command `disable interface 1/1/1` from the NSIP address of a cluster node, the interface is disabled only on that node. Since the command is not propagated, the interface 1/1/1 remains enabled on all the other cluster nodes.

Support for heterogeneous cluster

September 14, 2021

Citrix ADC appliance supports a heterogeneous cluster in a cluster deployment. A heterogeneous cluster spans nodes of different Citrix ADC hardware and you can have a combination of different platforms in the same cluster.

> **Important**
>
> The formation or supportability of a heterogeneous cluster is possible and limited only to MPX hardware platforms.

The supportability and formation of the heterogeneous cluster depend on certain Citrix ADC models. The following table lists the platforms that are supported in the formation of a heterogeneous cluster, with an equal number of packet engines.

Number of Packet Engines	MPX Hardware Platforms	Supported MPX Hardware Platforms to form Heterogeneous Cluster
5	MPX 11500	MPX 14020
7	MPX 11515	MPX 14040
9	MPX 11530	MPX 14060

The following table lists the platforms that are supported in the formation of a heterogeneous cluster, with an unequal number of packet engines.

Hardware Platforms	Supported Hardware Platforms to form Heterogeneous Cluster
MPX 150XX	MPX 140XX

For more information on how to form a heterogeneous cluster deployment of Citrix ADC MPX appliances with the different number of packet engines across different SSL chipsets, see the **Heterogeneous cluster deployments** section in SSL offloading configuration.

> **Note**
>
> Before release 13.0 build 47.x, if you run the “join cluster” command from the node that has an unequal number of packet engines, the following error message appears: “Mismatch in the number of active PPEs between CCO and local node”.

Points to note

1. The extra management CPU setting must be the same on all the cluster nodes.
2. The newly added node must have the same capacity on the data planes and backplane, as that of existing cluster nodes.

3. If there are mixed platform devices supporting different ciphers, then the cluster would agree upon a common cipher list.

FAQs

September 14, 2021

A list of the FAQ about clustering.

How many Citrix ADC appliances can be included in a single Citrix ADC cluster?

A Citrix ADC cluster can include one appliance or as many as 32 Citrix ADC nCore hardware or virtual appliances. Each of these nodes must satisfy the criteria specified in Prerequisites for Cluster Nodes.

Can a Citrix ADC appliance be a part of multiple clusters?

No. A Citrix ADC appliance can belong to one cluster only.

What is a cluster IP address? What is its subnet mask?

The cluster IP address is the management address of a Citrix ADC cluster. All cluster configurations must be performed by accessing the cluster through this address. The subnet mask of the cluster IP address is fixed at 255.255.255.255.

How can I make a specific cluster node as the cluster configuration coordinator?

To manually set a specific node as the cluster configuration coordinator, you must set the priority of that node to the lowest numeric value (highest priority). To understand, let us consider a cluster with three nodes that have the following priorities:

n1 - 29, n2 - 30, n3 - 31

Here, n1 is the configuration coordinator. If you want to make n2 the configuration coordinator, you must set its priority to a value that is lower than n1, for example, 28. On saving the configuration, n2 becomes the configuration coordinator.

> **Note**
>
> n2 with its original priority value of 30 becomes the configuration coordinator when n1 goes down. The node with the next lowest priority value is selected in case the configuration coordinator goes down.

Why are the network interfaces of a cluster represented in 3-tuple (n/u/c) notation instead of the regular 2-tuple (u/c) notation?

When a Citrix ADC appliance is part of a cluster, you must be able to identify the node to which the interface belongs. Therefore, the network interface naming convention for cluster nodes is modified from u/c to n/u/c, where n denotes the node Id.

How can I set the host name for a cluster node?

The host name of a cluster node must be specified by running the **set ns hostname** command through the cluster IP address. For example, to set the host name of the cluster node with ID 2, the command is:

set ns hostname hostName1 -**ownerNode** 2

Can I automatically detect Citrix ADC appliances so that I can add them to a cluster?

Yes. The configuration utility allows you to discover appliances that are present in the same subnet as the NSIP address of the configuration coordinator. For more information, see Discovering NetScaler Appliances.

Is the traffic serving capability of a cluster affected if a node is removed or disabled or reboot or shutdown or made inactive?

Yes. When any of these operations are performed on an active node of the cluster, the cluster has one less node to serve traffic. Also, existing connections on this node are terminated.

I have multiple standalone appliances, each of which has different configurations. Can I add them to a single cluster?

Yes. You can add appliances that have different configurations to a single cluster. However, when the appliance is added to the cluster, the existing configurations are cleared. To use the configurations that are available on each of the individual appliances, you must:

1. Create a single *.conf file for all the configurations.
2. Edit the configuration file to remove features that are not supported in a cluster environment.
3. Update the naming convention of interfaces from 2-tuple (u/c) format to 3-tuple (n/u/c) format.
4. Apply the configurations to the configuration coordinator node of the cluster by using the batch command.

Can I migrate the configurations of a standalone Citrix ADC appliance or an HA setup to the clustered setup?

No. When a node is added to a clustered setup, its configurations are implicitly cleared by using the **clear ns config** command (with the **extended** option). In addition, the SNIP addresses and all VLAN configurations (except default VLAN and NSVLAN) are cleared. Therefore, it is recommended that you back up the configurations before adding the appliance to a cluster. Before using the backed-up configuration file for the cluster, you must:

1. Edit the configuration file to remove features that are not supported in a cluster environment.
2. Update the naming convention of interfaces from two-tuple (x/y) format to three-tuple (x/y/z) format.
3. Apply the configurations to the configuration coordinator node of the cluster by using the **batch** command.

Is backplane interfaces part of the L3 VLANs?

Yes, by default, backplane interfaces have presence on all the L3 VLANs that are configured on the cluster.

How can I configure a cluster that includes nodes from different networks?

> **Note**
>
> Supported from NetScaler 11.0 onwards.

A cluster that includes nodes from different networks is called a L3 cluster (sometimes referred to as a cluster in INC mode). In an L3 cluster, all nodes that belong to a single network must be grouped in a single node group. Therefore, if a cluster includes two nodes each from three different networks, you have to create 3 node groups (one for each network) and associate each of these node groups with the nodes that belong to that network. For configuration information, see the steps to set up a cluster.

How can I configure/un-configure the NSVLAN on a cluster?

Do either one of the following:

- To make the NSVLAN available in a cluster, make sure that each appliance has the same NSVLAN configured before it is added to a cluster.
- To remove the NSVLAN from a cluster node, first remove the node from the cluster and then delete the NSVLAN from the appliance.

I have a cluster set up where some Citrix ADC nodes are not connected to the external network. Can the cluster still function normally?

Yes. The cluster supports a mechanism called linksets, which allows unconnected nodes to serve traffic by using the interfaces of connected nodes. The unconnected nodes communicate with the connected nodes through the cluster backplane. For more information, see Using Linksets.

How can deployments that require MAC-Based Forwarding (MBF) be supported in a clustered setup?

Deployments that use MBF must use linksets. For more information, see Using Linksets.

Can I run commands from the NSIP address of a cluster node?

No. Access to individual cluster nodes through the NSIP addresses is read-only. Therefore, when you log on to the NSIP address of a cluster node you can only view the configurations and the statistics. You cannot configure anything. However, there are some operations you can run from the NSIP address of a cluster node. For more information, see Operations Supported on Individual Nodes.

Can I disable configuration propagation among cluster nodes?

No, you cannot explicitly disable the propagation of cluster configurations among cluster nodes. However, during a software upgrade or downgrade, a version mismatch error can automatically disable configuration propagation.

Can I change the NSIP address or change the NSVLAN of a Citrix ADC appliance when it is a part of the cluster?

No. To make such changes you must first remove the appliance from the cluster, perform the changes, and then add the appliance to the cluster.

Does the Citrix ADC cluster support L2 and L3 VLANs?

Yes. A cluster supports VLANs between cluster nodes. The VLANs must be configured on the cluster IP address.

- **L2 VLAN.** You can create a layer2 VLAN by binding interfaces that belong to different nodes of the cluster.
- **L3 VLAN.** You can create a layer3 VLAN by binding IP addresses that belong to different nodes of the cluster. The IP addresses must belong to the same subnet. Make sure that one of the following criteria is satisfied. Otherwise, the L3 VLAN bindings can fail.

- All nodes have an IP address on the same subnet as the one bound to the VLAN.
- The cluster has a striped IP address and the subnet of that IP address is bound to the VLAN.

When you add a node to a cluster that has only spotted IPs, the sync happens before spotted IP addresses are assigned to that node. In such cases, L3 VLAN bindings can be lost. To avoid this loss, either add a striped IP or add the L3 VLAN bindings on the NSIP of the newly added node.

How can I configure SNMP on a Citrix ADC cluster?

SNMP monitors the cluster, and all the nodes of the cluster, in the same way that it monitors a standalone appliance. The only difference is that SNMP on a cluster must be configured through the cluster IP address. When generating hardware specific traps, two more varbinds are included to identify the node of the cluster: node ID and NSIP address of the node.

What details must I have available when I contact technical support for cluster-related issues?

The Citrix ADC appliance provides a **show techsupport -scope cluster** command that extracts configuration data, statistical information, and logs of all the cluster nodes. Run this command on the cluster IP address.

The output of this command is saved in a file named *collector_cluster_<nsip_CCO>_P_<date-timestamp>.tar.gz* which is available in the */var/tmp/support/cluster/* directory of the configuration coordinator.

Send this archive to the technical support team to debug the issue.

Can I use striped IP addresses as the default gateway of servers?

In cluster deployments, make sure the default gateway of the server points to a striped IP address (if you are using a Citrix ADC-owned IP address). For example, in the case of LB deployments with USIP enabled, the default gateway must be a striped SNIP address.

Can I view the routing configurations of a specific cluster node from the cluster IP address?

Yes. You can view and clear the configurations specific to a node by specifying the owner node while entering the VTYSH shell.

For example, to view the output of a command on nodes 0 and 1, the command is as follows:

```
1 \> vtysh
2 ns# owner-node 0 1
```



```
ns(node-0 1)\# show cluster state
ns(node-0 1)\# exit-cluster-node
ns\#
```

How can I specify the node for which I want to set the LACP system priority?

> **Note**
>
> Supported from NetScaler 10.1 onwards.

In a cluster, you must set that node as the owner node by using the **set lacp** command.

For example: To set the LACP system priority for a node with ID 2:

```
set lacp -sysPriority 5 -ownerNode 2<!--NeedCopy-->
```

How are IP tunnels configured in a cluster setup?

> **Note**
>
> Supported from NetScaler 10.1 onwards.

Configuring IP tunnels in a cluster is the same as on a standalone appliance. The only difference is that in a cluster setup, the local IP address must be a striped SNIP address.

How can I add a failover interface set (FIS) on the nodes of a Citrix ADC cluster?

> **Note**
>
> Supported from NetScaler 10.5 onwards.

On the cluster IP address, specify the ID of the cluster node on which the FIS must be added, using the command as follows:

```
add fis <name> -ownerNode <nodeId>
```

> **Notes**
>
> - The FIS name for each cluster node must be unique.
> - A cluster LA channel can be added to a FIS. You make sure that the cluster LA channel has a local interface as a member interface.

For more information on FIS, see Configuring failover interface set.

How are net profiles configured in a cluster setup?

> **Note**
>
> Supported from NetScaler 10.5 onwards.

You can bind spotted IP addresses to a net profile. This net profile can then be bound to a spotted load balancing virtual server or service (that is defined using a node group). The following recommendations must be followed, failing which, the net profile configurations are not honored and the USIP/USNIP settings are used:

> **Note**
>
> - If the **strict** parameter of the node group is set to **Yes**, the net profile must contain a minimum of one IP address from each node group member.
> - If the **strict** parameter of the node group is set to **No**, the net profile must include at least one IP address from each of the cluster nodes.

How can I configure WIonNS in a cluster setup?

> **Note**
>
> Supported from NetScaler 11.0 Build 62.x onwards.

To use WIonNS on a cluster, you must do the following:

1. Make sure that the Java package and the WI package are present in the same directory on all the cluster nodes.
2. Create a load balancing virtual server that has persistency configured.
3. Create services with IP addresses as the NSIP address of each of the cluster nodes that you want to serve WI traffic. This step can only be configured using the Citrix ADC CLI.
4. Bind the services to the load balancing virtual server.

> **Note**
>
> If you are using WIonNS over a VPN connection, make sure that the load balancing virtual server is set as WIHOME.

Can the cluster LA channel be used for management access?

No. Management access to a cluster node, must not be configured on a cluster LA channel (for example, CLA/1) or its member interfaces. It is because when the node is INACTIVE, the corresponding cluster LA interface is marked as power down, and therefore looses management access.

How cluster nodes communicate to each other and what are the different types of traffic that goes through the backplane?

A backplane is a set of interfaces in which one interface of each node is connected to a common switch, which is called the cluster backplane switch. The different types of traffic that goes through a backplane, which is used by internode communication are:

- Node to Node Messaging (NNM)
- Steered traffic
- Configuration propagation and synchronization

Each node of the cluster uses a special MAC cluster backplane switch address to communicate with other nodes through the backplane. The cluster special MAC is of the form: **0x02 0x00 0x6F <cluster_id> <node_id> <reserved>**, where <cluster_id> is the cluster instance ID. The <node_id> is the node number of the Citrix ADC appliance that is added to a cluster.

> **Note**
>
> The amount of traffic that is handled by a backplane has negligible CPU overhead.

What gets routed over the GRE tunnel for the Layer 3 cluster?

Only steered data traffic goes over the GRE tunnel. The packets are steered through the GRE tunnel to the node on the other subnet.

How Node to Node Messaging (NNM) and heartbeat messages are exchanged, and how are they routed?

The NNM, heartbeat messages, and cluster protocol are non-steering traffic. These messages are not sent through the tunnel, but they are routed directly.

What are the MTU recommendations when Jumbo frames are enabled for layer 3 cluster tunneled traffic?

The following are the layer 3 cluster recommendations of Jumbo MTU over GRE tunnel:

- The Jumbo MTU can be configured among cluster nodes across the L3 path to accommodate GRE tunnel overhead.
- The fragmentation does not happen for full sized packets which must be steered.
- Steering of traffic continues to work even if Jumbo frames are not allowed, but with more overhead due to fragmentation.

How the global hash key is generated and shared across all nodes?

The `rsskey` for a standalone appliance is generated at the boot time. In a cluster setup, the first node holds the `rsskey` of the cluster. Every new node joining to the cluster synchronizes the `rsskey`.

What is the need of `set rsskeytype -rsskey symmetric` command for *:*, USIP on, `useproxyport` off, topologies?

It is not specific to a cluster, applies to a standalone appliance as well. With USIP ON, and use proxy port disabled, symmetric `rsskey` reduces both Core to Core (C2C) steering and node to node steering.

What are the factors that contribute to change the CCO node?

The first node added to form a cluster setup becomes the configuration coordinator (CCO) node. The following factors contribute to change the CCO node in the cluster setup:

- When the current CCO node is removed from the cluster setup
- When the current CCO node crashes
- When the priority of the non-CCO node is changed (lower priority has higher precedence)
- In dynamic conditions like, network reachability between the nodes
- When there is change in node states – active, spare, and passive. Active nodes are preferred as CCO.
- When there is a change in configuration, and the node having the latest configuration is preferred as CCO.

Troubleshooting the Citrix ADC cluster

September 14, 2021

If a failure occurs in a Citrix ADC cluster, the first step in troubleshooting is to get information on the cluster instance. You can get the information by running the `show cluster instance clId` and `show cluster node nodeId` commands on the cluster nodes respectively.

If you are not able to find the issue by using the above two approaches, you can use one of the following:

- **Isolate the source of the failure.** Try bypassing the cluster to reach the server. If the attempt is successful, the problem is probably with the cluster setup.
- **Check the commands recently executed.** Run the history command to check the recent configurations performed on the cluster. You can also review the ns.conf file to verify the configurations that have been implemented.

- **Check the ns.log files.** Use the log files, available in the /var/log/ directory of each node, to identify the commands run, the status of commands, and the state changes.
- **Check the newnslog files.** Use the `newnslog` files, available in the /var/nslog/ directory of each node, to identify the events that have occurred on the cluster nodes. You can view multiple `newnslog` files as a single file, by copying the files to a single directory, and then running the following command:

```
nsconmsg -K newnslog-node<id> -K     newnslog.node<id> -d current
```

If you still cannot resolve the issue, you can try tracing the packets on the cluster or use the show `techsupport -scope cluster` command. You can use the command to send the report to the technical support team.

Tracing the packets of a Citrix ADC cluster

September 14, 2021

The Citrix ADC operating system provides a utility called *ns trace* to get a dump of the packets that are received and sent out by an appliance. The utility stores the packets in trace files. You can use these files to debug problems in the flow of packets to the cluster nodes. The trace files must be viewed with the Wireshark application.

Some salient aspects of the ns trace utility are:

- Can be configured to trace packets selectively by using classic expressions and default expressions.
- Can capture the trace in multiple formats: ns trace format (.cap) and TCP dump format (.pcap).
- Can aggregate the trace files of all cluster nodes on the configuration coordinator.
- Can merge multiple trace files into a single trace file (only for.cap files).

You can use the ns trace utility from the Citrix ADC command line or the Citrix ADC shell.

To trace packets of a standalone appliance

Run the start ns trace command on the appliance. The command creates trace files in the /var/nstrace/<date-timestamp> directory. The trace file names are of the form nstrace<id\ >.cap.

You can view the status by running the show ns trace command. You can stop tracing the packets by running the stop ns trace command.

> **Note**
>
> You can also run the ns trace utility from the Citrix ADC shell by running the nstrace.sh file. However, it is recommended that you use the ns trace utility through the Citrix ADC command line interface.

To trace packets of a cluster

You can trace the packets on all the cluster nodes and obtain all the trace files on the configuration coordinator.

Run the start ns trace command on the cluster IP address. The command is propagated and run on all the cluster nodes. The trace files are stored in individual cluster nodes in the /var/nstrace/<date-timestamp> directory. The trace file names are of the form nstrace<id>_node<id\ >.cap.

You can use the trace files of each node to debug the nodes operations. But if you want the trace files of all cluster nodes in one location, you must run the stop ns trace command on the cluster IP address. The trace files of all the nodes are downloaded on the cluster configuration coordinator in the /var/nstrace/<date-timestamp> directory as follows:



Merge multiple trace files

You can prepare a single file from the trace files (supported only for. Cap files) obtained from the cluster nodes. The single trace files give you a cumulative view of the trace of the cluster packets. The trace entries in the single trace file are sorted based on the time the packets were received on the cluster.

To merge the trace files, at the Citrix ADC shell, type:

```
1 > nstracemerge.sh -srcdir \<DIR\> -dstdir \<DIR\> -filename \<name\> -
    filesize \<num\>
```

Where,

- `srcdir` is the directory from which the trace files are merged. All trace files within this directory are merged into a single file.
- `dstdir` is the directory where the merged trace file is created.
- `Filename` is the name of the trace file that is created.

- `Filesize` is the size of the trace file.

Examples

Following are some examples of using the ns trace utility to filter packets.

- To trace the packets on the backplane interfaces of three nodes:

  Using classic expressions:

```
1   > start nstrace -filter "INTF == 0/1/1 && INTF == 1/1/1 && INTF
      == 2/1/1"
```

  Using default expressions:

```
1   > start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") &&
      CONNECTION.INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
```

- To trace the packets from a source IP address 10.102.34.201 or from a system whose source port is greater than 80 and the service name is not “s1”:

  Using classic expressions

```
1   > start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME !=
      s1 && SOURCEPORT > 80)"
```

  Using default expressions

```
1   > start nstrace -filter "CONNECTION.SRCIP.EQ(10.102.34.201) || (
      CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.GT(80))"
```

> **Note**
>
> For more information about filters used in ns trace, see ns trace.

Capturing SSL Session Keys During a Trace

When you run the “start ns trace” command, you can set the new `capsslkeys` parameter to capture the SSL master keys for all SSL sessions. If you include this parameter, a file named nstrace.sslkeys is generated along with the packet trace. This file can be imported into Wireshark to decrypt the SSL traffic in the corresponding trace file.

This functionality is similar to web browsers exporting session keys that can later be imported into Wireshark for decrypting SSL traffic.

Advantages of using SSL session keys

Following are the advantages of using SSL session keys:

1. Generates smaller trace files that do not include the extra packets created by the SSLPLAIN mode of capturing.
2. Provides the ability to view plaintext [SP(1] from the trace and choose whether to share the master keys file or protect sensitive data by not sharing it.

Limitations of using SSL session keys

Following are the limitations of using SSL session keys:

1. SSL sessions cannot be decrypted if the initial packets of the session are not captured.
2. SSL sessions cannot be captured if the Federal Information Processing Standard (FIPS) mode is enabled.

To capture SSL session keys by using the command line interface (CLI)

At the command prompt, type the following commands to enable or disable SSL session keys in a trace file and verify trace operation.

```
1  > start nstrace -capsslkeys ENABLED
2  > show nstrace
3  Example
4  > start nstrace -capsslkeys ENABLED
5  > show nstrace
6        State:  RUNNING          Scope:  LOCAL            TraceLocation:
             "/var/nstrace/04May2016_17_51_54/..."
7        Nf:  24                  Time:  3600              Size:  164
                        Mode:  TXB NEW_RX
8        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  04
            May2016_17_51_54 Link:  DISABLED
9        Merge:  ONSTOP           Doruntimecleanup:  ENABLED TraceBuffers:
              5000      SkipRPC:  DISABLED
10       SkipLocalSSH:  DISABLED  Capsslkeys:  ENABLED     InMemoryTrace:
             DISABLED
11   Done
```

To configure SSL session keys by using the Citrix ADC GUI

1. Navigate to **Configuration > System > Diagnostics > Technical Support Tools** and click **Start new Trace** to start tracing encrypted packets on an appliance.

2. On the **Start Trace** page, select the **Capture SSL Master Keys** check box.
3. Click **OK** and **Done**.

To import the SSL Master Keys into Wireshark

On the Wireshark GUI, navigate to **Edit > Preferences > Protocols > SSL > (Pre)-Master-Secret log filename** and specify the master key files obtained from the appliance.

Troubleshooting common issues

September 14, 2021

While joining a node to the cluster, I get the following message, "ERROR: Invalid interface name/number." What must I do to resolve this error?

The said error occurs if you provided an invalid or incorrect backplane interface while using the add cluster node command to add the node. To resolve this error, verify the interface you provided while adding the node. Make sure that you have not specified the appliance's management interface as the backplane interface, and that the <nodeId> bit of the interface is the same as the node's Id. For example, if the nodeId is 3, the backplane interface must be 3/<c>/<u>.

While joining a node to the cluster, I get the following message, "ERROR: Clustering cannot be enabled, because the local node is not a member of the cluster." What must I do to resolve this error?

This error occurs when you try to join a node without adding the node's NSIP to the cluster. To resolve this error, you must first add the node's NSIP address to the cluster by using the **add cluster node** command and then run the **join cluster** command.

While joining a node to the cluster, I get the following message, "ERROR: Connection refused." What must I do to resolve this error?

This error can occur due to the following reasons:

- **Connectivity problems.** The node cannot connect to the cluster IP address. Try pinging the cluster IP address from the node that you are trying to join.
- **Duplicate cluster IP address.** Check to see if the cluster IP address exists on some non-cluster node. If it does, create a cluster IP address and try rejoining the cluster.

While joining a node to the cluster, I get the following message, “ERROR: License mismatch between the configuration coordinator and the local node.” What must I do to resolve this error?

The appliance that you are joining to the cluster must have the same licenses as the configuration coordinator. This error occurs when the licenses on the node you are joining do not match the licenses on the configuration coordinator. To resolve this error, run the following commands on both the nodes and compare the outputs.

From the command line:

- `show ns hardware`
- `show ns license`

From the shell:

- `nsconmsg -g feature -d stats`
- `ls /nsconfig/license`
- View the contents of the /var/log/license.log file

What must I do when the configurations of a cluster node are not in synchronize with the cluster configurations?

Usually, the configurations are automatically synchronized between all the cluster nodes. However, if you feel that the configurations are not synchronized on a specific node, you must force the synchronization by running the force cluster sync command from the node that you want to synchronize. For more information, see Synchronizing Cluster Configurations.

When configuring a cluster node, I get the following message, “ERROR: Session is read-only; connect to the cluster IP address to modify the configuration.”

All configurations on a cluster must be done through the cluster IP address and the configurations are propagated to the other cluster nodes. All sessions established through the NSIP address of individual nodes are read-only.

Why does the node state show “INACTIVE” when the node health shows “UP”?

A healthy node can be in the INACTIVE state for various reasons. A scan of the ns.log or error counters can help you determine the exact reason.

How can I resolve the health of a node when its health shows “NOT UP”?

Node health “**Not UP**” indicates that there are some issues with the node. To know the root cause, you must run the **show cluster node** command. This command displays the node properties and the

reason for the node failure.

What must I do when the health of a node shows as "NOT UP" and the reason indicates that configuration commands have failed on a node?

This issue arises when some commands are not run on the cluster nodes. In such cases, you must make sure that the configurations are synchronized using one of the following options:

- If some of the cluster nodes are in this state, you must perform the force cluster synchronization operation on those nodes. For more information, see Synchronizing Cluster Configurations.
- If all cluster nodes are in this state, you must disable and then enable the cluster instance on all the cluster nodes.

When I run the set virtual server command, I get the following message, "No such resource." What must I do to resolve this issue?

The **set vserver** command is not supported in clustering. The **unset vserver, enable vserver, disable vserver,** and **rm vserver** commands are also not supported. However, the **show vserver** command is supported.

I cannot configure the cluster over a Telnet session. What must I do?

Over a telnet session, the cluster IP address can be accessed only in read-only mode. Therefore, you cannot configure a cluster over a telnet session.

I notice a significant time difference across the cluster nodes. What must I do to resolve this issue?

When PTP packets are dropped due to the backplane switch or if the physical resources are over-committed in a virtual environment, the time will not get synchronized.

To synchronize the times, you must do the following on the cluster IP address:

1. Disable PTP.

   set ptp -state disable

2. Configure Network Time Protocol (NTP) for the cluster. For more information, see Setting up Clock Synchronization.

What must I do, if there is no connectivity to the cluster IP address and the NSIP address of a cluster node?

If you cannot access to the cluster IP address or the NSIP of a cluster node, you must access the appliance through the serial console. If the NSIP address is reachable, you can SSH to the cluster IP address from the shell by running the following command at the shell prompt:

"'# ssh nsroot@

```
## What must I do to recover a cluster node that has connectivity
   issues?

To recover a node that has connectivity issues:

1.  Disable the cluster instance on that node (since you cannot run
   commands from the NSIP of a cluster node).

1.  Run the commands required to recover the node.

1.  Enable the cluster instance on that node.

## Some nodes of the cluster have two default routes. How can I remove
   the second default route from the cluster node?

To delete the additional default route, do the following on each node
   that has the extra route:

1.  Disable the cluster instance.

    ```disable cluster instance <clId><!--NeedCopy-->
```

1. Remove the route.

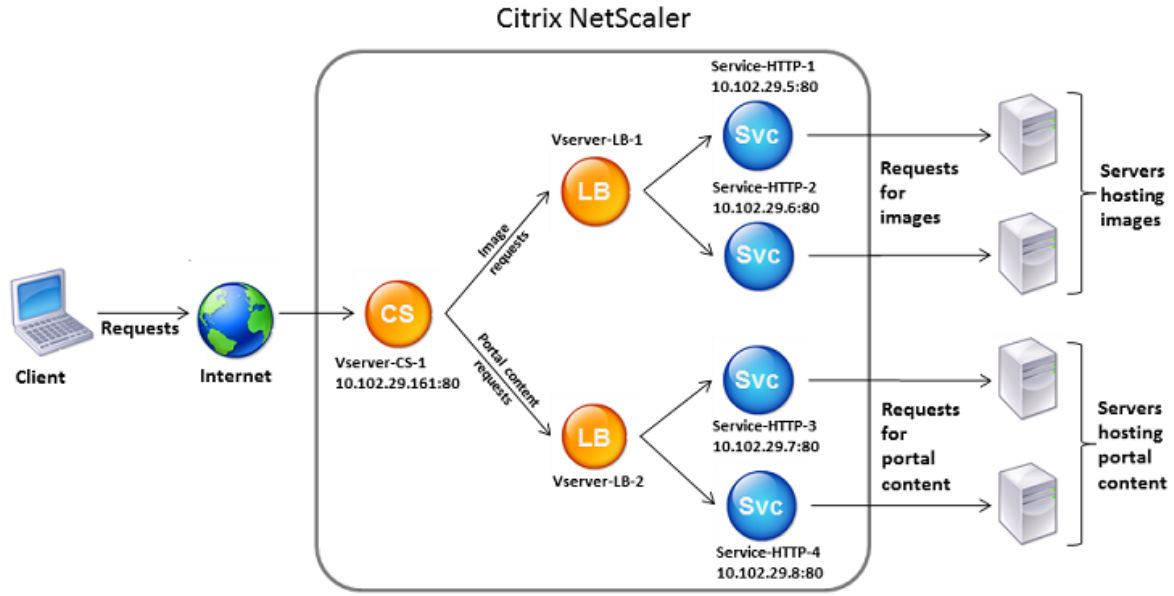
   ```
   rm route <network> <netmask> <gateway><!--NeedCopy-->
   ```

2. Enable the cluster instance.

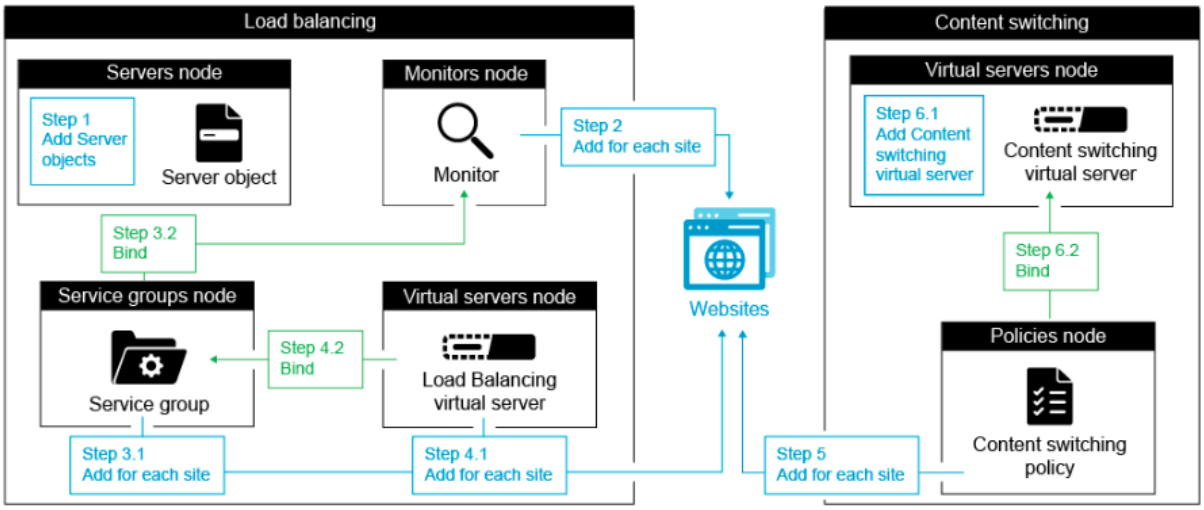
   ```
   enable cluster instance <clId><!--NeedCopy-->
   ```

The cluster functionality gets affected when an existing cluster node comes online. What must I do to resolve this issue?

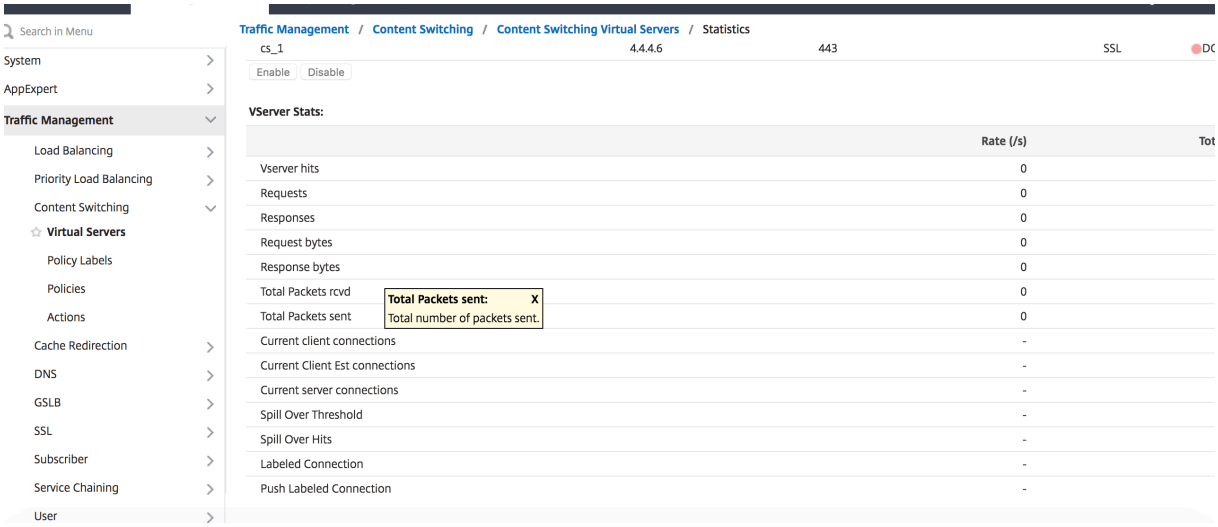
If the RPC password of a node is changed from the cluster IP address when that node is out of the cluster, then, when the node comes online, there is a mismatch in RPC credentials and can affect cluster functionality. To solve this issue, use the set ns rpcNode command to update the password on the NSIP of the node which has come online.


   ```

# Content Switching

September 14, 2021

In today's complex websites, you might want to present different content to different users. For example, you might want to allow users from the IP range of a customer or partner to have access to a special Web portal. You might want to present content relevant to a specific geographical area to users from that area. You might want to present content in different languages to the speakers of those languages. You might want to present content tailored to specific devices, such as smartphones, to them who use the devices. The Citrix ADC content switching feature enables the appliance to distribute client requests across multiple servers based on specific content that you want to present to those users.

To configure content switching, first create a basic content switching setup, and then customize it to meet your needs. This entails enabling the content switching feature, setting up load balancing for the server or servers that host each version of the content that is being switched, creating a content switching virtual server, creating policies to choose which requests are directed to which load balancing virtual server, and binding the policies to the content switching virtual server. You can then customize the setup to meet your needs by setting precedence for your policies, protecting your setup by configuring a backup virtual server, and improving the performance of your setup by redirecting requests to a cache.

## How Content Switching Works

Content Switching enables the Citrix ADC appliance to direct requests sent to the same Web host to different servers with different content. For example, you can configure the appliance to direct requests for dynamic content (such as URLs with a suffix of .asp, .dll, or .exe) to one server and requests for static content to another server. You can configure the appliance to perform content switching based on TCP/IP headers and payload.

You can also use content switching to configure the appliance to redirect requests to different servers with different content based on various client attributes. Some of those client attributes are:

- **Device Type.** The appliance examines the user agent or custom HTTP header in the client request for the type of device from which the request originated. Based on the device type, it directs the request to a specific Web server. For example, if the request came from a cell phone, the request is directed to a server, capable of serving content that the user can view on the cell phone. A request from a computer is directed to a different server, capable of serving content designed for a computer screen.
- **Language.** The appliance examines the Accept-Language HTTP header in the client request and determines the language used by the client's browser. The appliance then sends the request to a server that serves content in that language. For example, using content switching based on

 

language, the appliance can send someone whose browser is configured to request content in French to a server with the French version of a newspaper. It can send someone else whose browser is configured to request content in English to a server with the English version.

- **Cookie.** The appliance examines the HTTP request headers for a cookie that the server set previously. If it finds the cookie, it directs requests to the appropriate server, which hosts custom content. For example, if a cookie is found that indicates that the client is a member of a customer loyalty program, the request is directed to a faster server or one with special content. If it does not find a cookie, or if the cookie indicates that the user is not a member, the request is directed to a server for the general public.
- **HTTP Method.** The appliance examines the HTTP header for the method used, and sends the client request to the right server. For example, GET requests for images can be directed to an image server, while POST requests can be directed to a faster server that handles dynamic content.
- **Layer 3/4 Data.** The appliance examines requests for the source or destination IP, source or destination port, or any other information present in the TCP or UDP headers, and directs the client request to the right server. For example, requests from source IPs that belong to customers can be directed to a custom web portal on a faster server, or one with special content.

A typical content switching deployment consists of the entities described in the following diagram.

Figure 1. Content Switching Architecture




A content switching configuration consists of a content switching virtual server, a load balancing setup consisting of load balancing virtual servers and services, and content switching policies. To configure content switching, you must configure a content switching virtual server and associate it with policies

 

and load balancing virtual servers. This process creates a *content group*—a group of all virtual servers and policies involved in a particular content switching configuration.

Content switching can be used with HTTP, HTTPS, TCP, and UDP connections. For HTTPS, you must enable SSL Offload.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. The priority of the policy defines the order in which the policies bound to the content switching virtual server are evaluated. If you are using default syntax policies, when you bind a policy to the content switching virtual server, you must assign a priority to that policy. If you are using Citrix ADC classic policies, you can assign a priority to your policies, but are not required to do so. If you assign priorities, the policies are evaluated in the order that you set. If you do not, the Citrix ADC appliance evaluates your policies in the order in which they were created.

In addition to configuring policy priorities, you can manipulate the order of policy evaluation by using Goto expressions and policy bank invocations. For more details about default syntax policy configuration, see Configuring Default Syntax Policies.

After it evaluates the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

Content switching virtual servers can only send requests to other virtual servers. If you are using an external load balancer, you must create a load balancing virtual server for it and bind its virtual server as a service to the content switching virtual server.

## Configuring basic content switching

September 14, 2021

Before you configure content switching, you must understand how content switching is set up and how the services and virtual servers are connected.

To configure a basic, functional content switching setup, first enable the content switching feature. Then, create at least one content group. For each content group, create a content switching virtual server to accept requests to a group of websites that use content switching. Also create a load balancing setup, which includes a group of load balancing virtual servers to which the content switching virtual server directs requests. To specify which requests to direct to which load balancing virtual server, create at least two content switching policies, one for each type of request that is to be redirected. When you have created the virtual servers and policies, bind the policies to the content switching virtual server. You can also bind a policy to multiple content switching virtual servers. When you bind a policy, you specify the load balancing virtual server to which requests that match the policy are to be directed.

 

In addition to binding individual policies to a content switching virtual server, you can bind policy labels. If you create more content groups, you can bind a policy or policy label to more than one of the content switching virtual servers.

> **Note**
>
> After creating a content group, you can modify its content switching virtual server to customize the configuration.

### Enabling content switching

To use the content switching feature, you must enable content switching. You can configure content switching entities even though the content switching feature is disabled. However, the entities will not work.

### To enable content switching by using the CLI

At the command prompt, type the following commands to enable content switching and verify the configuration:

```
enable ns feature CS

show ns feature
<!--NeedCopy-->
```

Example:

```
> enable  feature ContentSwitch
    Done
> show feature

    Feature                       Acronym             Status
    -------                       -------             ------
1)     Web Logging                   WL                  OFF
2)     Surge Protection              SP                  ON
3)     Load Balancing                LB                  ON
4)     Content Switching             CS                  ON
.
.
.
22)    Responder                     RESPONDER           ON
23)    HTML Injection                HTMLInjection       ON
24)    NetScaler Push                push                OFF
Done
```



```
<!--NeedCopy-->
```

To enable content switching by using the GUI

Navigate to **System > Settings** and, in the **Modes and Features** group, select **Configure Basic Features**, and select **Content Switching**.

The following figure illustrates the step wise configuration of Content Switching.



Creating content switching virtual servers

You can add, modify, and remove content switching virtual servers. The state of a virtual server is DOWN when you create it, because the load balancing virtual server is not yet bound to it.

To create a virtual server by using the CLI

At the command prompt, type:

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

Example:

```
add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80
<!--NeedCopy-->
```


To add a content switching virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and add a virtual server.
2. Specify a name for content switching virtual server.

   > **Note**
   >
   > There is different Content Switching virtual server for each protocol. (For example, HTTP and SSL).

3. Populate the relevant fields and click **OK**.

Content switching virtual server statistics

The content switching virtual server statistics display information such as virtual server select, request bytes, response bytes, total packets received, total packets sent, spillover threshold, spillover select, current client established connections, and virtual server down backup select.

The content switching virtual server statistics also display the summary details of the bound default load balancing virtual server.

To view statistics of content switching virtual server by using the CLI

At the command prompt, type:

```
stat cs vserver <name>
<!--NeedCopy-->
```

Example:

```
stat cs vserver CS_stats
<!--NeedCopy-->
```



```
Vserver Summary
                                IP  port     Protocol        State
CS_stats                   1.1.1.1    80         HTTP           UP

VServer Stats:
                                                  Rate (/s)                Total
Vserver hits                                              0                    0
Requests                                                  0                    0
Responses                                                 0                    0
Request bytes                                             0                    0
Response bytes                                            0                    0
Total Packets rcvd                                        0                    0
Total Packets sent                                        0                    0
Current client connections                               --                    0
Current Client Est connections                           --                    0
Current server connections                               --                    0
Spill Over Threshold                                     --                    0
Spill Over Hits                                          --                    0
Labeled Connection                                       --                    0
Push Labeled Connection                                  --                    0
Deferred Request                                          0                    0
Invalid Request/Response                                 --                    0
Invalid Request/Response Dropped                         --                    0
Vserver Down Backup Hits                                 --                    0
Current Multipath TCP sessions                           --                    0
Current Multipath TCP subflows                           --                    0
Apdex for client response times.                         --                 1.00
Average client TTLB                                      --                    0

 Done
```

To view statistics of content switching virtual server by using the GUI

1. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**.
2. Select the virtual server and click **Statistics**.



Configuring a load balancing setup for content switching

The content switching virtual server redirects all requests to a load balancing virtual server. You must create one load balancing virtual server for each version of the content that is being switched. It is true even when your setup has only one server for each version of the content, and you are therefore not doing any load balancing with those servers. You can also configure actual load balancing with multiple load-balanced servers that mirror each version of the content. In either scenario, the content switching virtual server needs to have a specific load balancing virtual server assigned to each version of the content that is being switched.

The load balancing virtual server then forwards the request to a service. If it has only one service bound to it, it selects that service. If it has multiple services bound to it, it uses its configured load balancing method to select a service for the request, and forwards that request to the service that it selected.

To configure a basic load balancing setup, you need to perform the following tasks:

- Create load balancing virtual servers
- Create services
- Bind services to the load balancing virtual server

For more information on load balancing, see How load balancing works. For detailed instructions on setting up a basic load balancing configuration, see Set up basic load balancing.

Configuring a content switching action

You specify the target load balancing virtual server for a content switching policy when binding the policy to the content switching virtual server. Therefore, you have to configure one policy for each load balancing virtual server to which to direct traffic.

However, if your content switching policy uses a default syntax rule, you can configure an action for the policy. In the action, you can specify the name of the target load balancing virtual server, or you can configure a request-based expression that, at run time, computes the name of the load balancing virtual server to which to send the request. The action expression must be specified in the default syntax.

The expression option can drastically reduce the size of your content switching configuration, because you need only one policy per content switching virtual server. Content switching policies that use an action can also be bound to multiple content switching virtual servers, because the target load balancing virtual server is no longer specified in the content switching policy. The ability to bind a single policy to multiple content switching virtual servers helps to further reduce the size of your content switching configuration.

After you create an action, you create a content switching policy and specify the action in the policy, so that the action is performed when that policy matches a request.

> **Note**
>
> You can also, for a content switching policy that uses a default syntax rule, specify the target load balancing virtual server when binding the policy to a content switching virtual server, instead of using a separate action. For domain-based policies, URL-based policies, and rule based policies that use classic expressions, an action is not available. So, for these types of policies, you specify the name of the target load balancing virtual server when binding the policy to a content switching virtual server.

Configuring an action that specifies the name of the target load balancing virtual server

If you choose to specify the name of the target load balancing virtual server in a content switching action, you need as many content switching policies as you have target load balancing virtual servers. Content switching decisions, in this case, are based on the rule in the content switching policy, and the action merely specifies the target load balancing virtual server. When a request matches the policy, the request is forwarded to the specified load balancing virtual server.

To create and verify a content switching action that specifies the name of the target load balancing virtual server, by using the CLI

At the command prompt, type:

```
add cs action <name> -targetLBVserver <string> [-comment <string>]

show cs action <name>
<!--NeedCopy-->
```

Example:

```
> add cs action mycsaction -targetLBVserver mylbvserver -comment "
    Forwards requests to mylbvserver."
 Done
> show cs action mycsaction
    Name: mycsaction
    Target LB Vserver: mylbvserver
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
    Comment: "Forwards requests to mylbvserver."

 Done
>
<!--NeedCopy-->
```


To configure a content switching action that specifies the name of the target load balancing virtual server, by using the GUI

1. Navigate to **Traffic Management > Content Switching > Actions**.
2. Configure a content switching action, and specify the name of the target load balancing virtual server.

Configuring an action that specifies an expression for selecting the target at run time

If you choose to configure a request-based expression that can dynamically compute the name of the target load balancing virtual server, you need to configure only one content switching policy to select the appropriate virtual server. The rule for the policy can be a simple TRUE (the policy matches all requests) because, in this case, content switching decisions are based on the expression in the action. By configuring an expression in an action, you can drastically reduce the size of your content switching configuration.

If you choose to configure a request-based expression for computing the name of the target load balancing virtual server at run time, you must carefully consider how to name the load balancing virtual servers in the configuration. You must be able to derive their names by using the request-based policy expression in the action.

For example, if you are switching requests based on the URL suffix (extension of the requested resource), when naming the load balancing virtual servers, you can follow the convention of appending the URL suffix to a predetermined string, such as `mylb_`. For example, load balancing virtual servers for HTML pages and PDF files can be named `mylb_html` and `mylb_pdf`, respectively. In that case, the rule that you can use in the content switching action, to select the appropriate load balancing virtual server, is `"mylb_"+HTTP.REQ.URL.SUFFIX`. If the content switching virtual server receives a request for an HTML page, the expression returns `mylb_html`, and the request is switched to the virtual server `mylb_html`.

To create a content switching action that specifies an expression, by using the CLI

At the command line, type the following commands to create a content switching action that specifies an expression and verify the configuration:

```
add cs action <name> -targetVserverExpr <expression>) [-comment <string
    >]

show cs action <name>
<!--NeedCopy-->
```

Example:


```
> add cs action mycsaction1 -targetvserverExpr '"mylb_" + HTTP.REQ.URL.
   SUFFIX'
 Done
> show cs action mycsaction1
    Name: mycsaction1
    Target Vserver Expression: "mylb_" + HTTP.REQ.URL.SUFFIX
    Target LB Vserver: No_Target
                ...
 Done
>
<!--NeedCopy-->
```

To configure a content switching action that specifies an expression by using the GUI

1. Navigate to **Traffic Management > Content Switching > Actions**.
2. Configure a content switching action, and specify an expression that will dynamically compute the name of the target load balancing virtual server.

Configuring content switching policies

A content switching policy defines a type of request that is to be directed to a load balancing virtual server. These policies are applied in the order of the priorities assigned to them or (if you are using Citrix ADC classic policies and do not assign priorities when binding them) in the order in which the policies were created.

The policies can be:

- **Domain-based policies.** The Citrix ADC appliance compares the domain of an incoming URL with the domains specified in the policies. The appliance then returns the most appropriate content. Domain-based policies must be classic policies. Default syntax policies are not supported for this type of content switching policy.
- **URL-based policies.** The appliance compares an incoming URL with the URLs specified in the policies. The appliance then returns the most appropriate URL-based content, which is usually the longest matching configured URL. URL-based policies must be classic policies. Default syntax policies are not supported for this type of content switching policy.
- **Rule-based policies.** The appliance compares incoming data to expressions specified in the policies. You create rule-based policies by using either a classic expression or a default syntax expression. Both classic and default syntax policies are supported for rule-based content switching policies.

> **Note**
>
> A rule based policy can be configured with an optional action. A policy with an action can be bound to multiple virtual servers or policy labels.

If you set a priority when binding your policies to the content switching virtual server, the policies are evaluated in order of priority. If you do not set specific priorities when binding your policies, the policies are evaluated in the order in which they were created.

For information about Citrix ADC classic policies and expressions, see Configuring Classic Policies and Expressions. For information about Default Syntax policies, see Configuring Default Syntax Expressions.

To create a content switching policy by using the CLI

At the command prompt, type one of the following commands:

```
add cs policy <policyName> -domain <domain>

add cs policy <policyName> -url <URLValue>

add cs policy <policyName> -rule <RULEValue>

add cs policy <policyName> -rule <RULEValue> -action <actionName>
<!--NeedCopy-->
```

Example:

```
add cs policy Policy-CS-1 -url "http://example.com"

add cs policy Policy-CS-1 -domain "example.com"

add cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(24).EQ
    (10.217.84.0)"

add cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009
     Dec)"

add cs policy Policy-CS-3 -rule "http.req.method.eq(GET)" -action act1
<!--NeedCopy-->
```

To rename a content switching policy by using the CLI

At the command prompt, type:


```
rename cs policy <policyName> <newName>
<!--NeedCopy-->
```

Example:

```
rename cs policy myCSPolicy myCSPolicy1
<!--NeedCopy-->
```

To rename a content switching policy by using the GUI

Navigate to **Traffic Management > Content Switching > Policies**, select a policy and, in the Action list, select Rename.

To create a content switching policy by using the GUI

1. Navigate to **Traffic Management > Content Switching > Policies**, and click **Add**.
2. Populate the relevant fields and click **Create**.

Configuring content switching policy labels

A policy label is a user-defined bind point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority that you assigned to them. A policy label can include one or more policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and return of control to the policy that invoked the policy label. You can create policy labels for default syntax policies only.

A content switching policy label consists of a name, a label type, and a list of policies bound to the policy label. The policy label type specifies the protocol that was assigned to the policies bound to the label. It must match the service type of the content switching virtual server to which the policy that invokes the policy label is bound. For example, you can bind TCP Payload policies to a policy label of type TCP only. Binding TCP Payload policies to a policy label of type HTTP is not supported.

Each policy in a content switching policy label is associated with either a target (which is equivalent to the action that is associated with other types of policies, such as rewrite and responder policies) or a gotoPriorityExpression option and an invoke option. That is, for a given policy in a content switching policy label, you can specify a target, or you can set the gotoPriorityExpression option and the invoke option. Also, if multiple policies evaluate to true, only the target of the last policy that evaluates to true is considered.

You can use either the Citrix ADC CLI or the GUI to configure content switching policy labels. In the Citrix ADC CLI, you first create a policy label by using the add cs policy label command. Then, you

bind policies to the policy label, one policy at a time, by using the bind cs policy label command. In the Citrix ADC GUI, you perform both tasks in a single dialog box.

To create a content switching policy label by using the CLI

At the command prompt, type:

```
add cs policylabel <labelName> <cspolicylabelTypetype>`
<!--NeedCopy-->
```

Example:

```
add cs policylabel testpollab http
<!--NeedCopy-->
```

To rename a content switching policy label by using the CLI

At the command prompt, type:

```
rename cs policylabel <labelName> <newName>`
<!--NeedCopy-->
```

Example:

```
rename cs policylabel oldPolicyLabelName newPolicyLabelName
<!--NeedCopy-->
```

To rename a content switching policy label by using the GUI

Navigate to **Traffic Management > Content Switching > Policy Labels**, select a policy label and, in the Action list, select Rename.

To bind a policy to a content switching policy label by using the CLI

At the command prompt, type the following commands to bind a policy to a policy label and verify the configuration:

```
bind cs policylabel <labelName> <policyName> <priority>[-targetVserver
    <string>] | [-gotoPriorityExpression <expression>] | [-invoke <
    labeltype> <labelName>] ]

show cs policylabel <labelName>
<!--NeedCopy-->
```


Example:

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 1
        Number of times invoked: 0
        Policy Name: test_Pol
        Priority: 100
        Target Virtual Server: LBVIP
<!--NeedCopy-->
```

> **Note**
>
> If a policy is configured with an action, the target virtual server (targetVserver), go to priority expression (gotoPriorityExpression), and invoke (invoke) parameters are not required. If a policy is not configure with an action, you need to configure at least one of the following parameters: targetVserver, gotoPriorityExpression, and invoke.

To unbind a policy from a policy label by using the CLI

At the command prompt, type the following commands to unbind a policy from a policy label and verify the configuration:

```
unbind cs policylabel <labelName> <policyName>

show cs policylabel <labelName>
<!--NeedCopy-->
```

Example:

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
<!--NeedCopy-->
```

To remove a policy label by using the CLI

At the command prompt, type:


```
rm cs policylabel <labelName>
<!--NeedCopy-->
```

To manage a content switching policy label by using the GUI

Navigate to **Traffic Management > Content Switching > Policy Labels**, configure a policy label, bind policies to the label, and optionally specify a priority, gotoPriority expression, and an invoke option.

Binding Policies to a Content Switching Virtual Server

After you create your content switching virtual server and policies, you bind each policy to the content switching virtual server. When binding the policy to the content switching virtual server, you specify the target load balancing virtual server.

> **Note**
>
> If your content switching policy uses a default syntax rule, you can configure a content switching action for the policy. If you configure an action, you must specify the target load balancing virtual server when you are configuring the action, not when you are binding the policy to the content switching virtual server. For more information about configuring a content switching action, see Configuring a Content Switching Action section.

To bind a policy to a content switching virtual server and select a target load balancing virtual server by using the CLI

At the command prompt, type:

```
bind cs vserver <name>[-lbvserver<string> -targetLBVServer<string> -
    policyname <string> -priority <positive_integer>] [-
    gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )]
    [-invoke (<labelType> <labelName>)]
<!--NeedCopy-->
```

Example:

```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -
    gotoPriorityExpression NEXT

bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -
    gotoPriorityExpression
'q.header("a").count' -flowtype REQUEST -invoke policylabel label1

```



```
bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -
    priority 20
<!--NeedCopy-->
```

> **Note**
>
> The parameters, target load balancing virtual server (targetVserver), go to priority expression (gotoPriorityExpression), and invoke method (invoke) cannot be used if a policy has an action.

To bind a policy to a content switching virtual server and select a target load balancing virtual server by using the GUI

Navigate to **Traffic Management > Content Switching > Virtual Servers**, open a virtual server and, in the
Content Switching Policy Binding section, bind a policy to the virtual server, and specify a target load balancing virtual server.

Configuring policy based logging for content switching

You can configure policy based logging for a content switching policy. Policy based logging enables you to specify a format for log messages. The contents of the log message are defined by using a default syntax expression in the content switching policy. When the content switching action specified in the policy is performed, the Citrix ADC appliance constructs the log message from the expression and writes the message to the log file. Policy based logging is particularly useful if you want to test and troubleshoot a configuration in which content switching actions identify the target load balancing virtual server at run time.

> **Note**
>
> If multiple policies bound to a given virtual server evaluate to TRUE and are configured with an audit message action, the Citrix ADC appliance does not perform all the audit message actions. It performs only the audit message action that is configured for the policy whose content switching action is performed.

To configure policy based logging for a content switching policy, you must first configure an audit message action. For more information about configuring an audit message action, see Configuring the Citrix ADC appliance for audit logging. After you configure the audit message action, you specify the action in a content switching policy.

To configure policy based logging for a content switching policy by using the CLI

At the command line, type the following commands to configure policy based logging for a content switching policy and verify the configuration:


```
set cs policy <policyName> -logAction <string>

show cs policy <policyName>
<!--NeedCopy-->
```

Example:

```
> set cs policy cspol1 -logAction csLogAction
 Done
> show cs policy cspol1

    Policy: cspol1  Rule: TRUE  Action: csact1
    LogAction: csLogAction
    Hits: 0

1)  CS Vserver: csvs1
    Priority: 10
 Done
>
<!--NeedCopy-->
```

To configure policy based logging for a content switching policy by using the GUI

Navigate to **Traffic Management > Content Switching > Policies**, open a policy and, in the Log Action list, select a log action for the policy.

Verifying the configuration

To verify that your content switching configuration is correct, you need to view the content switching entities. To verify proper operation after your content switching configuration has been deployed, you can view the statistics that are generated as the servers are accessed.

Viewing the properties of content switching virtual servers

You can view the properties of content switching virtual servers that you have configured on the Citrix ADC appliance. You can use the information to verify whether the virtual server is correctly configured and, if necessary, to troubleshoot. In addition to details such as name, IP address, and port, you can view the various policies bound to a virtual server, and its traffic-management settings.

The content switching policies are displayed in the order of their priority. If more than one policy has the same priority, they are shown in the order in which they are bound to the virtual server.

> **Note**
>
> If you have configured the content switching virtual server to forward traffic to a load balancing virtual server, you can also view the content switching policies by viewing the properties of the load balancing virtual server.

To view the properties of content switching virtual servers by using the CLI

To list basic properties of all content switching virtual servers in your configuration, or detailed properties of a specific content switching virtual server, at the command prompt, type one of the following commands:

```
show cs vserver

show cs vserver <name>
<!--NeedCopy-->
```

Example

```
1.
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
```



```
Flowtype: REQUEST

2) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

3) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

4) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
GotoPriority Expression: END

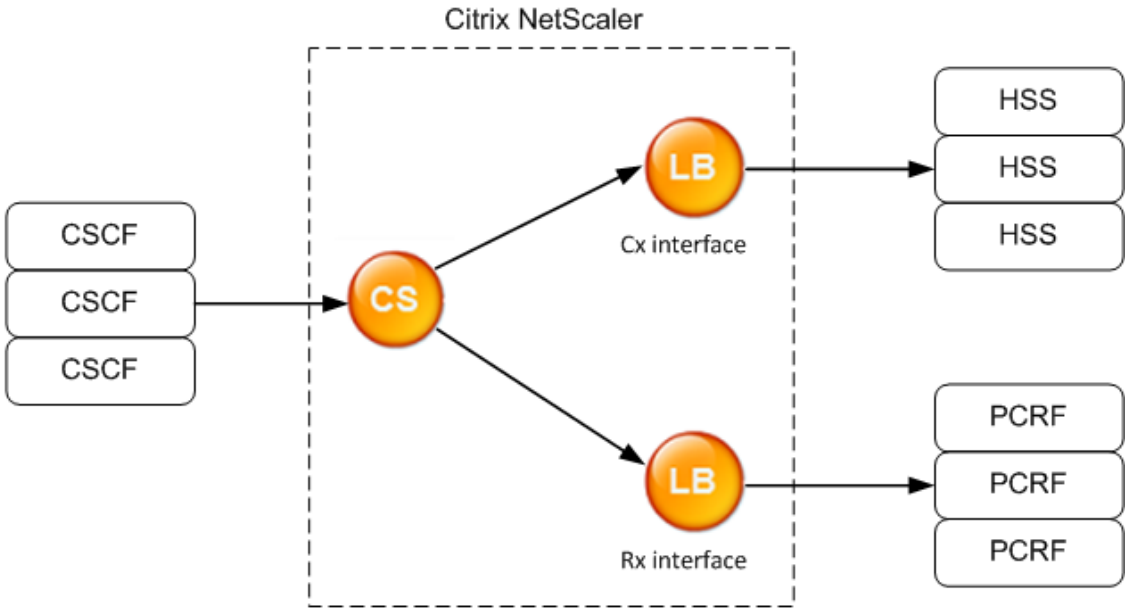
1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
 Done
>

show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
…
<!--NeedCopy-->
```


Viewing content switching policies

You can view the properties of the content switching policies that you defined, such as the name, domain, and URL or expression, and use the information to find any mistakes in the configuration, or to troubleshoot if something is not working as it must.

To view the properties of content switching policies by using the CLI

To list either the basic properties of all content switching policies in your configuration or the detailed properties of a specific content switching policy, at the command prompt, type one of the following commands:

```
show cs policy

show cs policy <PolicyName>
<!--NeedCopy-->
```

Example:

```
show cs policy

show cs policy Policy-CS-1
<!--NeedCopy-->
```

To view the properties of content switching policies by using the GUI

Navigate to **Traffic Management** > **Content Switching** > **Policies**, select a policy and, in the Action list, select **Show Bindings**.

Viewing a content switching virtual server configuration by using the visualizer

The Content Switching Visualizer is a tool that you can use to view a content switching configuration in graphical format. You can use the visualizer to view the following configuration items:

- A summary of the load balancing virtual servers to which the content switching virtual server is bound.
- All services and service groups that are bound to the load balancing virtual server and all monitors that are bound to the services.
- The configuration details of any displayed element.

- Any policies bound to the content switching virtual server. These policies need not be content switching policies. Many types of policies, such as Rewrite policies, can be bound to a content switching virtual server.

After you configure the various elements in a content switching and load balancing setup, you can export the entire configuration to an application template file.

> **Note**
>
> The Visualizer requires a graphical interface, so it is available only through the GUI.

To view a content switching configuration by using the Visualizer in the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the **Content Switching Visualizer** window, you can adjust the viewable area as follows:
   - Click the **Zoom In** and **Zoom Out** icons to increase or decrease the viewable area.
   - Click the **Save Image** icon to save the graph as an image file.
   - In the Search in text field, begin typing the name of the item you are looking for. When you have typed enough characters to identify the item, its location is highlighted. To restrict the search, click the drop-down menu and select the type of element that you want to search for.
4. To view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view policies that are bound to the virtual server, in the tool bar at the top of the dialog box select one or more feature-specific policy icons. If policy labels are configured, they appear in the main view area.
   - To view the configuration details for a bound service or service group, click the icon for the service, click the Related Tasks tab, and then click Show Member Services.
   - To view the configuration details for a monitor, click the icon for the monitor, click the **Related Tasks** tab, and then click **View Monitor**.
5. To view detailed statistics for any virtual server in the content switching configuration, click the virtual server for which you want to view statistics, then click the Related Tasks tab, and then click **Statistics**.
6. To view a comparative list of the parameters whose values either differ or are not defined across service containers for a load balancing virtual server, click the icon for a container, click the **Related Tasks** tab, and then click **Service Attributes Diff**.
7. To view monitor binding details for the services in a container, in the Service Attributes Diff dialog box, in the Group column for the container, click **Details**. This comparative list helps you

determine which service container has the configuration you want to apply to all the service containers.

8. To view the number of requests received per second at a given point in time by the virtual servers in the configuration, and the number of select per second at a given point in time for rewrite, responder, and cache policies, click Show Stats. The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time. It is refreshed manually. To refresh the information, click Refresh Stats.

   > **Note**
   >
   > This option is available only on Citrix ADC nCore builds.

9. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click Related Tasks, click Copy Properties, and then paste the information into a document.
10. To export the entire configuration that is displayed in the Visualizer to an application template file, click the icon for the content switching virtual server, click Related Tasks, and then click Create Template. When creating the application template, you can configure variables in some policy expressions and actions. For more information about creating the application template file and configuring variables for a template, see AppExpert.

Customizing the basic content switching configuration

September 14, 2021

After you configure a basic content switching setup, you might need to customize it to meet your requirements. If your web servers are UNIX-based and rely on case sensitive pathnames, you can configure case sensitivity for policy evaluation. You can also set precedence for evaluation of the content switching policies that you configured. You can configure HTTP and SSL content switching virtual servers to listen on multiple ports instead of creating separate virtual servers. If you want to configure content switching for a specific a virtual LAN, you can configure a content switching virtual server with a listen policy.

Configuring case sensitivity for policy evaluation

You can configure the content switching virtual server to treat URLs as case sensitive in URL-based policies. When case sensitivity is configured, the Citrix ADC appliance considers the case when evaluating policies. For example, if case sensitivity is off, the URLs /a/1.htm and /A/1.HTM are treated as identical. If case sensitivity is on, those URLs are treated as separate and can be switched to different targets.

To configure case sensitivity by using the command line interface

At the command prompt, type:

```
set cs vserver \<name\> -caseSensitive (ON|OFF)
```

Example:

```
set cs vserver Vserver-CS-1 -caseSensitive ON
<!--NeedCopy-->
```

To configure case sensitivity by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and then select Case Sensitive.

Setting the Precedence for Policy Evaluation

Precedence refers to the order in which policies that are bound to a virtual server are evaluated. You need not configure precedence, the default precedence often works correctly.

You can configure either URL-based precedence or rule-based precedence in the following scenarios:

- One policy or set of policies must be applied first
- Another policy or set of policies is applied only if the first set does not match a request.

Precedence with URL-Based Policies

If there are multiple matching URLs for the incoming request, the precedence (priority) for URL-based policies is:

1. Domain and exact URL
2. Domain, prefix, and suffix
3. Domain and suffix
4. Domain and prefix
5. Domain only
6. Exact URL
7. Prefix and suffix
8. Suffix only
9. Prefix only
10. Default

If you configure precedence based on URL, the request URL is compared to the configured URLs. If none of the configured URLs match the request URL, then rule-based policies are checked. If the request URL does not match any rule-based policies, or if the content group selected for the request is down, then the request is processed as follows:

- If you configure a default group for the content switching virtual server, then the request is forwarded to the default group.
- If the configured default group is down or if no default group is configured, then an "HTTP 404 Not Found" error message is sent to the client.

> **Note**
>
> You must configure URL-based precedence if the content type (for example, images) is the same for all clients. However, if different types of content must be served based on client attributes (such as Accept-Language), you must use rule-based precedence.

Precedence with Rule-Based Policies

If you configure precedence based on rules, which is the default setting, the request is tested based on the rule-based policies you have configured. If the request does not match any rule-based policies, or if the content group selected for the incoming request is down, the request is processed in the following manner:

- If a default group is configured for the content switching virtual server, the request is forwarded to the default group.
- If the configured default group is down or if no default group is configured, an "HTTP 404 Not Found" error message is sent to the client.

To configure precedence by using the command line interface

At the command prompt, type:

```
set cs vserver \<name\> -precedence ( RULE | URL )
```

Example:

```
set cs vserver Vserver-CS-1 -precedence RULE
```

To configure precedence by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **Traffic Settings**, and then specify Precedence.

Support for Multiple Ports for HTTP and SSL Type Content Switching Virtual Servers

You can configure the Citrix ADC so that HTTP and SSL content switching virtual servers listen on multiple ports, without having to configure separate virtual servers. This feature is especially useful if you want to base a content switching decision on a part of the URL and other L7 parameters. Instead of configuring multiple virtual servers with the same IP address and different ports, you can configure one IP address and specify the port as *. As a result, the configuration size is also reduced.

To configure an HTTP or SSL content switching virtual server to listen on multiple ports by using the command line

At the command prompt, type:

```
add cs vserver \<name\> \<serviceType\> \<IPAddress\> Port \*
```

Example

```
> add cs vserver cs1 HTTP 10.102.92.215 *
 Done
> sh cs vserver cs1
        cs1 (10.102.92.215:*) - HTTP    Type: CONTENT
        State: UP
        Last state change was at Tue May 20 01:15:49 2014
        Time since last state change: 0 days, 00:00:03.270
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Vserver IP and Port insertion: OFF
        L2Conn: OFF     Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
        IcmpResponse: PASSIVE
        RHIstate:  PASSIVE
        TD: 0
 Done
<!--NeedCopy-->
```


To configure an HTTP or SSL content switching virtual server to listen on multiple ports by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and create a virtual server of type HTTP or SSL.
2. Use an asterisk (*) to specify the port.

Configuring per-VLAN Wildcard Virtual Servers

If you want to configure content switching for traffic on a specific VLAN, you can create a wildcard virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcard virtual server that listens to a specific VLAN by using the command line interface

At the command prompt, type:

```
add cs vserver \<name\> \<serviceType\> IPAddress `* Port *` -
   listenpolicy \<expression\> \[-listenpriority \<positive\_integer
   \>\]
<!--NeedCopy-->
```

Example:

```
add cs vserver Vserver-CS-vlan1 ANY * *
-listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
<!--NeedCopy-->
```

To configure a wildcard virtual server that listens to a specific VLAN by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, and configure a virtual server. Specify a listen policy that restricts it to processing traffic only on the specified VLAN.

After you have created this virtual server, you bind it to one or more services as described in Setup basic load balancing.

Configuring the Microsoft SQL Server Version Setting

You can specify the version of Microsoft® SQL Server® for a content switching virtual server that is of type MSSQL. The version setting is recommended if you expect some clients to not be running the

same version as your Microsoft SQL Server product. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version.

To set the Microsoft SQL Server version parameter by using the command line interface

At the command prompt, type the following commands to set the Microsoft SQL Server version parameter for a content switching virtual server and verify the configuration:

- `set cs vserver \<name\> -mssqlServerVersion \<mssqlServerVersion\>`
- `show cs vserver \<name\>`

Example

```
1 > set cs vserver myMSSQLcsvip -mssqlServerVersion 2008R2 Done > show cs
      vserver myMSSQLcsvip myMSSQLcsvip (192.0.2.13:1433) - MSSQL Type:
     CONTENT State: UP . . . . . MSsql Server Version: 2008R2 . . . . .
      . Done >
2 <!--NeedCopy-->
```

To set the Microsoft SQL Server version parameter by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, configure a virtual server, and specify the protocol as MSSQL.
2. In **Advanced Settings**, specify the **Server Version**.

Enable external TCP health check for UDP virtual servers

In public clouds, you can use the Citrix ADC appliance as a second-tier load balancer when the native load balancer is used as a first tier. The native load balancer can be an application load balancer (ALB) or a network load balancer (NLB). Most of the public clouds do not support UDP health probes in their native load balancers. To monitor the health of the UDP application, public clouds recommend adding a TCP-based endpoint to your service. The endpoint reflects the health of the UDP application.

The Citrix ADC appliance supports the external TCP-based health check for a UDP virtual server. This feature introduces a TCP listener on the VIP of the content switching virtual server and the configured port. The TCP listener reflects the status of the virtual server.

To enable external TCP health check for UDP virtual servers by using CLI

At the command prompt, type the following command to enable an external TCP health check with the tcpProbePort option:


```
add cs vserver <name> <protocol> <IPAddress> <port> -tcpProbePort <
    tcpProbePort>
<!--NeedCopy-->
```

Example:

```
add cs vserver Vserver-CS-1 UDP 10.102.29.161 5002 -tcpProbePort 5000
<!--NeedCopy-->
```

To enable external TCP health check for UDP virtual servers by using GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and then create a virtual server.
2. Click **Add** to create a virtual server.
3. In the **Basic Settings** pane, add the port number in the **TCP Probe Port** field.
4. Click **OK**.

Content Switching for Diameter Protocol

September 14, 2021

For Diameter-protocol traffic, you can configure the Citrix ADC appliance (or virtual appliance) to act as a relay agent that load balances and forwards a packet to the appropriate destination on the basis of the message content (AVP value in the message). Since the appliance does not perform any application-level processing, it provides relaying services for all diameter applications as specified by the configured content switching policies. Therefore, the appliance advertises the Relay Application ID in the capability exchange answer (CEA) message when the client establishes a diameter connection. You must configure a content switching virtual server, load balancing virtual servers, and services to represent the diameter nodes. When a request reaches the content switching virtual server, the virtual server applies the content switching policies associated with that type of request. After evaluating the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

A diameter interface provides a connection between the different diameter nodes. The following sample deployment uses Cx and Rx interfaces. A Cx interface provides a connection between a CSCF and an HSS. An Rx interface provides a connection between a CSCF and a PCRF. All the messages reach the Citrix ADC appliance. Depending on whether the message is for a Cx or an Rx interface, and on the content switching policies defined, the Citrix ADC selects an appropriate load balancing server pool.



CSCF=Call Session Control Function
HSS=Home Subscriber Server
PCRF=Policy and Charging Rules Function

Sample Configuration

1. For each entity, create a service, a load balancing server, and bind the service to the virtual server.

```
add service svc_pcrf[1-3] 1.1.1.1[1-3] DIAMETER 3868
add service svc_hss[1-3] 1.1.1.2[1-3] DIAMETER 3868
add lb vserver vs_rx DIAMETER -persistenceType DIAMETER -
   persistavpno 263
add lb vserver vs_cx DIAMETER -persistenceType DIAMETER -
   persistavpno 263
bind lb vserver vs_rx svc_pcrf[1-3]
bind lb vserver vs_cx svc_hss[1-3]
<!--NeedCopy-->
```

2. Create a content switching virtual server and two actions (one for each load balancing virtual server). Create two content switching policies and bind these policies to the content switching virtual server, specifying a priority for each policy.

```
add cs vserver cs_diameter DIAMETER 10.1.1.10 3868
add cs action cx_action -targetLBVserver vs_cx
add cs action rx_action  - targetLBvserver vs_rx
```



```
add cs policy cx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777216)" -action cx_action
add cs policy rx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777236)" -action rx_action
bind cs vserver cs_diameter -policyName rx_policy -priority 100
bind cs vserver cs_diameter -policyName cx_policy -priority 110
<!--NeedCopy-->
```

Protecting the content switching setup against failure

September 14, 2021

Content switching might fail when the content switching virtual server goes DOWN or fails to handle excessive traffic, or for other reasons. To reduce the chances of failure, you can take the following measures to protect the content switching setup against failure:

Configuring a backup virtual server

If the primary content switching virtual server is marked DOWN or DISABLED, the Citrix ADC appliance can direct requests to a backup content switching virtual server. It can also send a notification message to the client regarding the site outage or maintenance. The backup content switching virtual server is a proxy and is transparent to the client.

When configuring the backup virtual server, you can specify the configuration parameter Disable Primary When Down to ensure that, when the primary virtual server comes back up, it remains the secondary until you manually force it to take over as the primary. It is useful if you want to ensure that any updates to the database on the server for the backup are preserved, enabling you to synchronize the databases before restoring the primary virtual server.

You can configure a backup content switching virtual server when you create a content switching virtual server or when you change the optional parameters of an existing content switching virtual server. You can also configure a backup content switching virtual server for an existing backup content switching virtual server, thus creating cascaded backup content switching virtual servers. The maximum depth of cascaded backup content switching virtual servers is 10. The appliance searches for a backup content switching virtual server that is up and accesses that content switching virtual server to deliver the content.

> **Note**
>
> If a content switching virtual server is configured with both a backup content switching virtual server and a redirect URL, the backup content switching virtual server takes precedence over the

> redirect URL. The redirect is used when the primary and backup virtual servers are down.

To set up a backup content switching virtual server by using the CLI

At the command prompt, type:

```
set cs vserver <name> -backupVserver <string> -disablePrimaryOnDown (ON
    |OFF)
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -
    disablePrimaryOnDown ON
<!--NeedCopy-->
```

To set up a backup content switching virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, configure a virtual server, and specify the protocol as MYSQL.
2. In **Advanced Settings**, select **Protection**, and specify a **Backup Virtual Server**.

Diverting excess traffic to a backup virtual server

The spillover option diverts new connections arriving at a content switching virtual server to a backup content switching virtual server when the number of connections to the content switching virtual server exceeds the configured threshold value. The threshold value is dynamically calculated, or you can set the value. The number of established connections (in TCP) at the virtual server is compared with the threshold value. When the number of connections reaches the threshold, new connections are diverted to the backup content switching virtual server.

If the backup content switching virtual servers reach the configured threshold and are unable to take the load, the primary content switching virtual server diverts all requests to the redirect URL. If a redirect URL is not configured on the primary content switching virtual server, subsequent requests are dropped.

To configure a content switching virtual server to divert new connections to a backup virtual server by using the CLI

At the command prompt, type:


```
set cs vserver \<name\> -soMethod \<methodType\> -soThreshold \<
    thresholdValue\> -soPersistence \<persistenceValue\> -
    soPersistenceTimeout \<timeoutValue\>
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

To set a content switching virtual server to divert new connections to a backup virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, configure a virtual server, and specify the protocol as MYSQL.
2. In **Advanced Settings**, select **Protection**, and configure spillover.

Configuring a redirection URL

You can configure a redirect URL to communicate the status of the Citrix ADC appliance if a content switching virtual server of type HTTP or HTTPS is DOWN or DISABLED. This URL can be local or remote.

Redirect URLs can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Citrix recommends using an absolute URL. That is, a URL ending in /, for example www.example.com/ instead of a relative URL. A relative URL redirection might result in the vulnerability scanner reporting a false positive.

> **Note**
>
> If a content switching virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect URL is used when the primary and backup virtual servers are down.

When redirection is configured and the content switching virtual server is unavailable, the appliance issues an HTTP 302 redirect to the user's browser.

To configure a redirect URL for when the content switching virtual server is unavailable by using the CLI

At the command prompt, type:

```
set cs vserver \<name\> -redirectURL \<URLValue\>
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/
   mysite/maintenance
<!--NeedCopy-->
```

To configure a redirect URL for when the content switching virtual server is unavailable by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, configure a virtual server, and specify the protocol as MYSQL.
2. In **Advanced Settings**, select **Protection**, and specify a Redirect URL.

Configuring the state update option

The content switching feature enables the distribution of client requests across multiple servers based on the specific content presented to the users. For efficient content switching, the content switching virtual server distributes the traffic to the load balancing virtual servers according to the content type, and the load balancing virtual servers distribute the traffic to the physical servers according to the specified load balancing method.

For smooth traffic management, it is important for the content switching virtual server to know the status of the load balancing virtual servers. The state update option helps to mark the content switching virtual server DOWN if the load balancing virtual server bound to it is marked DOWN. A load balancing virtual server is marked DOWN if all the physical servers bound to it are marked DOWN.

When State Update is disabled:

The status of the content switching virtual server is marked as UP. It remains UP even if there is no bound load balancing virtual server that is UP.

When State Update is enabled:

When you add a content switching virtual server, initially, its status is shown as DOWN. When you bind a load balancing virtual server whose status is UP, the status of the content switching virtual server becomes UP.

If more than one load balancing virtual server is bound and if one of them is specified as the default, the status of the content switching virtual server reflects the status of the default load balancing virtual server.

If more than one load balancing virtual server is bound without any of them being specified as the default, the status of the content switching virtual server is marked UP only if all the bound load balancing virtual servers are UP.

To configure the state update option by using the CLI

At the command prompt, type:

```
add cs vserver \<name\> \<protocol\> \<ipAddress\> \<port\> -
   stateUpdate ENABLED
<!--NeedCopy-->
```

Example

```
 add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED
    -cltTimeout 180
<!--NeedCopy-->
```

To configure the state update option by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, configure a virtual server, and specify the protocol as MYSQL.
2. In **Advanced Settings**, select **Traffic Settings**, and then select **State Update**.

Flushing the surge queue

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the Citrix ADC appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between the client

and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you might want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed. The other requests in the surge queue of the load balancing virtual server are not flushed.

> **Note**
>
> You cannot flush the surge queues of cache redirection, authentication, VPN, or GSLB virtual servers or GSLB services.
> Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the CLI

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while running the command, the surge queue of the specified entity is flushed. If more than one entity has the same name, the appliance flushes the surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while running the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a

<serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.

- If you run the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>].
<!--NeedCopy-->
```

Examples

```
1. flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual
    server that is named SVC1ANZGB and has IP address as 10.10.10

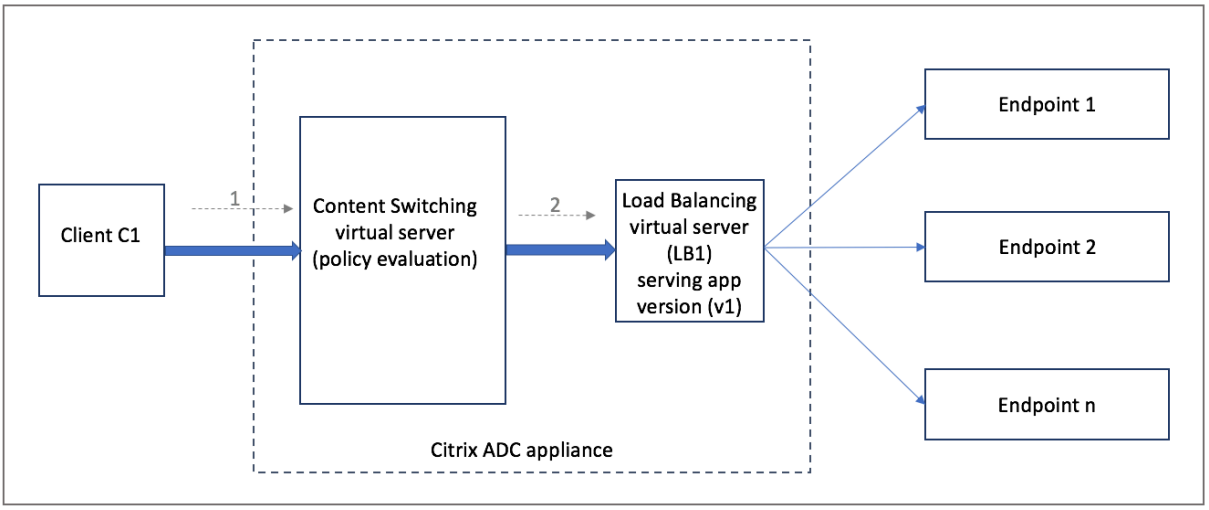
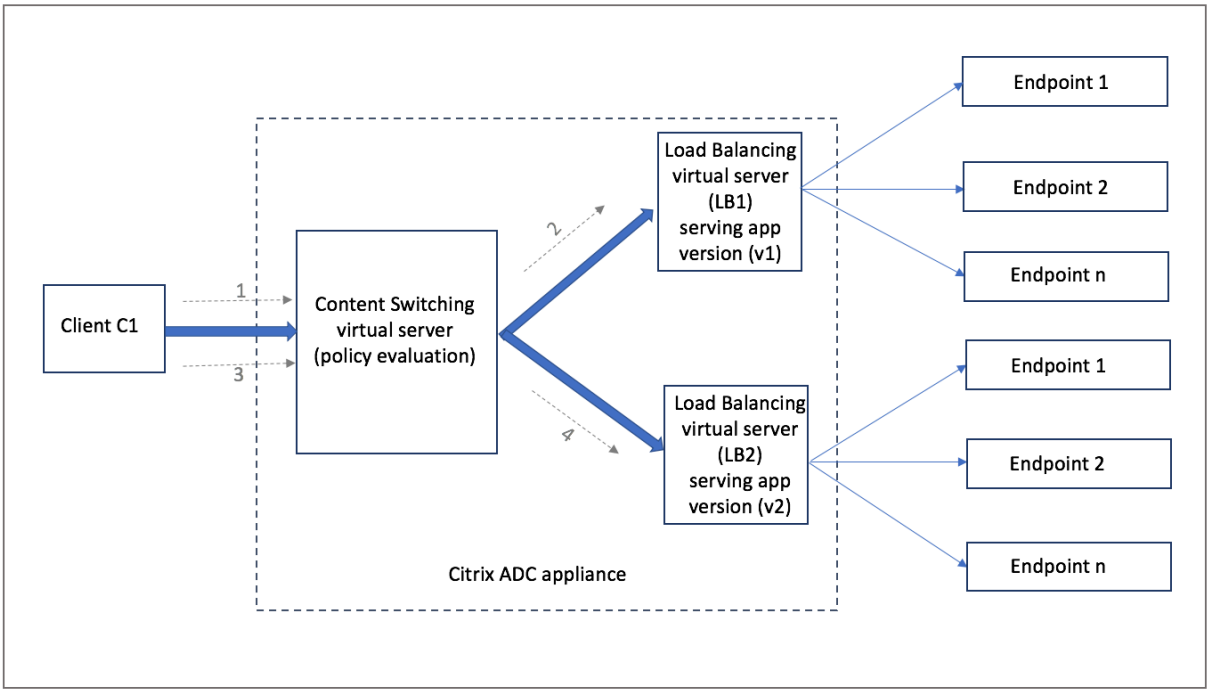
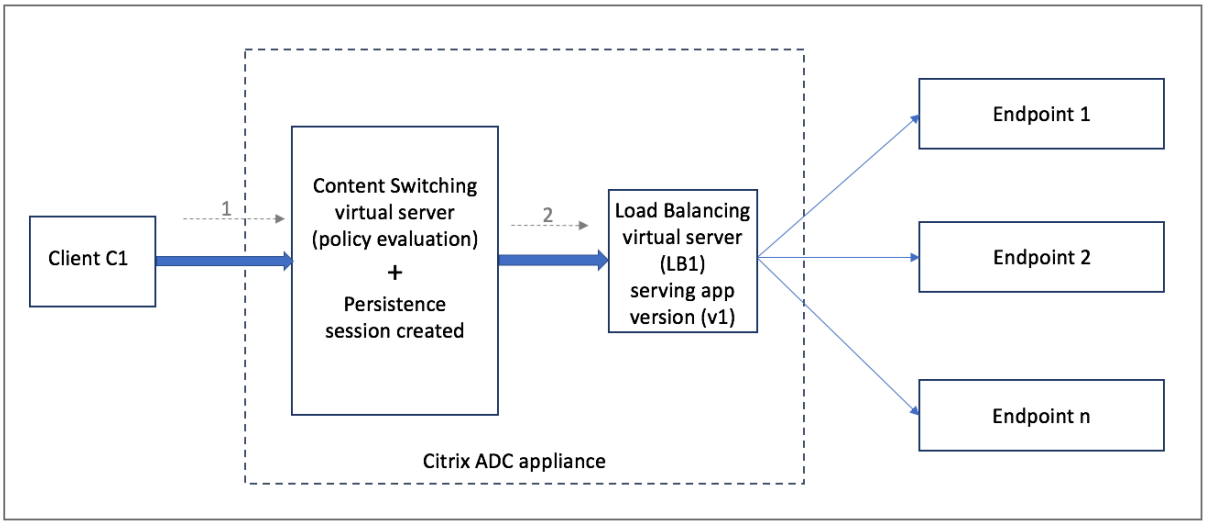
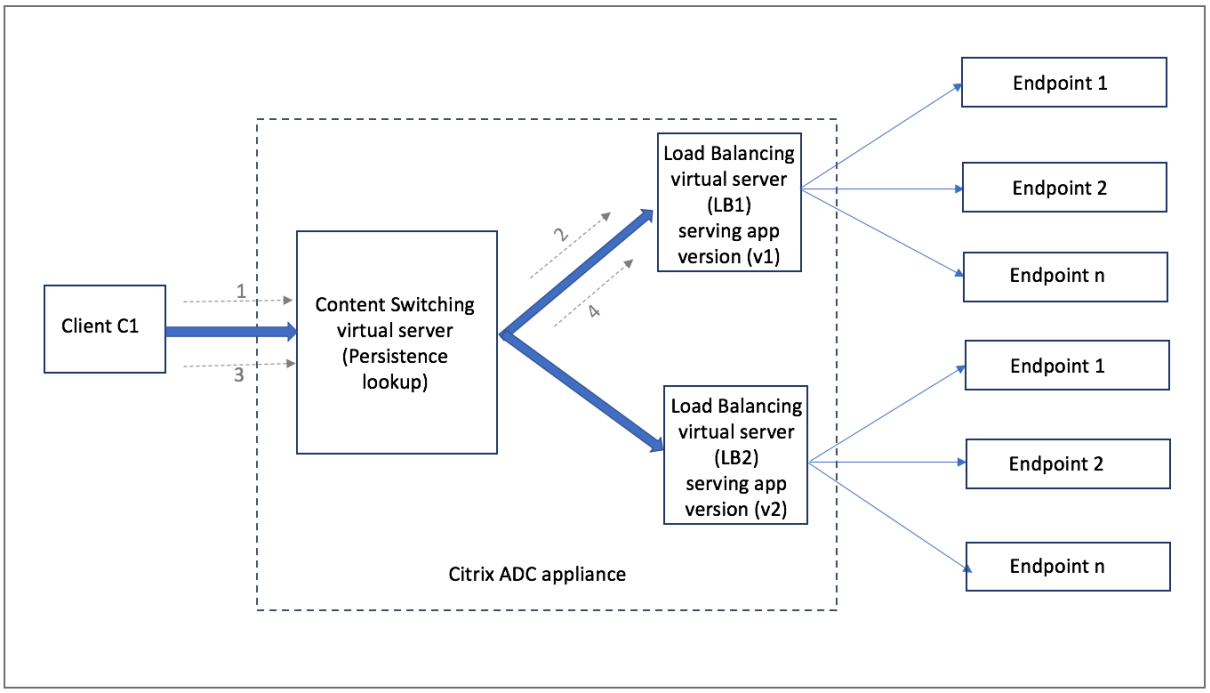
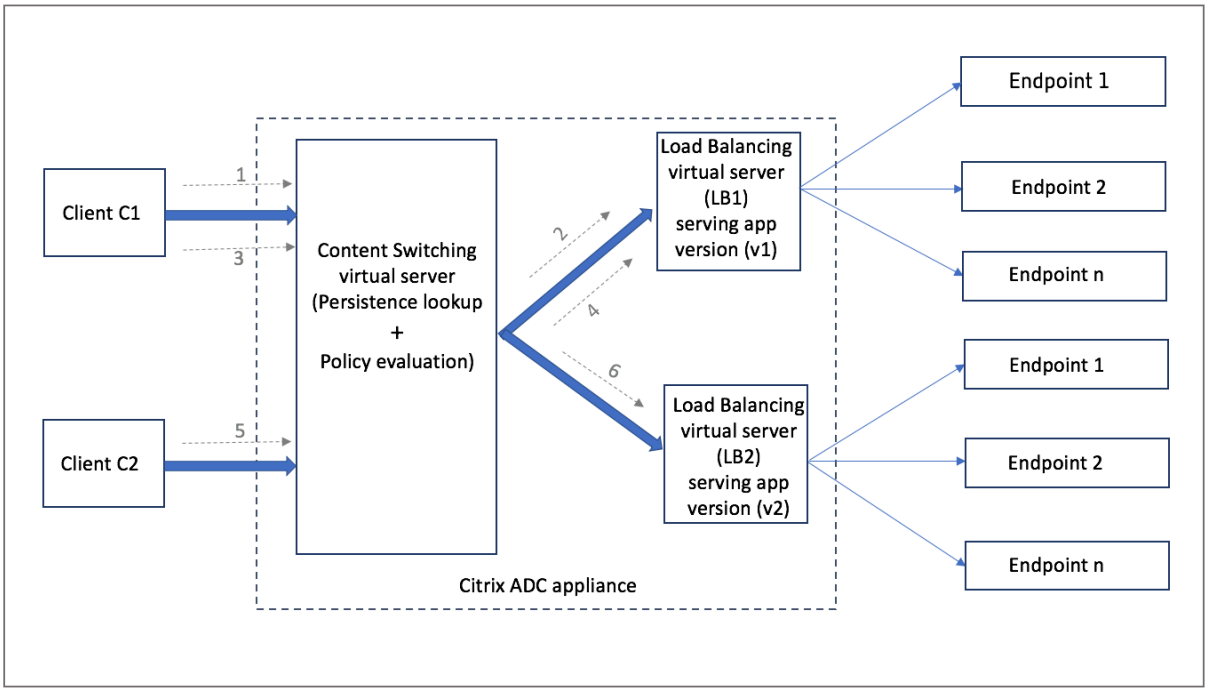
2. flush ns surgeQ
The above command flushes all the surge queues on the appliance.
<!--NeedCopy-->
```

To flush a surge queue by using the GUI

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server and, in the Action list, select **Flush Surge Queue**.

Managing a content switching setup

September 14, 2021

After a content switching setup is configured, it might require periodic changes. When operating systems or software is updated, or hardware wears out and is replaced, you might need to take down your setup. Load on your setup might increase, requiring more resources. You might also modify the configuration to improve performance.

These tasks might require unbinding policies from the content switching virtual server, or disabling or removing content switching virtual servers. After you have changed your setup, you might need to re-enable servers and rebind policies. You might also want to rename your virtual servers.

Unbinding policies from the content switching virtual server

When you unbind a content switching policy from its virtual server, the virtual server no longer includes that policy when determining where to direct requests.

To unbind a policy from a content switching virtual server by using the CLI

At the command prompt, type:

```
unbind cs vserver <name> -policyname <string>
```

Example:

```
unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1
```

To unbind a policy from a content switching virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open the virtual server.
2. Click **Policies** section, select the policy, and click **Unbind**.

Removing Content Switching Virtual Servers

You normally remove a content switching virtual server only when you no longer require the virtual server. When you remove a content switching virtual server, the Citrix ADC appliance first unbinds all policies from the content switching virtual server, and then removes it.

To remove a content switching virtual server by using the CLI

At the command prompt, type:

```
rm cs vserver <name>
```

Example:

```
rm cs vserver Vserver-CS-1
```

To remove a content switching virtual server by using the GUI

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server, and click **Delete**.

Disabling and Re-Enabling Content Switching Virtual Servers

Content switching virtual servers are enabled by default when you create them. You can disable a content switching virtual server for maintenance. If you disable the content switching virtual server, the state of the content switching virtual server changes to Out of Service. While out of service, the content switching virtual server does not respond to requests.

To disable or re-enable a virtual server by using the CLI

At the command prompt, type one of the following commands:

- `disable cs vserver <name>`
- `enable cs vserver <name>`

Example:

```
disable cs vserver Vserver-CS-1
enable cs vserver Vserver-CS-1
```

To disable or re-enable a virtual server by using the GUI

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server and, in the **Action** list, select **Enable** or **Disable**.

Renaming Content Switching Virtual Servers

You can rename a content switching virtual server without unbinding it. The new name is propagated automatically to all affected parts of the Citrix ADC configuration.

To rename a virtual server by using the CLI

At the command prompt, type:

```
rename cs vserver <name> <newName>
```

Example:

```
1 `rename cs vserver Vserver-CS-1 Vserver-CS-2`
```

To rename a virtual server by using the GUI

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server and, in the **Action** list, select **Rename**.

Managing content switching policies

You can modify an existing policy by configuring the rules or changing the URL of the policy, or you can remove a policy. You can also rename an existing advanced content switching policy. You can create different policies based on the URL. URL-based policies can be of different types, as described in the following table.

For more information, see Examples of URL-Based Policies.

> **Note**
>
> You can configure rule-based content switching using classical policy expressions or advanced policy expressions.

To modify, remove, or rename a policy by using the CLI

At the command prompt, type one of the following commands:

- `set cs policy <policyName> [-domain <domainValue>] [-rule <ruleValue>] [-url <URLValue>]`
- `rm cs policy <policyName>`
- `rename cs policy <policyName> <newPolicyName>`

Example:

```
set cs policy Policy-CS-1 -domain "www.domainxyz.com"

set cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(22).EQ
   (10.100.148.0)"

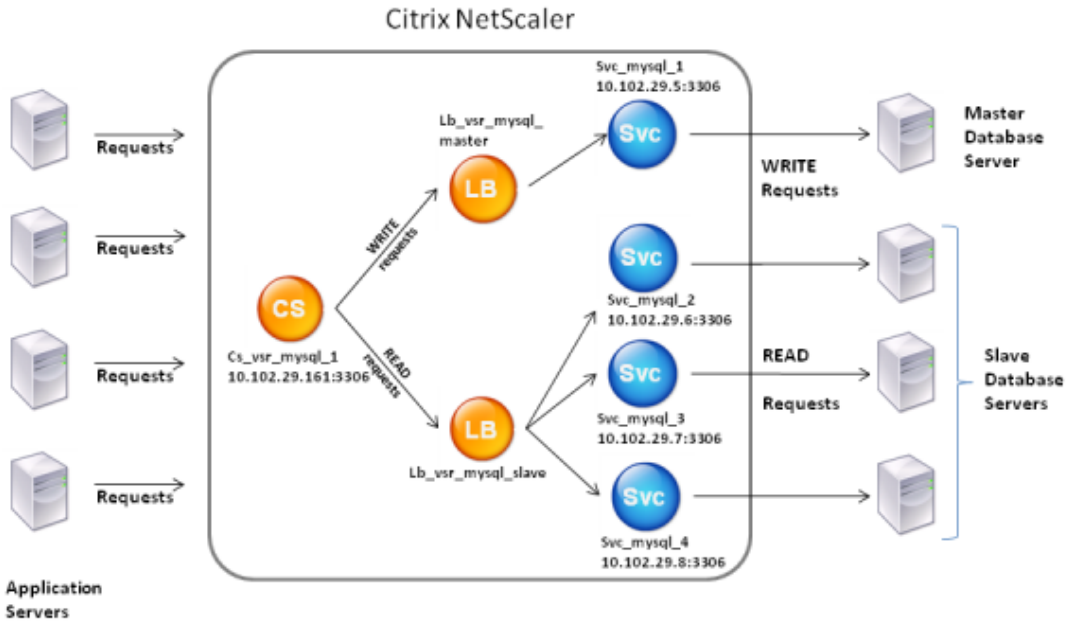
set cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010
    Jul)"

set cs policy Policy-CS-1 -url /sports/*

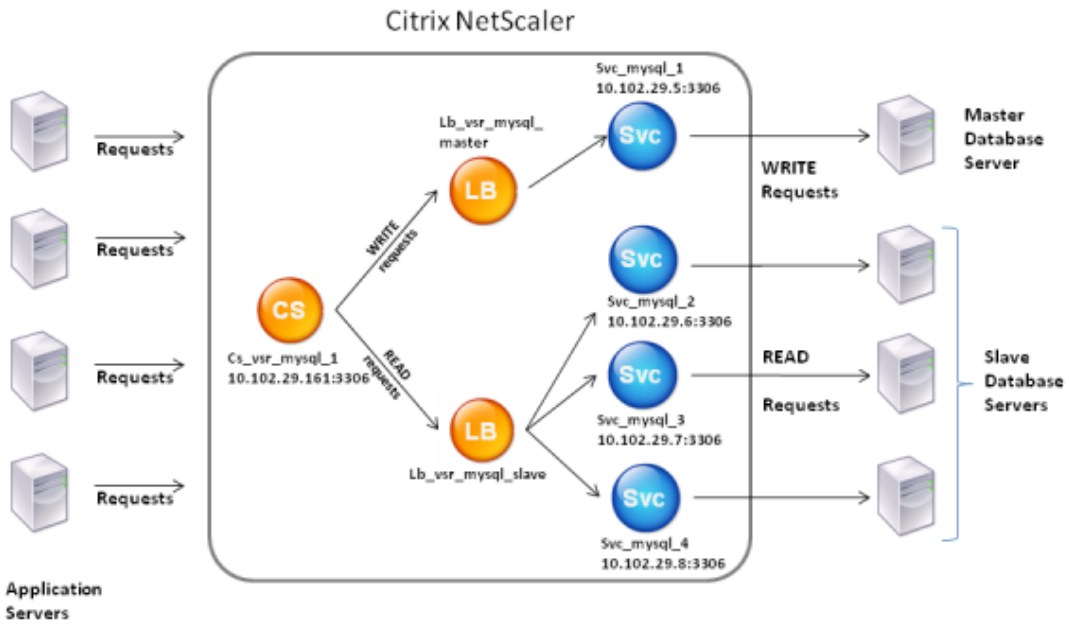
rename cs policy Policy-CS-1 Policy-CS-11

rm cs policy Policy-CS-1
```

To modify, remove, or rename a policy by using the GUI

1. Navigate to **Traffic Management > Content Switching > Policies**.
2. Select the policy, and either delete it, edit it or, in the **Action** list, click **Rename**.

Managing client connections

September 14, 2021

To ensure efficient management of client connections, you can configure the content switching virtual servers on the Citrix ADC appliance to use the following features:

- **Configuring the ICMP Response.** You can configure the Citrix ADC appliance to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP virtual server RESPONSE.
  The following settings can be made on a virtual server:
  - When you set ICMP virtual server RESPONSE to PASSIVE on all virtual servers, the Citrix ADC appliance always responds.
  - When you set ICMP virtual server RESPONSE to ACTIVE on all virtual servers, the ADC appliance responds even if one virtual server is UP.
  - When you set ICMP virtual server RESPONSE to ACTIVE on some and PASSIVE on others, the ADC appliance responds even if one virtual server set to ACTIVE is UP.

Redirecting Client Requests to a Cache

The Citrix ADC cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the burden of responding to HTTP requests and improve your website performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the Citrix ADC appliance sends cacheable HTTP requests to the cache and non-cacheable HTTP requests to the origin Web server. For more information on cache redirection, see "Cache Redirection.

To configure cache redirection on a virtual server by using the CLI

At the command prompt, type:

```
set cs vserver \<name\> -cacheable \<Value\>
```

Example

```
set cs vserver Vserver-CS-1 -cacheable yes
```


To configure cache redirection on a virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and select **Cacheable**.

Enabling Delayed Cleanup of Virtual Server Connections

Under certain conditions, you can configure the down state flush setting to terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources and in certain cases speeds recovery of overloaded load balancing setups.

To configure the down state flush setting on a virtual server by using the CLI

At the command prompt, type:

```
set cs vserver \<name\> -downStateFlush \<Value\>
```

Example

```
1 set cs vserver Vserver-CS-1 -downStateFlush enabled
2 <!--NeedCopy-->
```

To configure the down state flush setting on a virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and then select **Down State Flush**.

Rewriting Ports and Protocols for Redirection

Virtual servers and the services that are bound to them might use different ports. When a service responds to an HTTP connection with a redirect, you might need to configure the Citrix ADC appliance to modify the port and the protocol to ensure that the redirection goes through successfully. You do it by enabling and configuring the redirectPortRewrite setting.

To configure HTTP redirection on a virtual server by using the CLI

At the command prompt, type:

```
set cs vserver \<name\> -redirectPortRewrite \<Value\>
```


Example

```
set cs vserver Vserver-CS-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

To configure HTTP redirection on a virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and select **Rewrite**.

Inserting the IP Address and Port of a Virtual Server in the Request Header

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the Citrix ADC appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, you must configure the required header tag on both virtual servers.

> **Note**
>
> This option is not supported for wildcard virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the CLI

At the command prompt, type:

```
set cs vserver \<name\> -insertVserverIPPort \<vServerIPPORT\>
```

Example

```
set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80
<!--NeedCopy-->
```

To insert the IP address and port of the virtual server in the client requests by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.

2. In **Advanced Settings**, select **Traffic Settings** and, in the Virtual Server IP Port Insertion list, select VIPADDR or V6TOV4MAPPING, and specify a port header in the virtual server IP Port Insertion Value.

Setting a Time-out Value for Idle Client Connections

You can configure a virtual server to terminate any idle client connections after a configured time-out period elapses. When you configure this setting, the Citrix ADC appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection.

To set a time-out value for idle client connections by using the CLI

At the command prompt, type:

```
set cs vserver \<name\> -cltTimeout \<Value\>
```

Example

```
1 set cs vserver Vserver-CS-1 -cltTimeout 100
2 <!--NeedCopy-->
```

To set a time-out value for idle client connections by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and specify a **Client Idle Time-Out** value.

Identifying Connections with the 4-tuple and Layer 2 Connection Parameters

You can now set the L2Conn option for a content switching virtual server. With the L2Conn option set, connections to the content switching virtual server are identified by the combination of the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) and Layer 2 connection parameters. The Layer 2 connection parameters are the MAC address, VLAN ID, and channel ID.

To set the L2Conn option for a content switching virtual server by using the CLI

At the command line, type the following commands to configure the L2Conn parameter for a content switching virtual server and verify the configuration:

```
1 -  set cs vserver \<name\> -l2Conn (**ON** | **OFF**)
2 -  show cs vserver \<name\>
```


Example

```
> set cs vserver mycsvserver -l2Conn ON
Done
> show cs vserver mycsvserver
     mycsvserver (192.0.2.56:80) - HTTP   Type: CONTENT
     State: UP
          . . .
          . . .
     L2Conn: ON Case Sensitivity: ON
          . . .
          . . .
Done
>
<!--NeedCopy-->
```

To set the L2Conn option for a content switching virtual server by using the GUI

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and then select **Layer 2 Parameters**.

Persistence support for content switching virtual server

September 14, 2021

Applications are moving from monolithic architectures toward microservices architecture. Different versions of the same application can co-exist in the microservices architecture. Citrix ADC appliance must support continuous deployment of applications. It is achieved by platforms that perform Canary deployments (such as Spinnaker). In a continuous deployment setup, a newer version of an application is deployed automatically and exposed to client traffic in stages until the application is stable to take complete traffic. Also, there must be uninterrupted services to the client.

The Citrix ADC content switching feature enables Citrix ADC the appliance to distribute client requests across multiple load balancing virtual servers based on the policies bound to the content switching virtual server.

For continuous deployments, content switching is used to select the load balancing virtual server serving various versions of an application.

In content switching, the selection of a load balancing virtual server for a specific application version changes at runtime because of the change in the content switching policies. During this transition,

if some sessions are present with older versions of the application, such traffic must continue to be served by older versions only. To support the requirement, the Citrix ADC appliance maintains persistence across multiple load balancing groups behind a content switching virtual server. Persistence for content switching virtual server enables seamless transition of clients from one version to another.

Supported persistence types on content switching virtual server

The following persistence types are supported on content switching virtual servers.

Persistence type	Description
Source IP	**SOURCEIP**. Connections from the same client IP address are parts of the same persistence session. For more details, see Source IP address persistence.
HTTP Cookie	**COOKIEINSERT**. Connections that have the same HTTP Cookie header are parts of the same persistence session. The format of the cookie that the Citrix ADC appliance inserts is: **NSC_<vid_str of CSvserver>=<vid_str of Lbvserver>** where NSC_XXXX is the virtual server ID that is derived from the virtual server name. For more details, see HTTP cookie persistence.
SSL Session ID	**SSLSESSION**. Connections that have the same SSL Session ID are parts of the same persistence session. For more details, see SSL session ID persistence.

You can configure a timeout value for persistence that is based on HTTP cookies. If you set the timeout value to 0, the ADC appliance does not specify the expiration time, regardless of the HTTP cookie version used. The expiration time then depends on the client software, and such cookies are valid only if the software is running.

Depending on the type of persistence that you have configured, the virtual server can support either 250,000 simultaneous persistent connections or any number of persistent connections up to the limits imposed by the amount of memory on your Citrix ADC appliance. The following table shows which types of persistence fall into each category.

Persistence type	Number of simultaneous persistent connections supported
Source IP, SSL Session ID	250,000
HTTP Cookie	Memory limit. In CookieInsert, if the timeout is not 0, the number of connections is limited by memory.

Some types of persistence are specific to particular types of virtual server. The following table lists each type of persistence and indicates which types of persistence are supported on which types of virtual server.

Persistence type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEI	Yes	Yes	Yes	Yes	Yes	Yes	No	No
COOKIEINSERT	Yes	Yes	No	No	No	No	No	No
SSLSESSI	No	Yes	No	No	Yes	Yes	No	No

Backup persistence support

You can configure the content switching virtual server to use the source IP persistence type as the backup persistence type when the cookie persistence type fails. It is useful for canary deployments in the microservices architecture.

When the cookie persistence type fails, the appliance falls back to source IP based persistence only when the client browser does not return any cookie in the request. However, if the browser returns a cookie (not necessarily the persistence cookie) it is assumed that the browser supports cookies and hence backup persistence is not triggered.

You can also set a timeout value for backup persistence. Timeout is the time period for which a persistence session is in effect.

How persistence on content switching virtual server works

Scenario 1: A content switching virtual server without persistence

The following example illustrates the deployment of multiple versions of an application with a content switching virtual server without persistence.



When client C1 sends a request to the application, the request is sent to the content switching virtual server in the Citrix ADC appliance. The content switching virtual server evaluates the policy and forwards the request to the load balancing virtual server (LB1) that is serving version v1 of the application.



Consider a new version v2 of the application is deployed and has to be exposed to a subset of users. The new load balancing virtual server (LB2) serving the v2 version is bound to the content switching virtual server by the appropriate content switching policy.

When client C1 sends a new request, the policy is evaluated again and the request is forwarded to the load balancing virtual server LB2. Thus, the transactions for stateful applications fail if multiple versions of the application are deployed.

Scenario 2: Content switching virtual server with persistence

The following example illustrates the deployment of multiple versions of the application with a content switching virtual server with persistence.



When client C1 sends a request to the application, the request is sent to the content switching virtual server in the Citrix ADC appliance. The content switching virtual server evaluates the policy, creates a persistence session entry, and forwards the request to the load balancing virtual server LB1 that is serving version v1 of the application.



The same client C1 requests again for the application, and the request is sent to the content switching virtual server in the Citrix ADC appliance. A lookup for the persistence session is done, and the

load balancing virtual server LB1 is taken from the existing persistence session and the request is forwarded to LB1. No breakage of the existing transaction happens with this solution; thus, maintaining the stateful nature of the application.



Let's consider a new client C2. The new request C2 is sent to the newer version of the application through policy evaluation as there is no existing persistence session for this client. It results in a successful rollout of the newer version of the application without breaking its statefulness.

Because of the persistence support, customers can deploy multiple content or different versions of the application seamlessly without impacting the existing transactions, specifically for stateful applications. It is not possible without persistence in the picture.

Configure persistence type on content switching virtual server by using the CLI

At the command prompt, type:

```
set cs vserver <name> -PersistenceType <type> [-timeout <integer>]
<!--NeedCopy-->
```

Example:

```
set cs vserver Vserver-CS-1 -persistenceType SOURCEIP -timeout 60
<!--NeedCopy-->
```


Configure persistence type on content switching virtual server by using the GUI

1. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and click **Add**.
2. In **Basic Settings**, configure the persistence details.

Troubleshooting

September 14, 2021

If the content switching feature does not work as expected after you have configured it, you can use some common tools to access Citrix ADC resources and diagnose the problem.

Resources for troubleshooting content switching

For best results, use the following resources to troubleshoot a content switching issue on a Citrix ADC appliance:

- Configuration file
- Relevant `newnslog` file
- Trace files
- Network topology diagram for the network setup of the customer
- Citrix documentation, such as release notes, Knowledge Center articles, and Product documentation.

In addition to the preceding resources, the following tools expedite troubleshooting:

- The `iehttpheaders` or a similar utility
- The Wireshark application customized for the Citrix ADC trace files
- An SSH utility for command line access
- A HyperTerminal utility to access the console

Troubleshooting content switching issues

The most common content switching issues involve the content switching feature not working at all, or working only intermittently, and Service Unavailable responses.

- **Issue**

  The content switching feature is not functioning.

  Resolution

  Check the configuration as follows:

- Verify that the appliance is licensed for content switching.
- Verify that the feature is enabled.
- From the configuration file, verify that valid content switching policies are correctly bound to the load balancing virtual servers.

- **Issue**

  Client receives a 503 - Service Unavailable response.

  Resolution

  - Verify the URL and policy bindings. The client receives the 503 response when none of the policies you have configured is evaluated and no default load balancing virtual server is defined and bound to the content switching virtual server.
  - From the configuration, verify the policies and the URL is accessed by the client.
  - Verify that for every type of request the respective policy is evaluated. If the policy is not evaluated, check the policy expression and update it if necessary.
  - Verify the URL and HTTP request and response headers. To do so, record an `HTTPHeader` trace and, if necessary, record the packet traces on the appliance and the client.

- **Issue**

  Intermittently, the content switching feature is not working as expected.

  Resolution

  - Study the network topology diagram, if available, of the setup to understand the various devices installed between the client and the servers.
  - Verify the configuration and policy bindings. Make sure that the URL in the policy expression matches to the one in the client request.
  - Verify that appropriate priorities are assigned to the policies. An incorrect precedence or priority assigned to a policy can cause a problem.
  - Run the following commands to verify the bindings and the values of the policy select counters in the output of the commands:

    ```
    show cs vserver \<CS VServer\>

    show cs policy \<CS Policy\>

    stat cs vserver \<CS VServer\>
    ```

  - Using `iehttpheaders` or a similar utility, determine whether the HTTP headers for the requests or responses provide any pointers to the issue.
  - Check the release notes and Knowledge Center articles.
  - If the issue is still not resolved, contact Citrix Technical Support with appropriate data for further investigation.

DataStream

September 14, 2021

The Citrix ADC DataStream feature provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a Citrix ADC appliance ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and based on database names, user names, character sets, and packet size.

You can configure load balancing to switch requests based on load balancing algorithms. Alternately, you can elaborate the switching criteria by configuring content switching to make a decision based on an SQL query parameter. You can further configure monitors to track the state of database servers.

> **Note**
>
> Citrix ADC DataStream is supported only for MySQL and MS SQL databases. For information about the supported protocol version, character sets, special queries, and transactions, see DataStream Reference.

How DataStream works

In DataStream, the ADC appliance is placed in-line between the application or Web servers and the database servers. On the appliance, the database servers are represented by services.

A typical DataStream deployment consists of the entities described in the following diagram.

Figure 1. DataStream Entity Model



As shown in this figure, a DataStream configuration can consist of:

- An optional content switching virtual server (CS).
- A load balancing setup consisting of load balancing virtual servers (LB1 and LB2).
- Services (Svc1, Svc2, Svc3, and Svc4).
- Content switching policies (optional).

The clients (application or Web servers) send requests to the IP address of a content switching virtual server (CS) configured on the Citrix ADC appliance. The appliance, then, authenticates the clients using the database user credentials configured on the appliance. The content switching virtual server (CS) applies the associated content switching policies to the requests. After evaluating the policies, the content switching virtual server (CS) routes the requests to the appropriate load balancing virtual server (LB1 or LB2). Then, the load balancing virtual server distributes the requests to the appropriate database servers (represented by services on the appliance) based on the load balancing algorithm. The Citrix ADC appliance uses the same database user credentials to authenticate the connection with the database server.

If a content switching virtual server is not configured on the appliance, the clients (application or Web servers) send their requests to a load balancing virtual server configured on the appliance. The Citrix ADC appliance authenticates the client by using the database user credentials configured on the ap-

pliance, and then uses the same credentials to authenticate the connection with the database server. The load balancing virtual server distributes the requests to the database servers according to the load balancing algorithm. The most effective load balancing algorithm for database switching is the least connection method.

DataStream uses connection multiplexing to enable multiple client-side requests to be made over the same server-side connection. The following connection properties are considered:

- User name
- Database name
- Packet size
- Character set

Configure database users

September 14, 2021

In databases, a connection is always stateful, which means that when a connection is established, it must be authenticated.

Configure your database user name and password on the NetScaler appliance. For example, if you have a user John configured on the database, you need to configure the user John on the ADC too. Adding database user names and passwords on the ADC adds them to the `nsconfig` file.

> **Note**
>
> Names are case sensitive.

The ADC uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

Add a database user by using the CLI

At the command prompt, type

```
add db user <username> - password <password>
```

Example:

```
1 add db user nsdbuser -password dd260427edf
2 <!--NeedCopy-->
```


Add a database user by using the GUI

Navigate to **System > User Administration > Database Users**, and configure a database user.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the ADC appliance.

Reset the password of a database user by using the CLI

At the command prompt, type

```
set db user <username> -password <password>
<!--NeedCopy-->
```

Example:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

Reset the password of database users by using the GUI

Navigate to **System > User Administration > Database Users**, select a user, and enter new values for the password.

If a database user no longer exists on the database server, you can remove the user from the ADC appliance. However, if the user continues to exist on the database server and you remove the user from the ADC appliance, any request from the client with this user name does not get authenticated. As a result, the request is not routed to the database server.

Remove a database user by using the CLI

At the command prompt, type

```
rm db user <username>
<!--NeedCopy-->
```

Example:

```
rm db user nsdbuser
<!--NeedCopy-->
```

Remove a database user by using the GUI

Navigate to **System > User Administration > Database Users**, select a user, and click **Delete**.

Configure a database profile

September 14, 2021

A database profile is a named collection of parameters that is configured once but applied to multiple virtual servers that require those particular parameter settings. After creating a database profile, you bind it to load balancing or content switching virtual servers. You can create as many profiles as you need.

Create a database profile by using the CLI

At the command line, type the following commands to create a database profile and verify the configuration:

```
add db dbProfile <name> [-interpretQuery ( YES | NO )] [-stickiness (
    YES | NO )] [-kcdAccount <string>]

show db dbProfile
<!--NeedCopy-->
```

Example:

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -
    kcdAccount mykcdaccnt
 Done
> show dbProfile myDBProfile
    Name: myDBProfile
    Interpret Query: YES
    Stickyness: YES
    KCD Account: mykcdaccnt
    Reference count: 0

 Done
>
<!--NeedCopy-->
```

Create a database profile by using the GUI

Navigate to **System > Profiles** and, on the **Database Profiles** tab, configure a database profile.

Bind a database profile to a load balancing or content switching virtual server by using the CLI

At the command line, type:

```
set (lb | cs) vserver <name> -dbProfileName <string>
<!--NeedCopy-->
```

Bind a database profile to a load balancing or content switching virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers or Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Profiles** and, in the **DB Profile** list, select a profile to bind to the virtual server. To create a profile, click plus (+).

Configure load balancing for DataStream

September 14, 2021

Before configuring a load balancing setup, you must enable the load balancing feature. Then, begin by creating at least one service for each database server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server and bind the services to the virtual server.

> **Note**:
>
> For databases, the load balancing can only occur on homogenous database servers (database servers that contain the exact same databases). For a configuration that contains unique databases on different servers, you must use content switching. If some of your database servers host identical content, you can use load balancing on those servers only. You can then use content switching policies to send requests to the load balancing virtual server that manages load balancing for those databases.
>
> The Citrix ADC appliance currently stores the database name and login information during the database session. When a query is made to the database, it uses that information to connect to the specific database server.

Parameter values specific to DataStream

- Protocol

Use the MYSQL protocol type for MySQL databases and MSSQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see http://dev.mysql.com/doc/internals/en/client-server-protocol.html. For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

- Port

  Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

- Method

  It is recommended that you use the Least Connection method for better load balancing and lower server load. However, other methods, such as Round Robin, Least Response Time, Source IP Hash, Source IP Destination IP Hash, Least Bandwidth, Least Packets, and Source IP Source Port Hash, are also supported.

  Note: URL Hash method is not supported for DataStream.

- MS SQL Server version

  If you are using the Microsoft SQL Server, and you expect some clients running a different version from your Microsoft SQL Server product, set the Server Version parameter for the load balancing virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the Server Version parameter, see Configuring the MySQL and Microsoft SQL server version setting.

- MySQL Server version

  If you are using the MySQL Server, and you expect some clients running a different version from your MySQL Server product, set the Server Version parameter for the load balancing virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the Server Version parameter, see Configuring the MySQL and Microsoft SQL server version setting.

Configure content switching for DataStream

September 14, 2021

You can segment traffic according to information in the SQL query, based on database names, user names, character sets, and packet size.

You can configure content switching policies with default syntax expressions to switch content based on connection properties. For example, user name and database name, command parameters, and the SQL query to select the server.

The default syntax expressions evaluate traffic associated with MYSQL and MS SQL database servers. Use request-based expressions in default syntax policies to make request switching decisions at the content switching virtual server bind point. Use response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

For information about default syntax expressions, see Default Syntax Expressions: DataStream.

> **Note:**
>
> For databases, the load balancing can only occur on homogenous database servers (database servers that contain the exact same databases). For a configuration that contains unique databases on different servers, you must use content switching. If some of your database servers host identical content, you can use load balancing on those servers only. You can then use content switching policies to send requests to the load balancing virtual server that manages load balancing for those databases.
>
> The Citrix ADC appliance currently stores the database name and login information during the database session. When a query is made to the database, it uses that information to connect to the specific database server.

Parameter values specific to DataStream

- Protocol

  Use the MYSQL protocol type for MySQL databases and MSSQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see http://dev.mysql.com/doc/internals/en/client-server-protocol.html. For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

- Port

  Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

- MS SQL Server Version

  If you use Microsoft SQL Server, and expect some clients running a different version from your Microsoft SQL Server product, set the Server Version parameter for the content switching virtual server. The version setting provides compatibility between the client-side and server-side

connections by ensuring that all communication conforms to the server's version. For more information about setting the Server Version parameter, see Configuring the Microsoft SQL Server Version Setting.

Configure monitors for DataStream

September 14, 2021

To track the state of each load balanced database server in real time, you need to bind a monitor to each service. The monitor is configured to test the service by sending periodic probes to the service, sometimes referred to as performing a health check. If the monitor receives a timely response to its probes, it marks the service as UP. If it does not receive a timely response to the designated number of probes, it marks the service as DOWN.

For DataStream, you need to use the built-in monitors: MYSQL-ECV and MSSQL-ECV. Using this monitor you can send an SQL request and parse the response for a string.

Before configuring monitors for DataStream, you must add database user credentials to your NetScaler appliance. For information about configuring monitors, see Configure monitors in a load balancing setup.

When you create a monitor, a TCP connection is established with the database server, and the connection is authenticated by using the user name provided while creating the monitor. You can then run an SQL query to the database server and evaluate the server response to check whether it matches the configured rule.

The following examples are for MYSQL servers.

Examples:

In the following example, the value of the error message is evaluated to determine the state of the server.

```
1 add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from
2 table2;" -evalrule "mysql.res.error.message.contains("Invalid
3 User")"-database "NS" -userName "user1"
4 <!--NeedCopy-->
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
1 add lb monitor lb_mon4 MYSQL-ECV -sqlQuery "select * from
2 table4;" -evalrule "mysql.res.atleast_rows_count(7)" -database "NS" -
     userName "user2"
```



```
<!--NeedCopy-->
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

```
add lb monitor lb_mon3 MYSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mysql.res.row(1).double_elem
   (2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```

The following examples are for MSSQL servers.

Examples:

In the following example, the value of the error message is evaluated to determine the state of the server.

```
add lb monitor lb_mon1 MSSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mssql.res.error.message.contains("Invalid
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
add lb monitor lb_mon4 MSSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mssql.res.atleast_rows_count(7)" -database "NS" -
   userName "user2"
<!--NeedCopy-->
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

```
add lb monitor lb_mon3 MSSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mssql.res.row(1).double_elem
   (2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```


Use Case 1: Configure DataStream for a primary/secondary database architecture

September 14, 2021

A commonly used deployment scenario is the primary/secondary database architecture where the primary database replicates all information to the secondary databases.

For primary/secondary database architecture, you might want all WRITE requests to be sent to the primary database and all READ requests to the secondary databases.

The following figure shows the entities and the values of the parameters you need to configure on the appliance.

Figure 1. DataStream Entity Model for Primary/Secondary Database Setup



In this example scenario, a service (Svc_mysql_1) is created to represent the primary database and is bound to a load balancing virtual server (Lb_vsr_mysql_primary). Three more services (Svc_mysql_2, Svc_mysql_3, and Svc_mysql_4) are created to represent the three secondary databases, and they are bound to another load balancing virtual server (Lb_vsr_mysql_secondary).

A content switching virtual server (Cs_vsr_mysql_1) is configured with associated policies to send all WRITE requests to the load balancing virtual server, Lb_vsr_mysql_primary. All READ requests are sent to the load balancing virtual server, Lb_vsr_mysql_secondary.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. After evaluating the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

The following table lists the names and values of the entities and the policy configured on the Citrix ADC appliance.

Entity Type	Name	IP Address	Protocol	Port	Expression
Services	Svc_mysql_1	198.51.100.5	MYSQL	3306	NA
	Svc_mysql_2	198.51.100.6	MYSQL	3306	NA
	Svc_mysql_3	198.51.100.7	MYSQL	3306	NA
	Svc_mysql_4	198.51.100.8	MYSQL	3306	NA
Load balancing virtual servers	Lb_vsr_mysql_	198.51.100.201	MYSQL	3306	NA
	Lb_vsr_mysql_secondary	198.51.100.202	MYSQL	3306	NA
Content switching virtual server	Cs_vsr_mysql_	198.51.100.161	MYSQL	3306	NA
Content switching policy	Cs_select	NA	NA	NA	`MYSQL.REQ.QUERY.COMMAND.contains("select")`

Table 1. Entity and Policy Names and Values

To configure DataStream for a primary/secondary database setup by using the command line interface

At the command prompt, type


```
add service Svc_mysql_1 198.51.100.5 mysql 3306

add service Svc_mysql_2 198.51.100.6 mysql 3306

add service Svc_mysql_3 198.51.100.7 mysql 3306

add service Svc_mysql_4 198.51.100.8 mysql 3306

add lb vserver Lb_vsr_mysql_primary mysql 198.51.100.201 3306

add lb vserver Lb_vsr_mysql_secondary mysql 198.51.100.202 3306

bind lb vserver Lb_vsr_mysql_primary svc_mysql_1

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_2

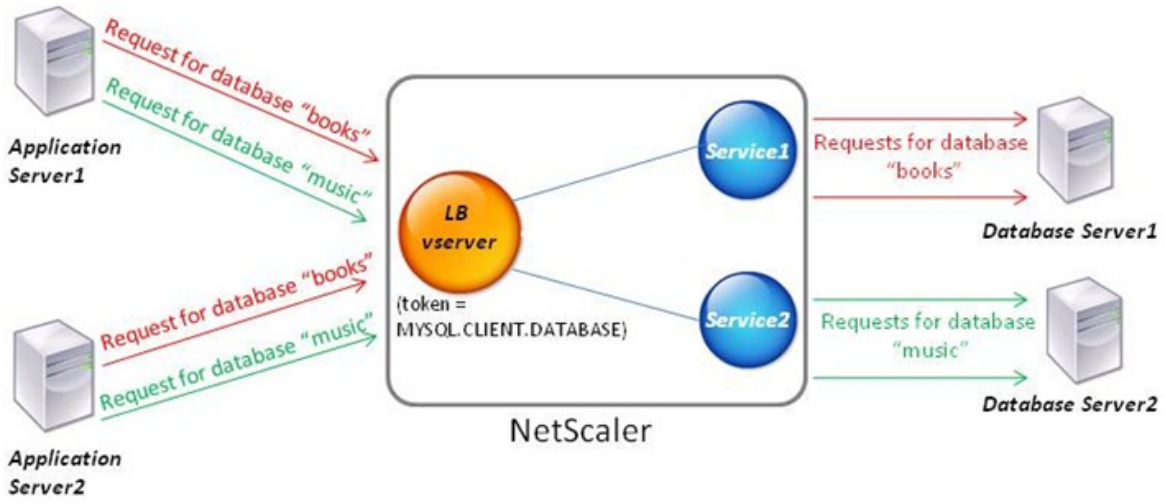
bind lb vserver Lb_vsr_mysql_secondary svc_mysql_3

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_4

add cs vserver Cs_vsr_mysql_1 mysql 198.51.100.161 3306

add cs policy Cs_select -rule "MYSQL.REQ.QUERY.COMMAND.contains(\"
   select"\)"

bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_primary

bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_secondary -policy
   Cs_select -priority 10
<!--NeedCopy-->
```

Use Case 2: Configure the token method of load balancing for DataStream

September 14, 2021

You can configure the token method of load balancing for DataStream to base the selection of database servers on the value of the token extracted from the client (application or web server) requests. These tokens are defined by using SQL expressions. For subsequent requests with the same token, the Citrix ADC appliance sends the requests to the same database server that handled the initial request. Requests with the same token are sent to the same database server until the

maximum connection limit is reached or the session entry has aged out.

You can use the following sample SQL expressions to define tokens:

MySQL	MS SQL
MYSQL.REQ.QUERY.TEXT	MSSQL.REQ.QUERY.TEXT
MYSQL.REQ.QUERY.TEXT(n)	MSSQL. REQ.QUERY.TEXT(n)
MYSQL.REQ.QUERY.COMMAND	MSSQL.REQ.QUERY.COMMAND
MYSQL.CLIENT.USER	MSSQL.CLIENT.USER
MYSQL.CLIENT.DATABASE	MSSQL.CLIENT.DATABASE
MYSQL.CLIENT.CAPABILITIES	

The following example shows how the Citrix ADC DataStream feature works when you configure the token method of load balancing.

Figure 1. DataStream and the Token Method of Load Balancing



In this example, the token is the name of the database. A request with token books is sent to Database Server1 and a request with token music is sent to Database Server2. All subsequent requests with token books are sent to Database Server1 and requests with token music are sent to Database Server2. This configuration provides pseudo persistence with the database servers.

Configure this example by using the CLI

At the command prompt, type:


```
add service Service1 192.0.2.9 MYSQL 3306

add service Service2 192.0.2.11 MYSQL 3306

add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -
    rule MYSQL.CLIENT.DATABASE

bind lb vserver token_lb_vserver Service1

bind lb vserver token_lb_vserver Service2
<!--NeedCopy-->
```

Configure this example by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, configure a virtual server, and specify the protocol as **MYSQL**.
2. Click in the **Service** section, and configure two services specifying the protocol as MYSQL. Bind these services to the virtual server.
3. In **Advanced Settings**, click **Method** and, in the **Load Balancing Method** list, select **TOKEN** and specify the expression as **MYSQL.CLIENT.DATABASE**.

Use Case 3: Log MSSQL transactions in transparent mode

September 14, 2021

You can configure the Citrix ADC appliance to operate transparently between MSSQL clients and servers, and to only log or analyze details of all client-server transactions. Transparent mode is designed so that the Citrix ADC appliance only forwards MSSQL requests to the server, and then relays the server's responses to the clients. As the requests and responses pass through the appliance, the appliance logs information gathered from them, as specified by the audit logging or AppFlow configuration, or collects statistics, as specified by the Action Analytics configuration. You do not have to add database users to the appliance.

When operating in transparent mode, the Citrix ADC appliance does not perform load balancing, content switching, or connection multiplexing for the requests. However, it responds to a client's pre-login packet on behalf of the server so that it can prevent encryption from being agreed upon during the pre-login handshake. The login packet and subsequent packets are forwarded to the server.

Summary of configuration tasks

For logging or analyzing MSSQL requests in transparent mode, you have to do the following:

- Configure the Citrix ADC appliance as the default gateway for both clients and servers.
- Do one of the following on the Citrix ADC appliance:
  - **Configure the use source IP address (USIP) option globally:** Create a load balancing virtual server with a wildcard IP address and the port number on which the MSSQL servers listen for requests (a port-specific wildcard virtual server). Then, enable the USIP option globally. If you configure a port-specific wildcard virtual server, you do not have to create MSSQL services on the appliance. The appliance discovers the services based on the destination IP address in the client requests.
  - **If you do not want to configure the USIP option globally:** Create MSSQL services with the USIP option enabled on each of them. If you configure services, you do not have to create a port-specific wildcard virtual server.
- Configure audit logging, AppFlow, or Action Analytics to log or collect statistics about the requests. If you configure a virtual server, you can bind your policies either to the virtual server or to the global bind point. If you do not configure a virtual server, you can bind your policies to only the global bind point.

Configure transparent mode by using a wildcard virtual server

You can configure transparent mode by configuring a port-specific wildcard virtual server and enabling Use Source IP (USIP) mode globally. When a client sends its default gateway (the Citrix ADC appliance) a request with the IP address of an MSSQL server in the destination IP address header, the appliance checks whether the destination IP address is available. If the IP address is available, the virtual server forwards the request to the server. Otherwise, it drops the request.

Create a wildcard virtual server by using the CLI

At the command prompt, type the following commands to create a wildcard virtual server and verify the configuration:

```
add lb vserver <name> <serviceType> <IPAddress> <port>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
> add lb vserver wildcardLbVs MSSQL * 1433
 Done
> show lb vserver wildcardLbVs
```



```
    wildcardLbVs (*:1433) - MSSQL   Type: ADDRESS
    State: UP
    . . .

 Done
>
<!--NeedCopy-->
```

Create a wildcard virtual server by using the GUI

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and create a virtual server. Specify MSSQL as the protocol and * as the IP address.

Enable Use Source IP (USIP) mode globally by using the CLI

At the command prompt, type the following commands to enable USIP mode globally and verify the configuration:

```
enable ns mode USIP

show ns mode
<!--NeedCopy-->
```

Example:

```
> enable ns mode USIP
 Done
> show ns mode

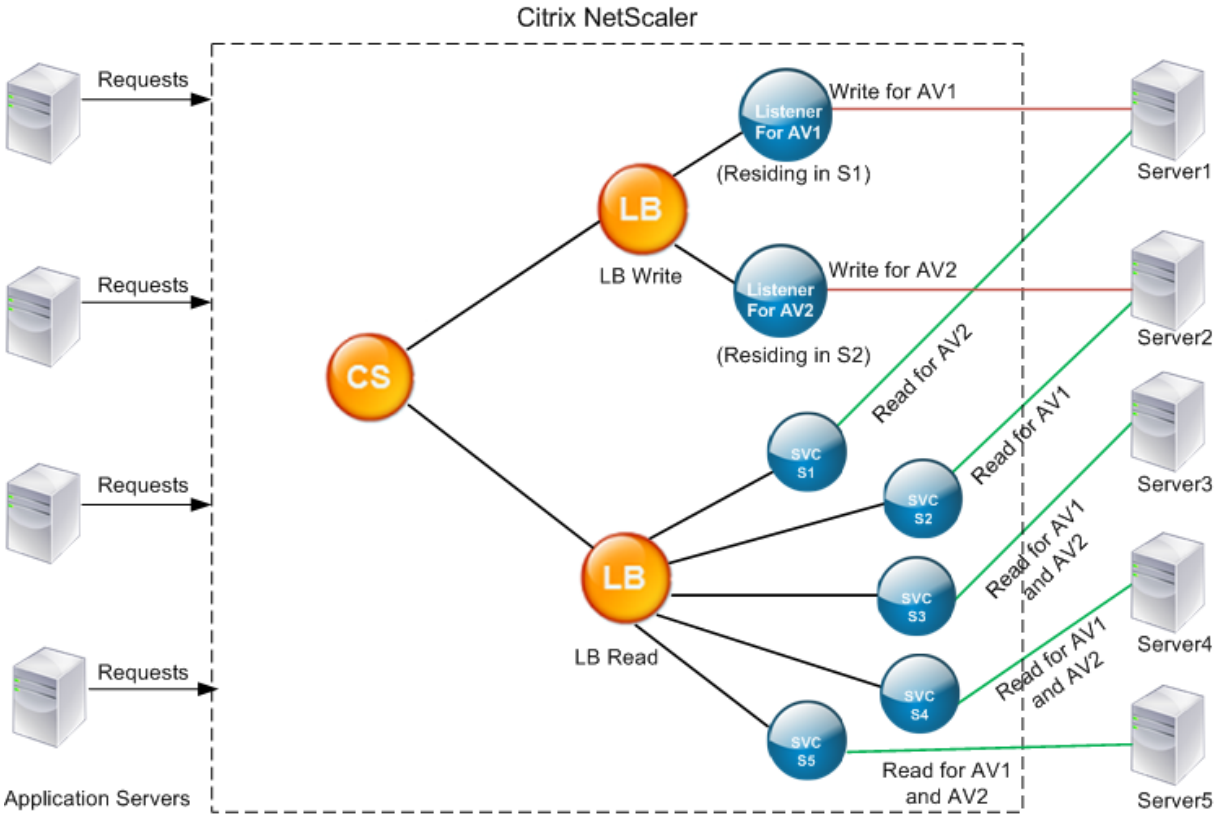
    Mode                              Acronym
       Status
    -------                           -------
       ------
                . . .
 3) Use Source IP                  USIP                            ON
                . . .
 Done
>
<!--NeedCopy-->
```

Enable USIP mode globally by using the GUI

1. Navigate to **System > Settings** and, in Modes and Features, select **Configure Modes**.

2. Select **Use Source IP**.

Configure transparent mode by using MSSQL services

You can configure transparent mode by configuring MSSQL services and enabling USIP on each service. When a client sends its default gateway (the Citrix ADC appliance) a request with the IP address of an MSSQL server in the destination IP address header, the appliance forwards the request to the destination server.

Create an MSSQL service and enable USIP mode on the service by using the CLI

At the command prompt, type the following commands to create an MSSQL service, with USIP enabled, and verify the configuration:

```
add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES
    `

show service <name>
<!--NeedCopy-->
```

Example

```
> add service myDBservice 192.0.2.0 MSSQL 1433 -usip YES
 Done
> show service myDBservice
    myDBservice (192.0.2.0:1433) - MSSQL
    State: UP
                . . .
    Use Source IP: YES      Use Proxy Port: YES
                . . .
 Done
>
<!--NeedCopy-->
```

Create an MSSQL service, with USIP enabled, by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and configure a service.
2. Specify the protocol as **MSSQL** and, in **Settings**, select **Use Source IP**.

Use Case 4: Database specific load balancing

September 14, 2021

A database server farm must be load balanced not only based on the states of the servers, but also based on the availability of the database on each server. A service might be up, and a load balancing device might show it as being in the UP state, but the requested database might be unavailable on that service. The request is not served if a query is forwarded to a service on which the database in unavailable. Therefore, a load balancing device must be aware of the availability of a database on each service. And when making a load balancing decision, it must consider only those services on which the database is available.

As an example, consider that database servers server1, server2, and server3 host databases mydatabase1 and mydatabase2. If mydatabase1 becomes unavailable on server2, the load balancing device must be aware of that change in state. It must load balance requests for mydatabase1 across only server1 and server3. After mydatabase1 becomes available on server2, the load balancing device must include server2 in load balancing decisions. Similarly, if mydatabase2 becomes unavailable on server3, the device must load balance requests for mydatabase2 across only server1 and server2. It must include server3 in its load balancing decisions only when mydatabase2 becomes available. This load balancing behavior must be consistent across all the databases that are hosted on the server farm.

The Citrix ADC appliance implements this behavior by retrieving a list of all the databases that are active on a service. To retrieve the list of active databases, the appliance uses a monitor that is configured with an appropriate SQL query. If the requested database is unavailable on a service, the appliance excludes the service from load balancing decisions until it becomes available. This behavior ensures uninterrupted service to clients.

> **Note**
>
> Database specific load balancing is supported for only MSSQL and MySQL service types. This support is also available for Microsoft SQL Server 2012 high availability deployment.

To set up database specific load balancing, you must configure the following:

- Enable the load balancing feature, and configure a load balancing virtual server of type MSSQL or MySQL.
- Configure the services that host the database, and bind the services to the virtual server. The monitor needs valid user credentials to log on to the database server, so you must configure a database user account on each of the servers and then add the user account to the Citrix ADC appliance.
- Then, you configure an MSSQL-ECV or MYSQL-ECV monitor and bind the monitor to each service.
- Finally, you must test the configuration to ensure that it is working as intended. Before you

perform these configuration tasks, make sure you understand how database specific load balancing works.

How database specific load balancing works

For database specific load balancing, you configure a monitor that periodically queries each database server for the names of all the active databases on it. The Citrix ADC appliance stores the results, and regularly updates the records based on the information retrieved through monitoring. When a client queries a particular database, the appliance uses the configured load balancing method to select a service, and then checks its records to determine whether the database is available on that service. If the records indicate that the database is not available, it uses the configured load balancing method to select the next available service, and then repeats the check. The appliance forwards the query to the first available service on which the database is active.

Enable load balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled. The entities do not function until you enable the feature.

Enable load balancing by using the CLI

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature LB

show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
```



```
 24)    NetScaler Push                  push                  OFF
 Done
<!--NeedCopy-->
```

Enable load balancing by using the GUI

Navigate to **System > Settings** and, in **Configure Basic Features**, select **Load Balancing**.

Configure a load balancing virtual server for database specific load balancing

To configure a virtual server to load balance databases based on availability, you enable the database specific load balancing parameter on the virtual server. Enabling the parameter modifies the load balancing logic so that the Citrix ADC appliance refers the results of the monitoring probe sent to the selected service, before forwarding the query to that service.

Configure a load balancing virtual server for database specific load balancing using the CLI

At the command prompt, type the following command to configure a load balancing virtual server for database specific load balancing and verify the configuration:

```
add lb vserver <name> <serviceType> <ipAddress> <port> -dbsLb ENABLED

show lb vserver <name>
<!--NeedCopy-->
```

Configure services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the Citrix ADC appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served.

If you create a service without first creating a server object, the IP address of the service is also the name of the server that hosts the service. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

Configure database users

In databases, a connection is always stateful, which means that when a connection is established, it must be authenticated.

Configure your database user name and password on the Citrix ADC. For example, if you have a user John configured on the database, you need to configure the user John on the ADC too. Database user names and passwords added to the ADC are added to the `nsconfig` file.

> **Note**
>
> Names are case sensitive.

The ADC uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

Add a database user by using the CLI

At the command prompt, type

```
add db user <username> - password <password>
<!--NeedCopy-->
```

Example:

```
add db user nsdbuser -password dd260427edf
<!--NeedCopy-->
```

Add a database user by using the GUI

Navigate to **System > User Administration > Database Users**, and configure a database user.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the Citrix ADC appliance.

Reset the password of a database user by using the CLI

At the command prompt, type

```
set db user <username> -password <password>
<!--NeedCopy-->
```

Example:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```


Reset the password of database users by using the GUI

Navigate to **System > User Administration > Database Users**, select a user, and enter new values for the password.

If a database user no longer exists on the database server, you can remove the user from the Citrix ADC appliance. However, if the user continues to exist on the database server and you remove the user from the ADC appliance, any request from the client with this user name does not get authenticated. Therefore, the user name does not get routed to the database server.

Remove a database user by using the CLI

At the command prompt, type

```
rm db user <username>
<!--NeedCopy-->
```

Example:

```
rm db user nsdbuser
<!--NeedCopy-->
```

Remove a database user by using the GUI

Navigate to **System > User Administration > Database Users**, select a user, and click **Delete**.

Configure a monitor to retrieve the names of active databases

You can create a monitor to retrieve the list of all active databases on a database instance. The monitor logs on to the database server by using a valid user credentials and runs an appropriate SQL query. The SQL query you need to use depends on your SQL server deployment. For example, in an MSSQL database mirroring setup, you can use the following query to retrieve a list of active databases available on a server instance.

```
select name from sys.databases where state=0
<!--NeedCopy-->
```

In a MySQL database setup you can use the following queries to retrieve a list of active databases available on a server instance.

Show databases:

You also configure the monitor to evaluate the response for an error condition, and to store the results if there is no error. If the response contains an error, the monitor marks the service as DOWN. The appliance excludes the service from load balancing decisions until an error is no longer returned.

> **Note**
>
> The database specific load balancing feature is supported only for the MSSQL and MySQL service types. Therefore, the monitor type must be MSSQL-ECV or MYSQL-ECV.

Configure a monitor to retrieve the names of all the active databases hosted on a service by using the CLI

At the command prompt, type the following commands to retrieve the names of all the active databases hosted on a service and verify the configuration:

```
add lb monitor <monitorName> <type> -userName <string> -sqlQuery <text>
    -evalRule <expression> -storedb ENABLED

show lb monitor <monitorName>
<!--NeedCopy-->
```

Configure a monitor to retrieve the names of all the active databases hosted on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors** and configure a monitor of type MSSQL-ECV or MYSQL-ECV.
2. In **Special Parameters**, specify a user name, query, and a rule. For example, for MSSQL-ECV, the query must be “select name from sys.databases where state=0”), and a rule must be MSSQL.RES.TYPE.NE(ERROR). For MYSQL-ECV, the query must be “show databases” and a rule must be MYSQL.RES.TYPE.NE(ERROR).

Availability groups deployment support for MSSQL

Consider the following scenario in which database specific load balancing is configured in a high availability group deployment. S1 through S5 is the services on the ADC appliance. DB1 through DB4 is the databases on the servers represented by the services S1 through S5. AV1 and AV2 are the availability groups. Each availability group contains up to one primary database server instance and up to four secondary database server instances. A service, representing the servers in the availability group, can be primary for one availability group and secondary for another availability group. Each availability group contains different databases and one listener, which is a service. All requests arrive on the listener service that resides on the primary database. AVI contains databases DB1 and DB2. AV2 contains

databases DB3 and DB4. L1 and L2 are the listeners on AV1 and AV2 respectively. S1 is the primary service for AV1 and S2 is the primary service for AV2.



Service	List of Active Databases on the Service
S1	DB1, DB2, DB3, DB4
S2	DB3, DB4
S3	DB3, DB4
S4	DB1, DB2
S5	DB1, DB2

Availability Group	Databases	Services representing the Servers in the Availability Group
AV1	DB1, DB2	S1, S4, S5
AV2	DB3, DB4	S1, S2, S3

Queries flow as follows:

1. A READ query for AV1 is load balanced between S4 and S5. S1 is the primary for AV1.
2. A WRITE query for AV1 is directed to L1.
3. A READ query for AV2 is load balanced between S1 and S3. S2 is the primary for AV2.
4. A WRITE query for AV1 is directed to L2.

Sample configuration

1. Configure load balancing and content switching virtual servers.
   - `add lb vserver lbwrite -dbslb enabled`
   - `add lbvserver lbread MSSQL -dbslb enabled`
   - `add csvserver csv MSSQL 1.1.1.10 1433`
2. Configure two listener services, one for each availability group, and five services S1 through S5 representing databases DB1 through DB4.
   - `add service L1 1.1.1.11 MSSQL 1433`
   - `add service L2 1.1.1.12 MSSQL 1433`
   - `add service s1 1.1.1.13 MSSQL 1433`
   - `add service s2 1.1.1.14 MSSQL 1433`
   - `add service s3 1.1.1.15 MSSQL 1433`
   - `add service s4 1.1.1.16 MSSQL 1433`
   - `add service s5 1.1.1.17 MSSQL 1433`
3. Bind the services to the load balancing virtual servers.
   - `bind lbvserver lbwrite L1`
   - `bind lbvserver lbwrite L2`
   - `bind lbvserver lbread s1`
   - `bind lbvserver lbread s2`
   - `bind lbvserver lbread s3`
   - `bind lbvserver lbread s4`
   - `bind lbvserver lbread s5`
4. Configure database users.
   - `add db user nsdbuser1 -password dd260427edf`
   - `add db user nsdbuser2 -password ccd1234xyzw`
5. Configure two monitors, monitor_L1 and monitor_L2 for each listener service, to retrieve the list of active databases in that availability group. Add a monitor, monitor1 to retrieve the list of databases for the secondary database server instance.
   - ```
     add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery "
     SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_
      b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners
      c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a
     ```


     ```

```
 d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address
 like '1.1.1.11'"-evalRule "MSSQL.RES.TYPE.NE(ERROR)" -storedb
ENABLED
```

- `add lb monitor monitor_L2 MSSQL-ECV -userNameuser1 -sqlQuery "`
  `SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replicca`
  `b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners`
  `c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a`
  `d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address`
  `like '1.1.1.12'"-evalRule "MSSQL.RES.TYPE.NE(ERROR)"-storedb`
  `ENABLED`
- `add lb monitor monitor1 MSSQL-ECV -userNameuser1 -sqlQuery "SELECT`
  `name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states`
  `b ON a.replica_id=b.replica_id WHERE b.role = 2"-evalRule "MSSQL.`
  `RES.TYPE.NE(ERROR)"-storedb ENABLED`

6. Configure read and write policies.
   - `add cs policy pol_write -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS("`
     `insert")"`
   - `add cs policy pol_read -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS("select`
     `")"`
7. Bind the policies to the content switching virtual server.
   - `bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -`
     `priority 11`
   - `bind csvserver csv -targetLBVserver lbread -policyName pol_read -`
     `priority 12`
8. Bind monitors to the services. Bind monitors to services L1 and L2 to get the list of active databases for the availability group for which it is the listener. Bind monitors to all the services that are bound to the read-only virtual server.
   - `bind service L1 -monitorName monitor_L1`
   - `bind service L2 -monitorName monitor_L2`
   - `bind service s1 -monitorName monitor1`
   - `bind service s2 -monitorName monitor1`
   - `bind service s3 -monitorName monitor1`
   - `bind service s4 -monitorName monitor1`
   - `bind service s5 -monitorName monitor1`

Configuration examples for MSSQL virtual server

To configure a load balancing virtual server for database specific load balancing:

```
1 add lb vserver DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb ENABLED
```



```
2
3 Done
4
5 show lb vserver DBSpecificLB1
6
7 DBSpecificLB1 (192.0.2.10:1433) - MSSQL Type: ADDRESS
8 . . .
9 DBS_LB: ENABLED
10
11 Done
12 <!--NeedCopy-->
```

To configure services:

add service msservice1 5.5.5.5 MSSQL 1433

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line:

```
1 add lb monitor mssql-monitor1 MSSQL-ECV -userName user1 -sqlQuery "
     select name from sys.databases where state=0" -evalRule "MSSQL.RES.
     TYPE.NE(ERROR)" -storedb EN
2
3 Done
4
5 show lb monitor mssql-monitor1
6
7 1)   Name.......: mssql-monitor1    Type......: MSSQL-ECV
8
9 ...
10
11 Special parameters: Database.....:""
12
13 User name.....:"user1"
14
15 Query..:select name from sys.databases where state=0 EvalRule...:MSSQL.
     RES.TYPE.NE(ERROR)
16
17 Version...:70 STORE_DB...:ENABLED
18
19 Done
20 <!--NeedCopy-->
```


Configuration examples for MySQL virtual server

To configure a load balancing virtual server for database specific load balancing:

```
add lb vserver DBSpecificLB1 MYSQL 192.0.2.10 3306 -dbsLb ENABLED

Done

show lb vserver DBSpecificLB1

DBSpecificLB1 (192.0.2.10:3306) - MYSQL Type: ADDRESS

. . .

DBS_LB: ENABLED

Done
<!--NeedCopy-->
```

To configure services:

```
add service msservice1 5.5.5.5 MYSQL 3306
<!--NeedCopy-->
```

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line:

```
add lb monitor mysql-monitor1 MYSQL-ECV -userName user1 -sqlQuery "show
    databases" -evalRule "MYSQL.RES.TYPE.NE(ERROR)" -storedb ENABLED

Done

show lb monitor mysql-monitor1

1)     Name.......: mysql-monitor1  Type......: MYSQL-ECV  State....:
   ENABLED

...

Special parameters: Database.....:""

User name.....:"user1" Query..:show databases

EvalRule...:MYSQL.RES.TYPE.NE(ERROR) STORE_DB...:ENABLED

Done
```



```
<!--NeedCopy-->
```

DataStream reference

September 14, 2021

This reference describes the MySQL and TDS protocols, the database versions, the authentication methods, and the character sets supported by the DataStream feature. It also describes how the Citrix ADC handles transaction requests and special queries that modify the state of a connection.

You can also configure the Citrix ADC appliance to generate audit log messages for the DataStream feature.

Supported database versions, protocols, and authentication methods

	MySQL Database	MS SQL Database
Database Versions	MySQL database versions 4.1, 5.0, 5.1, 5.4, 5.5, 5.6	MS SQL database versions 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014 (Kerberos Authentication support)
Protocols	MySQL protocol version 10. For information about the MySQL protocol, see MySQL Client/Server Protocol	Tabular Data Stream (TDS) protocol version 7.1 and higher. For information about the TDS protocol, see Tabular Data Stream Protocol
Authentication Methods	MySQL native authentication is supported.	SQL server authentication and Windows Authentication (Kerberos/NTLM) are supported.

Character sets

The DataStream feature supports only the UTF-8 charset.

The character set used by the client while sending a request might be different from the character set used in the database server responses. Although the charset parameter is set during the connection establishment, it can be changed at any time by sending an SQL query. The character set is associated with a connection, and therefore, requests on connections with one character set cannot be

multiplexed onto a connection with a different character set.

The Citrix ADC appliance parses the queries sent by the client and the responses sent by the database server.

The character set associated with a connection can be changed after the initial handshake by using the following two queries:

```
SET NAMES <charset> COLLATION <collation>

SET CHARACTER SET <charset>
<!--NeedCopy-->
```

Transactions

In MySQL, transactions are identified by using the connection parameter AUTOCOMMIT or the BEGIN:COMMIT queries. The AUTOCOMMIT parameter can be set during the initial handshake, or after the connection is established by using the query SET AUTOCOMMIT.

The Citrix ADC appliance explicitly parses each query to determine the beginning and end of a transaction.

In the MySQL protocol, the response contains two flags to indicate whether the connection is a transaction: the TRANSACTION and AUTOCOMMIT flags.

If the connection is a transaction, the TRANSACTION flag is set. Or, if the AutoCommit mode is OFF, the AUTOCOMMIT flag is not set. The ADC appliance parses the response, and if either the TRANSACTION flag is set or the AUTOCOMMIT flag is not set, it does not do connection multiplexing. When these conditions are no longer true, the ADC appliance begins connection multiplexing.

> **Note**
>
> Transactions are also supported for MS SQL.

Special queries

There are special queries, such as SET and PREPARE, that modify the state of the connection and might break request switching, and therefore, these queries need to be handled differently.

On receiving a request with special queries, the Citrix ADC appliance sends an OK response to the client and also, stores the request in the connection.

When a non-special query, such as INSERT and SELECT, is received along with a stored query, the ADC appliance looks for the server-side connection on which the stored query has already been sent to the database server. If no such connections exist, the ADC appliance creates a connection, and sends the stored query first, and then, sends the request with the non-special query.

In SET, USE db, and INIT_DB special queries, the appliance modifies a field in the server side connection corresponding to the special query. This modification results in better reuse of the server side connection.

Only 16 queries are stored in each connection.

The following is a list of the special queries for which the ADC appliance has a modified behavior.

- SET query

  The SET SQL queries define variables that are associated with the connection. These queries are also used to define global variables, but as of now, the ADC appliance is unable to differentiate between local and global variables. For this query, the ADC appliance uses the 'store and forward' mechanism.

- USE <db> query

  Using this query, the user can change the database associated with a connection. In this case, the ADC appliance parses the <db> value sent and modifies a field in the server side connection to reflect the new database to be used.

- INIT_DB command

  Using this query, the user can change the database associated with a connection. In this case, the ADC appliance parses the <init_db> value sent and modifies a field in the server side connection to reflect the new database to be used.

- COM_PREPARE

  The ADC appliance stops request switching on receiving this command.

- PREPARE query

  This query is used to create prepared statements that are associated with a connection. For this query, the ADC appliance uses the 'store and forward' mechanism.

Audit log message support

You can now configure the Citrix ADC appliance to generate audit log messages for the DataStream feature. Audit log messages are generated when client-side and server-side connections are established, closed, or dropped. The categories of messages that you can log and view are ERROR and INFO. Error messages for client-side connections begin with "CS" and error messages for server-side connections begin with "SS." Additional information is provided where necessary. For example, log messages for closed connections (CS_CONN_CLOSED) include only the connection ID. However, log messages for established connections (CS_CONN_ESTD) include information such as the user name, database name, and the client IP address in addition to the connection ID.

Domain Name System

September 14, 2021

Note: From release 13.0 build 41.x, the Citrix ADC appliance in ADNS and proxy mode is fully compliant with DNS flag day 2019.

You can configure the Citrix ADC appliance to function as an authoritative domain name server (ADNS server) for a domain. Add the DNS resource records that belong to the domain for which the appliance is authoritative and configure resource record parameters. You can also configure the appliance as a proxy DNS server that load balances a farm of DNS name servers that are either within or outside your network. Configure the appliance as an end resolver and forwarder. You can configure DNS suffixes that enable name resolution when fully qualified domain names are not configured. The appliance also supports the DNS ANY query that retrieves all the records that belong to a domain.

You can configure the appliance to concurrently function as an authoritative DNS server for one domain and a DNS proxy server for another domain. When you configure the appliance as the authoritative DNS server or DNS proxy server for a zone, you can enable the appliance to use the TCP for response sizes that exceed the size limit specified for the User Datagram Protocol (UDP).

How DNS Works on the Citrix ADC

You can configure the Citrix ADC appliance to function as an ADNS server, DNS proxy server, end resolver, and forwarder. You can add DNS resource records on the Citrix ADC appliance, including the following records:

- Service (SRV) records
- IPv6 (AAAA) records
- Address (A) records
- Mail exchange (MX) records
- Canonical name (CNAME) records
- Pointer (PTR) records
- Start of authority (SOA) records
- Text (TXT) records

Also, you can configure the Citrix ADC to load balance external DNS name servers.

The Citrix ADC appliance can be configured as the authority for a domain. Add valid SOA and NS records for the domain.

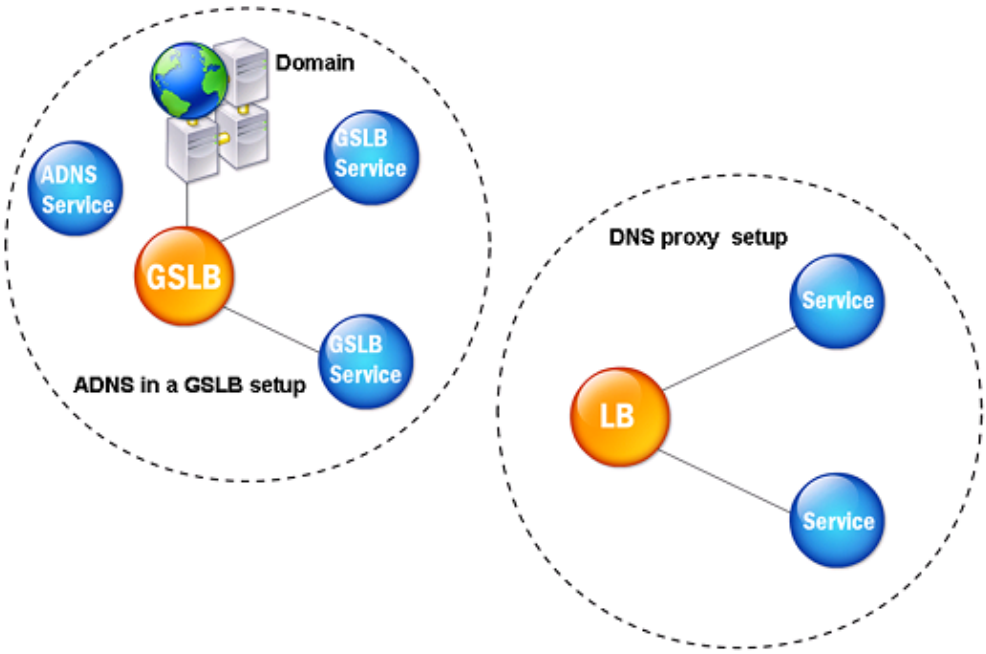
An ADNS server is a DNS server that contains complete information about a zone.

To configure the Citrix ADC appliance as an ADNS server for a zone, you must add an ADNS service, and then configure the zone. To do so, you add valid SOA and NS records for the domain. When a

client sends a DNS request, the Citrix ADC appliance searches the configured resource records for the domain name. You can configure the ADNS service to be used with the Citrix ADC Global Server Load Balancing (GSLB) feature.

You can delegate a subdomain, by adding NS records for the subdomain to the zone of the parent domain. You can then make the Citrix ADC authoritative for the subdomain, by adding a “glue record” for each of the subdomain name servers. If GSLB is configured, the Citrix ADC makes a GSLB load balancing decision based on its configuration and replies with the IP address of the selected virtual server. The following figure shows the entities in an ADNS GSLB setup and a DNS proxy setup.

Figure 1. DNS Proxy Entity Model



The Citrix ADC appliance can function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by default on the Citrix ADC appliance. Caching enables the Citrix ADC appliance to provide quick responses for repeated translations. Create a load balancing DNS virtual server, and DNS services, and then bind these services to the virtual server.

The Citrix ADC provides two options, minimum time to live (TTL) and maximum TTL for configuring the lifetime of the cached data. The cached data times out as specified by your settings for these two options. The Citrix ADC checks the TTL of the DNS record coming from the server. If the TTL is less than the configured minimum TTL, it is replaced with the configured minimum TTL. If the TTL is

greater than the configured maximum TTL, it is replaced with the configured maximum TTL.

The Citrix ADC also allows caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called *negative caching*. Negative caching helps speed up responses to queries on a domain, and can optionally provide the record type.

A negative response can be one of the following:

- NXDOMAIN error message - If a negative response is present in the local cache, the Citrix ADC returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the Citrix ADC. The Citrix ADC caches the response locally, then returns the error message to the client.
- NODATA error message - The Citrix ADC sends a NODATA error message, if the domain name in query is valid but records of the given type are not available.

The Citrix ADC supports recursive resolution of DNS requests. In recursive resolution, the resolver (DNS client) sends a recursive query to a name server for a domain name. If the queried name server is authoritative for the domain, it responds with the requested domain name. Otherwise, the Citrix ADC queries the name servers recursively until the requested domain name is found.

Before you can apply the recursive query option, you must first enable it. You can also set the number of times the DNS resolver must send a resolution request (DNS retries) if a DNS lookup fails.

You can configure the Citrix ADC as a DNS forwarder. A forwarder passes DNS requests to external name servers. The Citrix ADC allows you to add external name servers and provides name resolution for domains outside the network. The Citrix ADC also allows you to set the name lookup priority to DNS or Windows Internet Name Service (WINS).

Enable the ADC appliance to use DNS to resolve the host name to its respective IP address

Note: You require an SSH utility to access the command line interface (CLI) of the appliance.

By default, the ADC appliance cannot resolve the host name to its respective IP address. Complete the following tasks to enable the name resolution on the appliance:

1. Define name servers.
2. Define a DNS suffix.

Points to note

Perform the DNS lookup from the CLI. DNS lookups from the shell prompt of the FreeBSD operating system fail because the entry in the /etc/resolv.conf file points to the 127.0.0.2 IP address.

The following commands are not available in the CLI of the appliance:


```
- host
- dig
- getent/MIP
- nslookup
<!--NeedCopy-->
```

If the appliance cannot ping the DNS server on its SNIP address, the server status shows as down. Successful ping is important when the appliance is behind a firewall.

CLI configuration

At the command prompt, type:

```
add dns nameServer <Name_Server_IP_Address>
add dns suffix <DNS_Suffix>
<!--NeedCopy-->
```

To verify the configuration, type:

```
show dns nameServer
show dns suffix
<!--NeedCopy-->
```

To test DNS resolution, type:

```
show dns addrec <Host_Name>
<!--NeedCopy-->
```

GUI configuration

1. Navigate to **Traffic Management > DNS > Names Servers > Add**.
2. In the **Create Name Server** dialog box enter the name server IP Address and click **Create**.
3. Navigate to **Traffic Management > DNS > DNS Suffix > Add**.
4. In the **Create DNS Suffix** dialog box, enter the DNS Suffix, such as example.com, to be used for all host queries and click **Create**.

Round Robin DNS

When a client sends a DNS request to find the DNS resource record, it receives a list of IP addresses resolving to the name in the DNS request. The client then uses one of the IP addresses in the list, generally, the first record or IP address. Hence, a single server is used for the total TTL of the cache and is overloaded when many requests arrive.

When the Citrix ADC receives a DNS request, it responds by changing the order of the list of DNS resource records in a round robin method. This feature is called *round robin DNS*. Round robin distributes the traffic equally between data centers. The Citrix ADC performs this function automatically. You do not have to configure this behavior.

Functional Overview

If the Citrix ADC is configured as an ADNS server, it returns the DNS records in the order in which the records are configured. When the Citrix ADC is configured as a DNS proxy, it returns the DNS records in the order in which it receives the records from the server. The order of the records present in the cache matches the order in which records are received from the server.

The Citrix ADC then changes the order in which records are sent in the DNS response in a round robin method. The first response contains the first record in sequence, the second response contains the second record in sequence, and the order continues in the same sequence. Thus, clients requesting the same name can connect to different IP addresses.

Round Robin DNS Example

As an example of round robin DNS, consider DNS records that have been added as follows:

```
1   add dns addRec ns1 1.1.1.1  add dns addRec ns1 1.1.1.2  add dns
       addRec ns1 1.1.1.3  add dns addRec ns1 1.1.1.4
2 <!--NeedCopy-->
```

The domain, abc.com is linked to an NS record as follows:

```
1   add dns nsrec abc.com. ns1
2 <!--NeedCopy-->
```

When the Citrix ADC receives a query for the A record of ns1, the Address records are served in a round robin method as follows. In the first DNS response, 1.1.1.1 is served as the first record:

```
1   ns1.                    1H IN A         1.1.1.1  ns1.
                           1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4
2 <!--NeedCopy-->
```

In the second DNS response, the second IP address, 1.1.1.2 is served as the first record:

```
1   ns1.                    1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4  ns1.
                           1H IN A         1.1.1.1
```



```
<!--NeedCopy-->
```

In the third DNS response, the third IP address, 1.1.1.2 is served as the first record:

```
  ns1.                    1H IN A         1.1.1.3  ns1.
                         1H IN A         1.1.1.4  ns1.
                         1H IN A         1.1.1.1  ns1.
                         1H IN A         1.1.1.2
<!--NeedCopy-->
```

Configure DNS resource records

September 14, 2021

You configure resource records on the Citrix® ADC appliance when you configure the appliance as an ADNS server for a zone. You can also configure resource records on the appliance if the resource records belong to a zone for which the appliance is a DNS proxy server. On the appliance, you can configure the following record types:

- Service records
- AAAA records
- Address records
- Mail Exchange records
- Name Server records
- Canonical records
- Pointer records
- NAPTR records
- Start of Authority records
- Text records

The following table lists the record types that you can configure for a domain name record on the Citrix ADC appliance. For example, you can configure a maximum of 25 IP addresses for one record.

Table 1. Record Type and Number Configurable

Record Type	Number of Records
Address (A)	25
IPv6 (AAAA)	5
Mail exchange (MX)	12
Name server (NS)	16

Record Type	Number of Records
Service (SRV)	8
Pointer (PTR)	20
Canonical name (CNAME)	1
Start of Authority (SOA)	1
Text (TXT)	20
Naming Authority Pointer (NAPTR)	20

> **Note:**
>
> The maximum number of IP addresses for a specific host name is 25. However, the number of different address records can be more than 25.

Create SRV records for a service

September 14, 2021

The SRV record provides information about the services available on the Citrix ADC appliance. An SRV record contains the following information:

- Name of the service and the protocol
- Domain name
- TTL
- DNS class
- Priority of the target
- Weight of records with the same priority
- Port of the service
- Host name of the service.

The Citrix ADC chooses the SRV record that has the lowest priority setting first. If a service has multiple SRV records with the same priority, clients use the weight field to determine which host to use.

Add an SRV record by using the CLI

At the command prompt, type the following commands to add an SRV record and verify the configuration:


```
1 -  add dns srvRec <domain> <target> -priority <positive_integer> -
     weight <positive_integer> -port <positive_integer> [-TTL <secs>]
2 -  sh dns srvRec <domain>
3 <!--NeedCopy-->
```

Example:

```
1 > add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -
     weight 1 -port 80
2  Done
3 > show dns srvRec _http._tcp.example.com
4 1)      Domain Name : _http._tcp.example.com
5         Target Host : nameserver1.com
6         Priority : 1    Weight : 1
7         Port : 80       TTL : 3600 secs
8  Done
9 <!--NeedCopy-->
```

Modify or remove an SRV record by using the CLI

- To modify an SRV record, type:
  - The `set dns srvRec` command
  - The name of the domain for which the SRV record is configured
  - The name of the target host that hosts the associated service
  - The parameters to be changed, with their new values
- To remove an SRV record, type:
  - The `rm dns srvRec` command
  - The name of the domain for which the SRV record is configured
  - The name of the target host that hosts the associated service

Configure an SRV record by using the GUI

Navigate to **Traffic Management > DNS > Records > SRV Records** and create an SRV record.

Create AAAA records for a domain name

September 14, 2021

An AAAA resource record stores a single IPv6 address.

Add an AAAA record by using the CLI

At the command prompt, type the following commands to add an AAAA record and verify the configuration:

```
-  add dns aaaaRec <hostName> <IPv6Address> ... [-TTL <secs>]
-  show dns aaaaRec <hostName>
<!--NeedCopy-->
```

Example:

```
> add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57
   ab
 Done
> show dns aaaaRec www.example.com
1)      Host Name : www.example.com
        Record Type : ADNS              TTL : 5 secs
        IPV6 Address : 2001:db8::1428:57ab
 Done
<!--NeedCopy-->
```

To remove an AAAA record and all the IPv6 addresses associated with the domain name, type the `rm dns aaaaRec` command and the domain name for which the AAAA record is configured. To remove only a subset of the IPv6 addresses associated with the domain name in an AAAA record, type the following:

- `rm dns aaaaRec` command
- The domain name for which the AAAA record is configured
- The IPv6 addresses that you want to remove

Add an AAAA record by using the GUI

Navigate to **Traffic Management > DNS > Records > AAAA Records** and create an AAAA record.

Create address records for a domain name

September 14, 2021

Address (A) records are DNS records that map a domain name to an IPv4 address.

You cannot delete Address records for a host participating in global server load balancing (GSLB). However, the Citrix ADC deletes Address records added for GSLB domains when you unbind the domain

from a GSLB virtual server. Only user-configured records can be deleted manually. You cannot delete a record for a host referenced by records such as NS, MX, or CNAME.

Add an Address record by using the CLI

At the command prompt, type the following commands to add an Address record and verify the configuration:

```
-  add dns addRec <hostName> <IPAddress> [-TTL <secs>]
-  show dns addRec <hostName>
<!--NeedCopy-->
```

Example:

```
> add dns addRec ns.example.com 192.0.2.0
 Done
> show dns addRec ns.example.com
1)      Host Name : ns.example.com
        Record Type : ADNS                      TTL : 5 secs
        IP Address : 192.0.2.0
 Done
<!--NeedCopy-->
```

To remove an Address record and all the IP addresses associated with the domain name, type the `rm dns addRec` command and the domain name for which the Address record is configured. To remove only a subset of the IP addresses associated with the domain name in an Address record, type the following:

- `rm dns addRec` command
- The domain name for which the Address record is configured
- The IP addresses that you want to remove

Add an Address record by using the GUI

Navigate to **Traffic Management > DNS > Records > Address Records** and create an Address record.

Create MX records for a mail exchange server

September 14, 2021

Mail Exchange (MX) records are used to direct email messages across the Internet. An MX record contains an MX preference that specifies the MX server to be used. The MX preference values range from 0

through 65536. An MX record contains a unique MX preference number. You can set the MX preference and the TTL values for an MX record.

When an email message is sent through the Internet, a mail transfer agent sends a DNS query requesting the MX record for the domain name. This query returns a list of host names of mail exchange servers for the domain, along with a preference number. If there are no MX records, the request is made for the Address record of that domain. A single domain can have multiple mail exchange servers.

Add an MX record by using the CLI

At the command prompt, type the following commands to add an MX record and verify the configuration:

```
1 -  add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <
     secs>]
2 -  show dns mxRec <domain>
3 <!--NeedCopy-->
```

Example:

```
1 > add dns mxRec example.com -mx mail.example.com -pref 1
2  Done
3 > show dns mxRec example.com
4 1)      Domain : example.com    MX Name : mail.example.com
5         Preference :   1        TTL : 5 secs
6  Done
7 <!--NeedCopy-->
```

Modify or remove an MX record by using the CLI

- To modify an MX record, type the `set dns mxRec` command, the name of the domain for which the MX record is configured, the name of the MX record, and the parameters to be changed, with their new values.
- To set the TTL parameter to its default value, type the `unset dns mxRec` command, the name of the domain for which the MX record is configured, the name of the MX record, and -TTL without any TTL value. You can use the `unset dns mxRec` command to unset only the TTL parameter.
- To remove an MX record, type the `rm dns mxRec` command, the name of the domain for which the MX record is configured, and the name of the MX record.

Add an MX record by using the GUI

Navigate to **Traffic Management > DNS > Records > Mail Exchange Records** and create an MX record.

Create NS Records for an authoritative server

September 14, 2021

Name Server (NS) records specify the authoritative server for a domain. You can configure a maximum of 16 NS records. You can use an NS record to delegate the control of a subdomain to a DNS server.

Create an NS record by using the CLI

At the command prompt, type the following commands to create an NS record and verify the configuration:

```
-  add dns nsRec <domain> <nameServer> [-TTL <secs>]
-  show dns nsRec <domain>
<!--NeedCopy-->
```

Example:

```
> add dns nsRec example.com nameserver1.example.com
 Done
> show dns nsRec example.com
1)      Domain : example.com    NameServer : nameserver1.example.com
        TTL : 5 sec
 Done
<!--NeedCopy-->
```

To remove an NS record, type the `rm dns nsRec` command, the name of the domain to which the NS record belongs, and the name of the name server.

Create an NS record by using the GUI

Navigate to **Traffic Management > DNS > Records > Name Server Records** and create an NS record.

Create CNAME records for a subdomain

September 14, 2021

A canonical name record (CNAME record) is an alias for a DNS name. These records are useful when multiple services query the DNS server. The host that has an address (A) record cannot have a CNAME record.

Sometimes, a Citrix ADC appliance in proxy mode requests an address record from the cache instead of the server.

Add a CNAME record by using the CLI

At the command prompt, type the following commands to create a CNAME record and verify the configuration:

```
-  add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
-  show dns cnameRec <aliasName>
<!--NeedCopy-->
```

Example:

```
> add dns cnameRec www.example.com www.examplenw.com
 Done
> show dns cnameRec www.example.com
        Alias Name      Canonical Name  TTL
1)      www.example.com        www.examplenw.com       5 secs
 Done
<!--NeedCopy-->
```

To remove a CNAME record for a given domain, type the `rm dns cnameRec` command and the alias of the domain name.

Add a CNAME record by using the GUI

Navigate to **Traffic Management > DNS > Records > Canonical Records** and create a CNAME record.

Cache CNAME records

When deployed in a proxy mode, the ADC appliance does not always send the query for an address record to the back-end server. This behavior occurs when for an answer to a query for an address record, a partial CNAME chain is present in the cache. There are few conditions in which the ADC caches the partial CNAME record and serves the query from the cache. Following are the conditions:

- Citrix ADC must be deployed in a proxy mode.
- The response from the back-end server must have a CNAME chain, for which the record type of the last entry in the answer section must be a CNAME and the question type not a CNAME.

- The response from the back-end server cannot be a No-data or NX-Domain.
- The response from the back-end server has to be an authoritative response.

Create NAPTR records for telecommunications domain

September 14, 2021

NAPTR (Naming Address Pointer) is one of the most commonly used DNS records in the telecommunications domain. NAPTR records map the Internet telephony address space to the Internet address space. They therefore enable a mobile device to send a request to the correct server. The combination of NAPTR records with Service Records (SRV) allows the chaining of multiple records to form complex rewrite rules that produce new domain labels or uniform resource identifiers (URIs). The DNS code for NAPTR is 35.

Citrix ADCs support NAPTR in two modes: ADNS mode and proxy mode. In proxy mode, the ADC caches the response from the servers and uses the cached records to server future queries. A maximum of 20 NAPTR records can be added for a particular domain in Citrix ADC. Citrix ADC caches the reply to a DNS NAPTR record query. Any subsequent requests for the NAPTR record are served from the cache.

Create a NAPTR record by using CLI

At the command prompt, type the following commands to add a NAPTR record and verify the configuration:

```
add dns naptrRec <order> <preference>[flags<string>][services<string>](
regexp<expressions>|-replacement<string>)[-TTL<secs>]
```

Remove a NAPTR record by using CLU

```
rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-services
 <string>] (-regexp <expression> | -replacement <string>))| -recordId <
positive_integer>@)
```

Configure a NAPTR record using GUI

Navigate to **Traffic Management > DNS > Records > NAPTR Records** and create an NAPTR record.

Create PTR records for IPv4 and IPv6 addresses

September 14, 2021

A pointer (PTR) record translates an IP address to its domain name. IPv4 PTR records are represented by the octets of an IP address in reverse order with the string "in-addr.arpa." appended at the end. For example, the PTR record for the IP address 1.2.3.4 is 4.3.2.1.in-addr.arpa.

IPv6 addresses are reverse mapped under the domain IP6.ARPA. IPv6 reverse-maps use a sequence of nibbles separated by dots with the suffix ".IP6.ARPA" as defined in RFC 3596. For example, the reverse lookup domain name corresponding to the address, 4321:0:1:2:3:4:567:89ab would be b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.

Add a PTR record by using the CLI

At the command prompt, type the following commands to add a PTR record and verify the configuration:

```
1 -  add dns ptrRec <reverseDomain> <domain> [-TTL <secs>]
2 -  show dns ptrRec <reverseDomain>
3 <!--NeedCopy-->
```

Example:

```
1 > add dns ptrRec 0.2.0.192.in-addr.arpa example.com
2  Done
3 > show dns ptrRec 0.2.0.192.in-addr.arpa
4 1)      Reverse Domain Name : 0.2.0.192.in-addr.arpa
5         Domain Name : example.com              TTL : 3600 secs
6  Done
7 <!--NeedCopy-->
```

To remove a PTR record, type the `rm dns ptrRec` command and the reverse domain name associated with the PTR record

Add a PTR record by using the GUI

Navigate to **Traffic Management > DNS > Records > PTR Records** and create a PTR record.

Create SOA records for authoritative information

September 14, 2021

A Start of Authority (SOA) record is created only at the zone apex and contains information about the zone. The record includes, among other parameters, the primary name server, contact information (email), and default (minimum) time-to-live (TTL) values for records.

Create an SOA record by using the CLI

At the command prompt, type the following commands to add an SOA record and verify the configuration:

```
1 - add dns soaRec <domain> -originServer <originServerName> -contact <
    contactName>
2 - sh dns soaRec <do main>
3 <!--NeedCopy-->
```

Example:

```
1 > add dns soaRec example.com -originServer nameserver1.example.com -
    contact admin.example.com
2  Done
3 > show dns soaRec example.com
4 1)      Domain Name : example.com
5         Origin Server : nameserver1.example.com
6         Contact : admin.example.com
7         Serial No. : 100        Refresh : 3600 secs     Retry : 3 secs
8         Expire : 3600 secs      Minimum : 5 secs        TTL : 3600 secs
9  Done
10 <!--NeedCopy-->
```

Modify or remove an SOA record by using the CLI

- To modify an SOA record, type the `set dns soaRec` command, the name of the domain for which the record is configured, and the parameters to be changed, with their new values.
- To remove an SOA record, type the `rm dns soaRec` command and the name of the domain for which the record is configured.

Configure an SOA record by using the GUI

Navigate to **Traffic Management > DNS > Records > SOA Records** and create an SOA record.

Create TXT records for holding descriptive text

September 14, 2021

Domain hosts store TXT records for informative purposes. A TXT record's RDATA component, which consists of one or more character strings of variable length, can store practically any information that a recipient might need to know about the domain. It can also include information about the service provider, contact person, email addresses, and associated details. SPF (Sender Policy Framework) protection has been the most prominent use case for the TXT record.

All configuration types (authoritative DNS, DNS proxy, end resolver, and forwarder configurations) on the Citrix ADC appliance support TXT records. You can add a maximum of 20 TXT resource records to a domain. Each resource record is stored with a unique, internally generated record ID. You can view the ID of a record and use it to delete the record. However, you cannot modify a TXT resource record.

Create a TXT resource record by using the CLI

At the command prompt, type the following commands to create a TXT resource record and verify the configuration:

```
1 -  add dns txtRec <domain> <string> ... [-TTL <secs>]
2 -  show dns txtRec [<domain> | -type <type>]
3 <!--NeedCopy-->
```

Example:

```
1 > add dns txtRec www.example.com "Contact: Mark" "Email: mark@example.
     com" -TTL 36000
2  Done
3 > show dns txtRec www.example.com
4 1)  Domain : www.example.com    Record id: 13783       TTL : 36000 secs
      Record Type : ADNS
5         "Contact: Mark"
6         "Email: mark@example.com"
7  Done
8 <!--NeedCopy-->
```

Remove a TXT resource record by using the CLI

At the command prompt, type the following commands to remove a TXT resource record and verify the configuration:


```
-  rm dns txtRec <domain> (<string> ... | -recordId <positive_integer>)
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

Example:

You can use the `show dns txtRec` command first to view the record ID of the TXT resource record that you want to remove, as shown:

```
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 36865       TTL : 36000 secs
     Record Type : ADNS
         "Contact: Evan"
         "Email: evan@example.com"
2)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
     Record Type : ADNS
         "Contact: Mark"
         "Email: mark1@example.com"
 Done
<!--NeedCopy-->
```

The simpler method of deleting a TXT record is to use the record ID. If you want to provide the strings, enter them in the order in which they are stored in the record. In the following example, the TXT record is deleted by using its record ID.

```
>rm dns txtRec www.example.com -recordID 36865
 Done
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
     Record Type : ADNS
         "Contact: Mark"
         "Email: mark1@example.com"
 Done
<!--NeedCopy-->
```

Configure a TXT record by using the GUI

Navigate to **Traffic Management > DNS > Records > TXT Records** and create a TXT record.

View DNS statistics

September 14, 2021

You can view the DNS statistics generated by the Citrix ADC appliance. The DNS statistics include run-time, configuration, and error statistics.

View DNS records statistics by using the CLI

At the command prompt, type:

```
stat dns
```

Example:

```
> stat dns
DNS Statistics

Runtime Statistics
Dns queries                             21
NS queries                               8
SOA queries                             18
.
.
.
Configuration Statistics
AAAA records                            17
A records                               36
MX records                               9
.
.
.
Error Statistics
Nonexistent domain                      17
No AAAA records                          0
No A records                            13
.
.
.
Done
<!--NeedCopy-->
```

View DNS records statistics by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, click **Statistics**.

Configure a DNS zone

September 14, 2021

A DNS zone entity on the Citrix ADC appliance facilitates the ownership of a domain on the appliance. A zone on the appliance also enables you to implement DNS Security Extensions (DNSSEC) for the zone, or to offload the zone's DNSSEC operations from the DNS servers to the appliance. DNSSEC sign operations are performed on all the resource records in a DNS zone. Therefore, if you want to sign a zone, or if you want to offload DNSSEC operations for a zone, you must first create the zone on the Citrix ADC appliance.

Create a DNS zone on the appliance in the following scenarios:

- The Citrix ADC appliance owns all the records in a zone, that is, the appliance is operating as the authoritative DNS server for the zone. The zone must be created with the proxyMode parameter set to NO.
- The Citrix ADC appliance owns only a subset of the records in a zone. All the other resource records in the zone are hosted on a set of back-end name servers. The appliance is configured as a DNS proxy server for these back-end servers. A typical configuration where the Citrix ADC appliance owns only a subset of the resource records in the zone is a global server load balancing (GSLB) configuration. The Citrix ADC appliance owns only the GSLB domain names, while the back-end name servers own all the other records. The zone must be created with the proxyMode parameter set to YES.
- You want to offload DNSSEC operations for a zone from your authoritative DNS servers to the appliance. The zone must be created with the proxyMode parameter set to YES. You might have to configure more settings for the zone.

The current topic describes how to create a zone for the first two scenarios. For more information about how to configure a zone for offloading DNSSEC operations to the appliance, see Offload DNSSEC operations to the Citrix ADC appliance.

> **Note**
>
> If the ADC appliance is operating as the authoritative DNS server for a zone, you must create the Start of Authority (SOA) and name server (NS) records for the zone before you create the zone. If the Citrix ADC is operating as the DNS proxy server for a zone, SOA and NS records must not be created on the Citrix ADC appliance. For more information about creating SOA and NS records, see Configure DNS resource records.

When you create a zone, all existing domain names and resource records that end with the name of the zone are automatically treated as a part of the zone. Also, any new resource records created with a suffix that matches the name of the zone are implicitly included in the zone.

Create a DNS zone on the Citrix ADC appliance by using the CLI

At the command prompt, type the following command to add a DNS zone to the Citrix ADC appliance and verify the configuration:

```
-  add dns zone <zoneName> -proxyMode ( YES | NO )
-  show dns zone [<zoneName> | -type <type>]
<!--NeedCopy-->
```

Example:

```
> add dns zone example.com -proxyMode Yes
 Done
> show dns zone example.com
          Zone Name : example.com
          Proxy Mode : YES
Done
<!--NeedCopy-->
```

Modify or remove a DNS zone by using the CLI

- To modify a DNS zone, type the `set dns zone` command, the name of the DNS zone, and the parameters to be changed, with their new values.
- To remove a DNS zone, type the `rm dns zone` command and the name of the DNS zone.

Configure a DNS zone by using the GUI

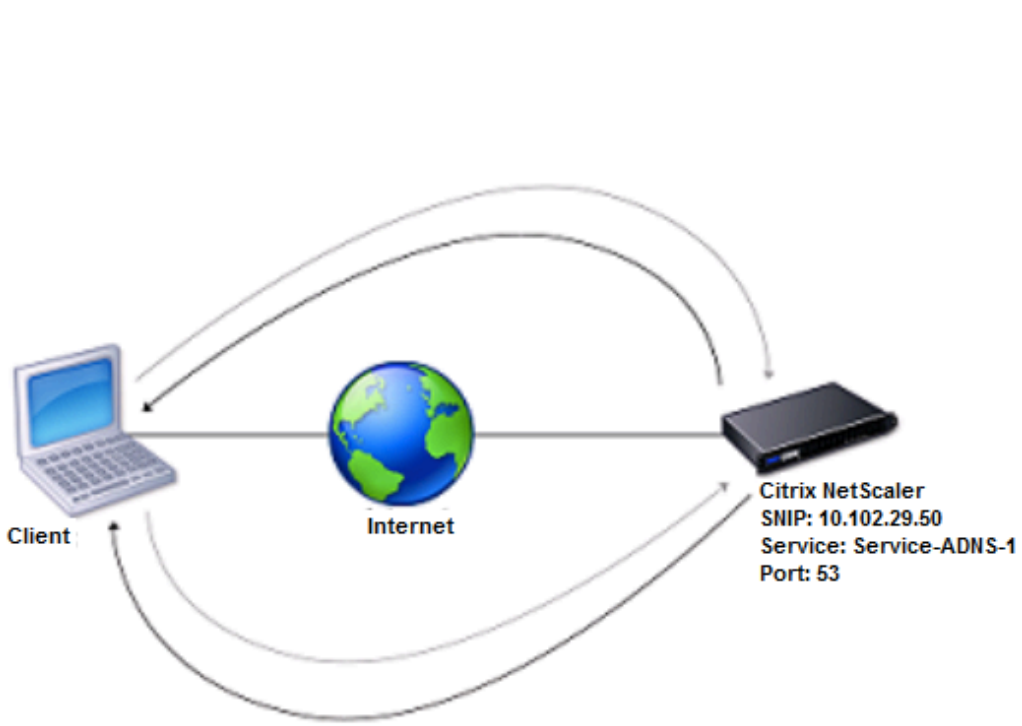
Navigate to **Traffic Management > DNS > Zones** and create a DNS zone.

Configure the Citrix ADC as an ADNS server

September 14, 2021

You can configure the ADC appliance to function as an authoritative domain name server (ADNS) for a domain. As an ADNS server for a domain, the Citrix ADC resolves DNS requests for all types of DNS records that belong to the domain. To configure the Citrix ADC to function as an ADNS server for a domain, you must create an ADNS service and configure NS and Address records for the domain on the Citrix ADC. The ADNS service can be configured using the subnet IP address (SNIP) or a separate IP address. The following topology diagram shows a sample configuration and the flow of requests and responses.

Figure 1. Citrix ADC as an ADNS



The following table shows the parameters that are configured for the ADNS service illustrated in the preceding topology diagram.

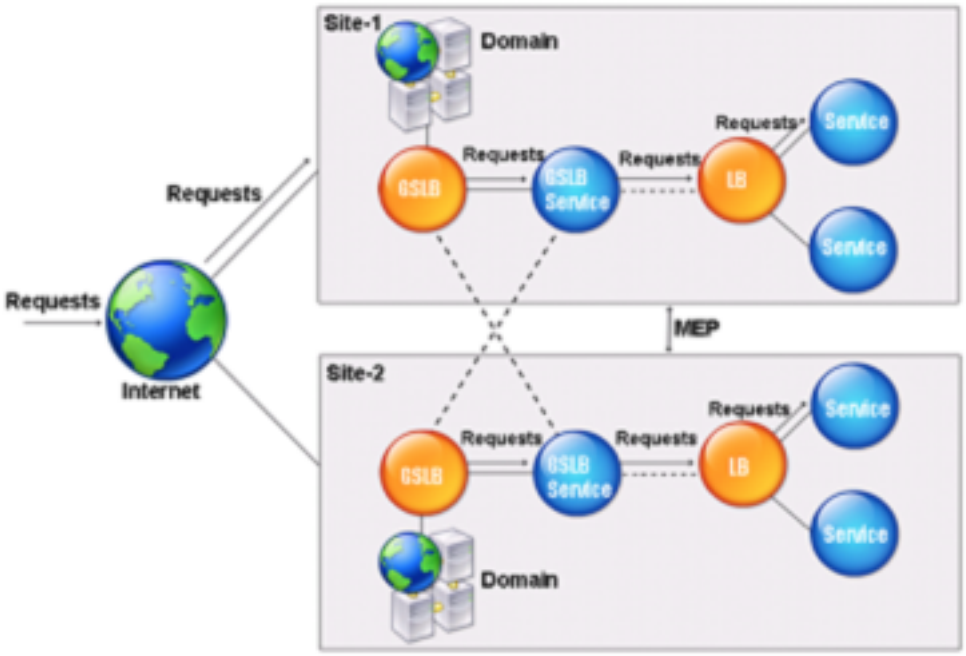
Entity type	Name	IP address	Type	Port
ADNS Service	Service-ADNS-1	10.102.29.51	ADNS	53

Table 1. Example of ADNS Service Configuration

To configure an ADNS setup, you must configure the ADNS service. For instructions on configuring the ADNS service, see Load balancing.

During DNS resolution, the ADNS server directs the DNS proxy or local DNS server to query the Citrix ADC for the IP address of the domain. Because the Citrix ADC is authoritative for the domain, it sends the IP address to the DNS proxy or local DNS server. The following diagram describes the placement and role of the ADNS server in a GSLB configuration.

Figure 2. GSLB Entity Model



Note: In ADNS mode, if you remove SOA and ADNS records, the following do not function for the domain hosted by the Citrix ADC: ANY query (for more information about the ANY query, see DNS ANY query), and negative responses, such as NODATA and NXDOMAIN.

Create an ADNS service

An ADNS service is used for global service load balancing. For more information about creating a GSLB setup, see Global server load balancing. You can add, modify, enable, disable, and remove an ADNS service. For instructions on creating an ADNS service, see Configure services.

Note: You can configure the ADNS service to use SNIP or any new IP address.

When you create an ADNS service, the Citrix ADC responds to DNS queries on the configured ADNS service IP and port.

You can verify the configuration by viewing the properties of the ADNS service. You can view properties such as name, state, IP address, port, protocol, and maximum client connections.

Configure the ADNS setup to use TCP

By default, some clients use the User Datagram Protocol (UDP) for DNS, which specifies a limit of 512 bytes for the payload length of UDP packets. To handle payloads that exceed 512 bytes in size, the

client must use TCP. To enable DNS communications over TCP, you must configure the Citrix ADC appliance to use the TCP protocol for DNS. The Citrix ADC then sets the truncation bit in the DNS response packets. The truncation bit specifies that the response is too large for UDP and that the client must send the request over a TCP connection. The client then uses the TCP protocol on port 53 and opens a new connection to the Citrix ADC. The Citrix ADC listens on port 53 with the IP address of the ADNS service to accept the new TCP connections from the client.

To configure the Citrix ADC to use the TCP protocol, you must configure an ADNS_TCP service. For instructions on creating an ADNS_TCP service, see Load balancing.

> **Important**
>
> To configure the Citrix ADC to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure the ADNS and ADNS_TCP services. The IP address of the ADNS_TCP service must be the same as the IP address of the ADNS service.

Add DNS resource records

After you create an ADNS service, you can add DNS records. For instructions on adding DNS records, see Configure DNS resource records.

Remove ADNS services

For instructions on removing services, see Load balancing

Configure domain delegation

Domain delegation is the process of assigning responsibility for a part of the domain space to another name server. Therefore, during domain delegation, the responsibility for responding to the query is delegated to another DNS server. Delegation uses NS records.

In the following example, sub1.abc.com is the subdomain for abc.com. The procedure describes the steps to delegate the subdomain to the name server ns2.sub1.abc.com and add an Address record for ns2.sub1.abc.com.

To configure domain delegation, you need to perform the following tasks, which are described in the sections that follow:

1. Create an SOA record for a domain.
2. Create an NS record to add a name server for the domain.
3. Create an Address record for the name server.
4. Create an NS record to delegate the subdomain.
5. Create a glue record for the name server.

Create an SOA record

For instructions on configuring SOA records, see Create SOA records for authoritative information.

Create an NS record for a name server

For instructions on configuring an NS record, see Create NS records for an authoritative server. In the **Name Server** list, select the primary authoritative name server, for example, ns1.abc.com.

Create an address record

For instructions on configuring Address records, see Create address records for a domain name. In the Host Name and IP address text boxes, type the domain name for the DNS Address record and the IP address, for example, ns1.abc.com and 10.102.11.135, respectively.

Create an NS record for domain delegation

For instructions on configuring NS records, see Create NS records for an authoritative server. In the **Name Server** list, select the primary authoritative name server, for example, ns2.sub1.abc.com.

Create a glue record

NS records are typically defined immediately after the SOA record (not a restriction.) A domain must have at least two NS records. If an NS record is defined within a domain, it must have a matching Address record. This Address record is referred to as a glue record. Glue records speed up DNS queries.

For instructions on adding glue records for a subdomain, see the procedure for adding an Address (A) record, Configure DNS resource records.

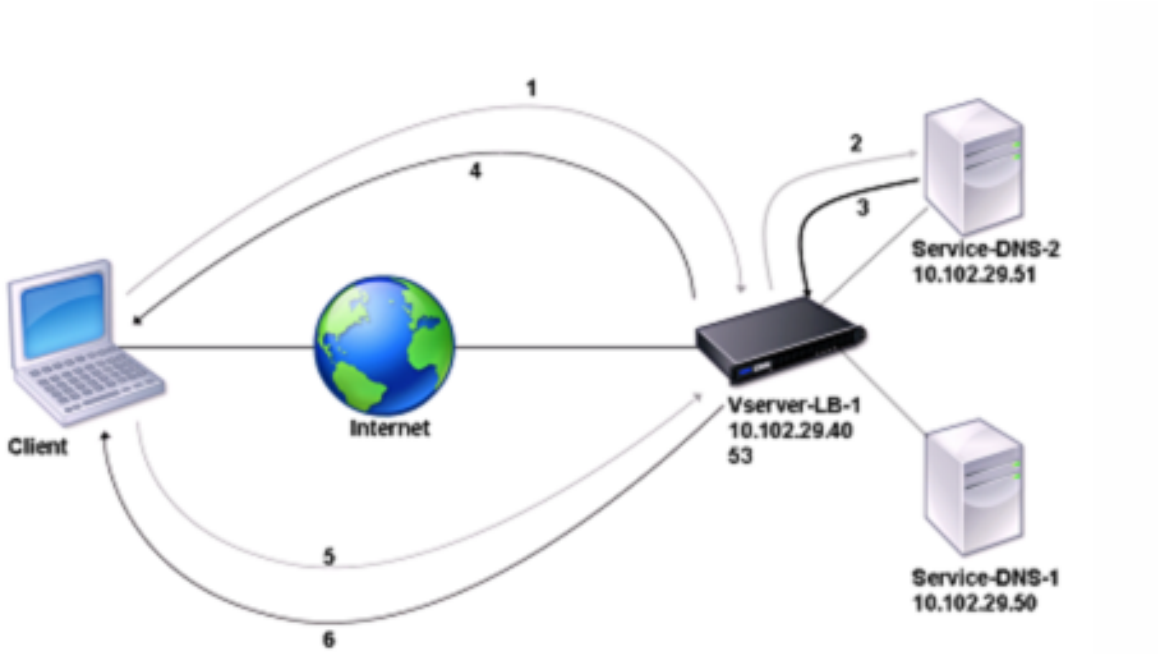
For instructions on configuring Address records, see Create address records for a domain name. In the Host Name and IP address text boxes, type the domain name for the DNS Address record and the IP address, for example, ns2.sub1.abc.com and 10.102.12.135, respectively.

Configure the Citrix ADC appliance as a DNS proxy server

September 14, 2021

As a DNS proxy server, the ADC appliance can function as a proxy for either a single DNS server or a group of DNS servers. The flow of requests and responses is illustrated in the following sample topology diagram.

Figure 1. Citrix ADC as DNS proxy



By default, the Citrix ADC appliance caches responses from DNS name servers. When the appliance receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the Citrix ADC returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the Citrix ADC. The Citrix ADC then returns the address to the client.

For requests for a domain that has been cached earlier, the Citrix ADC serves the Address record of the domain from the cache without querying the configured DNS server.

The appliance discards a record stored in its cache when the time-to-live (TTL) value of the record reaches the configured value. A client that requests an expired record has to wait until the Citrix ADC retrieves the record from the server and updates its cache. To avoid this delay, the Citrix ADC proactively updates the cache by retrieving the record from the server before the record expires.

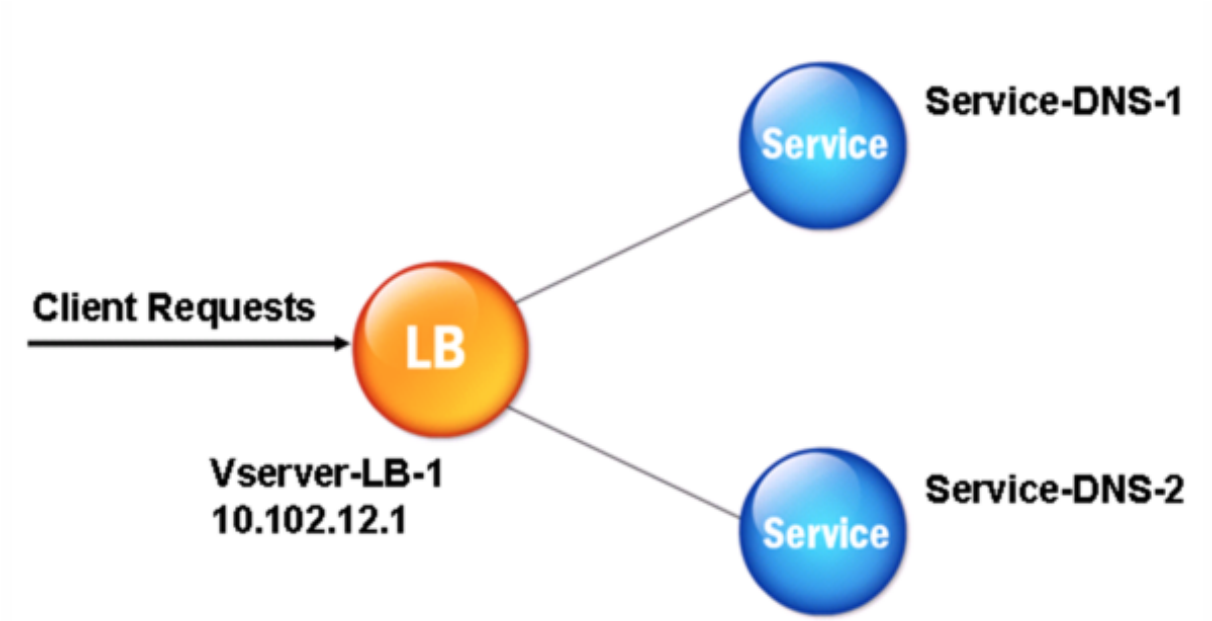
The following table lists sample names and the values of the entities that need to be configured on the Citrix ADC.

Table 1. Example of DNS Proxy Entity Configuration

Entity type	Name	IP address	Type	Port
LB virtual server	Vserver-DNS-1	10.102.29.40	DNS	53
Services	Service-DNS-1	10.102.29.50	DNS	53
Services	Service-DNS-2	10.102.29.51	DNS	53

The following diagram shows the entities of a DNS Proxy and the values of the parameters to be configured on the Citrix ADC.

Figure 2. DNS Proxy Entity Model



> **Note**
>
> To configure the DNS proxy feature, you need to know how to configure load balancing services and virtual servers.

Create a load balancing virtual server

To configure a DNS Proxy on the Citrix ADC, configure a load balancing virtual server of type DNS. To configure a DNS virtual server to load balance a set of DNS servers that support recursive queries, you must set the Recursion Available option. With this option, the RA bit is set to ON in the DNS replies from the DNS virtual server.

For instructions on creating a load balancing virtual server, see Load Balancing.

Create DNS services

After creating a load balancing virtual server of type DNS, you must create DNS services. You can add, modify, enable, disable, and remove a DNS service. For instructions on creating a DNS service, see Load Balancing.

Bind a load balancing virtual server to DNS services

To complete the DNS Proxy configuration, you must bind the DNS services to the load balancing virtual server. For instructions on binding a service to a load balancing virtual server, see Load Balancing.

Configure the DNS proxy setup to use TCP

Some clients use the User Datagram Protocol (UDP) for DNS communications. However, UDP specifies a maximum packet size of 512 bytes. When payload lengths exceed 512 bytes, the client must use TCP. When a client sends the Citrix ADC appliance a DNS query, the appliance forwards the query to one of the name servers. If the response is too large for a UDP packet, the name server sets the truncation bit in its response to the Citrix ADC. The truncation bit indicates that the response is too large for UDP and that the client must send the query over a TCP connection. The ADC appliance relays the response to the client with the truncation bit intact. It waits for the client to initiate a TCP connection with the IP address of the DNS load balancing virtual server, on port 53. The client sends the request over a TCP connection. The Citrix ADC appliance then forwards the request to the name server and relays the response to the client.

To configure the Citrix ADC to use the TCP protocol for DNS, you must configure a load balancing virtual server and services, both of type DNS_TCP. You can configure monitors of type DNS_TCP to check the state of the services. For instructions on creating DNS_TCP virtual servers, services, and monitors, see Load Balancing.

For updating the records proactively, the Citrix ADC uses a TCP connection to the server to retrieve the records.

> **Important**
>
> To configure the Citrix ADC to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure both DNS and DNS_TCP services. The IP address of the DNS_TCP service must be the same as the IP address of the DNS service.

Configure time-to-live values for DNS entries

The TTL is the same for all DNS records with the same domain name and record type. If the TTL value is changed for one of the records, the new value is reflected in all records of the same domain name and type. The default TTL value is 3600 seconds. The minimum is 0, and the maximum is 604800. If a DNS entry has a TTL value less than the minimum or greater than the maximum, it is saved as the minimum or maximum TTL value, respectively.

Specify the minimum and maximum TTL by using the CLI

At the Citrix ADC command prompt, type the following commands to specify the minimum and maximum TTL and verify the configuration:

```
-  set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -minTTL 1200 -maxTTL 1800
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        Minimum TTL: 1200               Maximum TTL: 1800
            .
            .
            .
 Done
>
<!--NeedCopy-->
```

Specify the minimum and maximum TTL by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, in TTL, in the Minimum and Maximum text boxes, type the minimum and maximum time to live (in seconds), respectively, and then click OK.

Note: When the TTL expires, the record is deleted from the cache. The Citrix ADC proactively contacts the servers and obtains the DNS record just before the DNS record expires.

Flush DNS records

You can delete all DNS records present in the cache. For example, you might want to flush DNS records when a server is restarted after modifications are made.

Delete all proxy records by using the CLI

At the Citrix ADC command prompt, type:

```
flush dns proxyRecords
```


Delete all proxy records by using the GUI

1. Navigate to **Traffic Management > DNS > Records**.
2. In the details pane, click Flush Proxy Records.

Add DNS resource records

You can add DNS records to a domain for which the Citrix ADC appliance is configured as a DNS proxy server. For information about adding DNS records, see Configuring DNS Resource Records.

Remove a load balancing DNS virtual server

For information about removing a load balancing virtual server, see Load Balancing.

Limit the number of concurrent DNS requests on a client connection

You can limit the number of concurrent DNS requests on a single client connection, which is identified by the <clientip:port>-<vserver ip:port> tuple. Concurrent DNS requests are those requests that the Citrix ADC appliance has forwarded to the name servers and for which the appliance is awaiting responses. Limiting the number of concurrent requests on a client connection enables you to protect the name servers when a hostile client attempts a Distributed Denial of Service (DDoS) attack by sending a flood of DNS requests. When the limit for a client connection is reached, subsequent DNS requests on the connection are dropped until the outstanding request count goes below the limit. This limit does not apply to the requests that the Citrix ADC appliance serves out of its cache.

The default value for this parameter is 255. This default value is sufficient in most scenarios. If the name servers serve many concurrent DNS requests under normal operating conditions, you can specify either a large value or a value of zero (0). A value of 0 disables this feature and specifies that there is no limit to the number of DNS requests that are allowed on a single client connection. This parameter is a global parameter and applies to all the DNS virtual servers that are configured on the Citrix ADC appliance.
The default value for this parameter is 255. This default value is sufficient in most scenarios. If the name servers serve many concurrent DNS requests under normal operating conditions, you can specify either a large value or a value of zero (0). A value of 0 disables this feature and specifies that there is no limit to the number of DNS requests that are allowed on a single client connection. This parameter is a global parameter and applies to all the DNS virtual servers that are configured on the Citrix ADC appliance.
The default value for this parameter is 255. This default value is sufficient in most scenarios. If the name servers serve many concurrent DNS requests under normal operating conditions, you can specify either a large value or a value of zero (0). A value of 0 disables this feature and specifies that there is no limit to the number of DNS requests that are allowed on a single client connection. This parameter

is a global parameter and applies to all the DNS virtual servers that are configured on the Citrix ADC appliance.

Specify the maximum number of concurrent DNS requests allowed on a single client connection by using the CLI

At the command prompt, type the following commands to specify the maximum number of concurrent DNS requests allowed on a single client connection and verify the configuration:

```
-  set dns parameter -maxPipeline <positive_integer>
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -maxPipeline 1000
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        .
        .
        .
        Max DNS Pipeline Requests: 1000
 Done
<!--NeedCopy-->
```

Specify the maximum number of concurrent DNS requests allowed on a single client connection by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, specify a value for Max DNS Pipeline Requests.
4. Click OK.

Configure the Citrix ADC as an end resolver

September 14, 2021

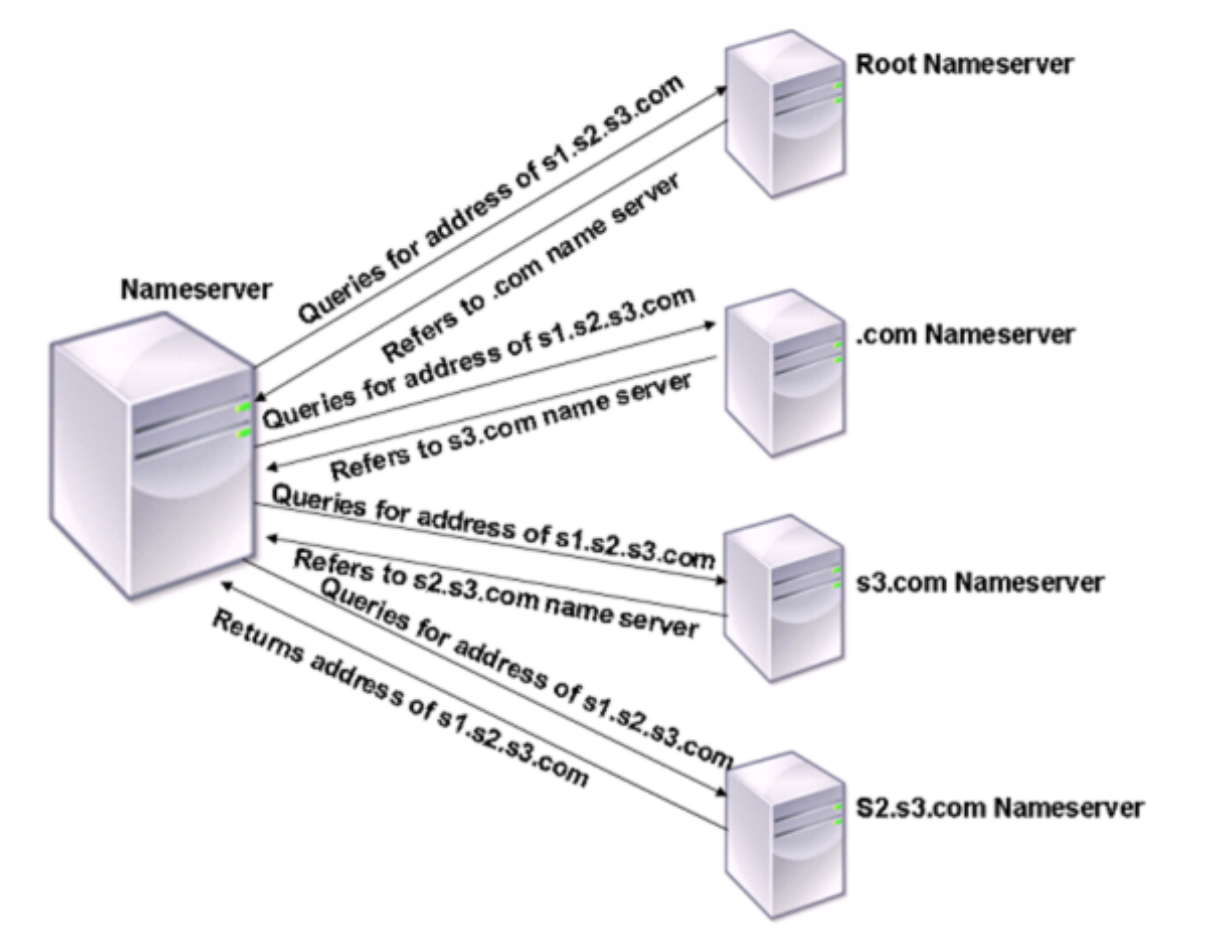
A resolver is a procedure that is invoked by an application program that translates a domain/host name to its resource record. The resolver interacts with the LDNS, which looks up the domain name

to obtain its IP address. The Citrix ADC can provide end-to-end resolution for DNS queries.

In recursive resolution, the Citrix ADC appliance queries different name servers recursively to access the IP address of a domain. When the Citrix ADC receives a DNS request, it checks its cache for the DNS record. If the record is not present in the cache, it queries the root servers configured in the ns.conf file. The root name server reports back with the address of a DNS server that has detailed information about the second-level domain. The process is repeated until the required record is found.

When you start the Citrix ADC appliance for the first time, 13 root name servers are added to the ns.conf file. The NS and Address records for the 13 root servers are also added. You can modify the ns.conf file, but the Citrix ADC does not allow you to delete all 13 records. At least one name server entry is required for the appliance to perform name resolution. The following diagram illustrates the process of name resolution.

Figure 1. Recursive resolution



In the process shown in the diagram, when the name server receives a query for the address of s1.s2.s3.com, it first checks the root name servers for s1.s2.s3.com. A root name server reports back with the address of the .com name server. If the address of s1.s2.s3.com is found in the name server, it responds with a suitable IP address. Otherwise, it queries other name servers for s3.com, then

for s2.s3.com to retrieve the address of s1.s2.s3.com. In this way, resolution always starts from root name servers and ends with the domain's authoritative name server.

Note: For recursive resolution functionality, caching must be enabled.

Enable recursive resolution

To configure the Citrix ADC appliance to function as an end resolver, you must enable recursive resolution on the appliance.

Enable recursive resolution by using the CLI

At the command prompt, type the following commands to enable recursive resolution and verify the configuration:

```
-  set dns parameter -recursion ENABLED
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
 Done
<!--NeedCopy-->
```

Enable recursive resolution by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, select the Enable recursion check box, and then click OK.

Set the Number of Retries

Configure the ADC appliance to make a preconfigured number of attempts (called DNS retries) when it does not receive a response from the server to which it sends a query. By default, the number of DNS retries is set to 5.

Set the number of DNS retries by using the CLI

At the command prompt, type the following commands to set the number of retries and verify the configuration:

```
-  set dns parameter -retries <positive_integer>
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set DNS parameter -retries 3
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 3
            .
            .
            .
 Done
<!--NeedCopy-->
```

Set the number of retries by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, in the DNS Retries text box, type the DNS resolver request retry count, and then click OK.

Configure the Citrix ADC appliance as a forwarder

September 14, 2021

A forwarder is a server that forwards DNS queries to DNS servers that are outside the forwarder server's network. Queries that cannot be resolved locally are forwarded to other DNS servers. A forwarder ac-

cumulates external DNS information in its cache as it resolves DNS queries. To configure the Citrix ADC appliance as a forwarder, you must add an external name server.

The Citrix ADC appliance allows you to add external name servers to which it can forward the name resolution queries that cannot be resolved locally. To configure the Citrix ADC appliance as a forwarder, you must add the name servers to which it must forward name resolution queries. You can specify the lookup priority to specify the name service that the Citrix ADC appliance must use for name resolution.

> **Note:**
>
> The Citrix ADC appliance in forwarder mode supports TCP, UDP, and UDP-TCP name servers.
>
> - If you have configured a TCP name server, then the Citrix ADC appliance sends the DNS request over TCP.
> - If you have configured a UDP name server, then the Citrix ADC appliance sends the DNS request over UDP.
> - If you have configured a UDP-TCP name server, then the Citrix ADC appliance sends the DNS request over UDP. However if the truncated bit is set in the DNS response, the appliance sends such DNS requests over TCP.

Add a name server

September 14, 2021

You can create a name server by specifying its IP address or by configuring an existing virtual server as the name server.

- **IP address-based name server** - An external name server to contact for domain name resolution. If multiple IP address-based name servers are configured on the appliance, and the local parameter is not set on any of them, incoming DNS queries are load balanced across all the name servers, in round robin fashion.
- **Virtual server-based name server** - A DNS virtual server configured in the Citrix ADC. For more fine-grained control on how external DNS name servers are load balanced (for example, you want a load balancing method other than round robin), do the following:
- Configure a DNS virtual server on the appliance
- Bind the external name servers as its services
- Specify the name of the virtual server in this command.

To verify the configuration, you can use the `show dns nameServer` command.

To remove a name server, at the Citrix ADC CLI, type the `rm dns nameServer` command followed by the IP address of the name server.

To view the details of the DNS nameserver, at the Citrix ADC CLI, type the `show dns nameServer` command followed by the IP address of the name server.

Add a name server (when the Citrix ADC appliance acts as a forwarder) by using the CLI

At the command prompt, type;

```
add dns nameServer ((<IP>) | <dnsVserverName>)
<!--NeedCopy-->
```

Or

```
add dns nameServer ((<IP> | <dnsVserverName>) [-type <type>]
<!--NeedCopy-->
```

Examples:

```
add dns nameServer dnsVirtualNS

add dns nameServer 192.0.2.11 -type TCP

add dns nameServer 192.0.2.12 -type UDP_TCP


add dns nameServer 192.0.2.10
show dns nameServer 192.0.2.10

1) 192.0.2.10  -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

> **Note:**
>
> If the name server type is not specified, a UDP name server is created by default. To create a name server of type TCP or UDP_TCP, you must specify the type.
>
> When you specify the type as UDP_TCP, two name servers (one UDP name server and one TCP name server) are created for the given IP address.

Add a name server (when the Citrix ADC appliance acts as a resolver) by using the CLI

At the command prompt, type:

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
<!--NeedCopy-->
```


Example:

```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

Local - Mark the IP address as one that belongs to a local recursive DNS server on the Citrix ADC appliance. The appliance recursively resolves queries received on an IP address that is marked as being local.
For recursive resolution to work, the global DNS parameter, `recursion`, must also be set.
If no name server is marked as being local, the appliance functions as a stub resolver and load balances the name servers.

Add a name server by using the GUI

Navigate to **Traffic Management** > **DNS** > **Name Servers** and create a name server.

Set DNS lookup priority

September 14, 2021

You can set the lookup priority to either DNS or WINS. This option is used in the SSL VPN mode of operation.

Set the lookup priority to DNS by using the CLI

At the command prompt, type the following commands to set the lookup priority to DNS and verify the configuration:

```
-  set dns parameter -nameLookupPriority (DNS | WINS)
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -nameLookupPriority DNS
 Done
> show dns parameter
            .
            .
```



```
6               .
7          Name lookup priority : DNS
8               .
9               .
10              .
11   Done
12 <!--NeedCopy-->
```

Set lookup priority to DNS by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, under **Settings**, click Change DNS settings.
3. In the **Configure DNS Parameters** dialog box, under **Name Lookup Priority**, select DNS or WINS, and then click OK.

> **Note**
>
> If the DNS virtual server that you have configured is DOWN and if you set the `-nameLookupPriority` to DNS, the Citrix ADC does not attempt WINS lookup. Therefore, if a DNS virtual server is not configured or is disabled, set the `-nameLookupPriority` to WINS.

Disable and enable name servers

September 14, 2021

The following procedure describes the steps to enable or disable an existing name server.

Enable or disable a name server by using the CLI

At the command prompt, type the following commands to enable or disable a name server and verify the configuration:

```
1 -  (enable | disable) dns nameServer <IPAddress>
2 -  show dns nameServer <IPAddress>
3 <!--NeedCopy-->
```

Example:

```
1 > disable dns nameServer 10.102.9.19
2  Done
3 > show dns nameServer 10.102.9.19
4 1)       10.102.9.19: LOCAL  -  State: OUT OF SERVICE
```



```
5   Done
6 <!--NeedCopy-->
```

Enable or disable a name server by using the GUI

1. Navigate to **Traffic Management > DNS > Name Servers**.
2. In the details pane, select the name server that you want to enable or disable.
3. Click Enable or Disable. If a name server is enabled, the Disable option is available. If a name server is disabled, the Enable option is available.

Configure Citrix ADC as a non-validating security aware stub-resolver

September 14, 2021

Starting with Citrix ADC 12.1 build 49.xx, Citrix ADC acts as a non-validating security aware stub-resolver. To enable this support, the AD bit is set in the DNS header and the DO bit is unset in the OPT header. When the AD bit is set and the DO bit is unset, the upstream recursive resolver validates the DNSSEC response. If the validation is successful, the recursive resolver responds without DNSSEC RRs. If the DNSSEC validation fails, then the recursive resolver returns with a SERVFAIL response.

> **Important:**
>
> The AD bit is set by default in the ADC forwarder. The AD bit is not set for DBS initiated queries.

Jumbo frames support for DNS to handle responses of large sizes

September 14, 2021

Starting with Citrix ADC 12.1 build 49.xx, DNS supports jumbo frames for handling UDP responses greater than 1,280 bytes. Previously, the Citrix ADC appliance only supported UDP packet size only up to 1,280 bytes.

You can set the maximum UDP packet size that the appliance can handle in proxy, ADNS, and forwarder modes by configuring the Maximum UDP Packet Size parameter value. For example, if the Maximum UDP Packet Size parameter value is set to 4096, the appliance can handle DNS response of size 4,096 bytes.

> **Important**
>
> - In proxy mode, the least size between the client request OPT payload size and the Maximum

> UDP Packet Size value is considered for sending DNS queries to the back end. For example, if the client request OPT payload size is 3000, and the Maximum UDP Packet Size value is 4096, 3,000 bytes DNS queries are sent to the back end.
>
> Also, from the back end, the appliance can receive responses of large sizes and process responses of large sizes.
>
> - In forwarder mode, the appliance sets the OPT payload size equal to the UDP packet size parameter value.
> - If the DNS records are local to the appliance, then the appliance can compose response sizes as large as the Maximum UDP Packet Size parameter value. This setting is applicable for ADNS, proxy, and recursive resolvers.

To configure the maximum UDP packet size using the CLI

At the command prompt, type:

```
set dns parameter [-maxUDPPacketSize <positive_integer>]
<!--NeedCopy-->
```

Example:

```
set dns parameter -maxUDPPacketSize 10000
<!--NeedCopy-->
```

> **Note**: The minimum and maximum value that you can set for the Maximum UDP Packet Size parameter is 512 and 16384 respectively. Default value is 1280.

To configure the maximum UDP packet size using the CLI

1. Navigate to **Traffic Management** > **DNS**.
2. In the details pane, click **Change DNS settings**.
3. In Maximum UDP Packet Size, specify the maximum UDP packet size.
4. Click **OK**.

Configure DNS logging

September 14, 2021

You can configure the Citrix ADC appliance to log the DNS requests and responses that it handles. The appliance logs the DNS requests and responses in SYSLOG format. You can choose to log either DNS

requests or DNS responses, or both, and send the syslog messages to a remote log server. The log messages can be used to:

- Audit the DNS responses to the client
- Audit DNS clients
- Detect and prevent DNS attacks
- Troubleshoot

A Citrix ADC appliance can log the following sections in the DNS request or response, based on your configuration:

- Header Section
- Questions Section
- Answer Section
- Authority Section
- **Additional** Section

DNS profiles

You can use a DNS profile to configure the various DNS parameters that you want the DNS endpoint to apply to the DNS traffic. In the profile, you can enable logging, caching, and negative caching.

Important: From the NetScaler 11.0 release, enabling DNS caching using global DNS parameters have been deprecated. You can enable or disable DNS caching using DNS profiles. You can now enable DNS caching for an individual virtual server by enabling DNS caching in a DNS profile and setting the DNS profile to the individual virtual server.

DNS profiles support the following types of DNS logging:

- DNS Query Logging
- DNS Answer Section Logging
- DNS Extended Logging
- DNS Error Logging

DNS query logging

You can configure a Citrix ADC appliance to log only the DNS queries that are received by the DNS endpoints on the appliance.

Note: If errors occur during processing of a query, they are logged if this option is set in the DNS profile.

Following is an example of a query log message:

```
1 DNS DNS_QUERY 143 0 :  U:10.102.27.70#61297:10.102.27.73#53/22142/Q/
2 (RD)/NO/1/0/0/0#test.com./1#
3 <!--NeedCopy-->
```


DNS answer section logging

You can configure a Citrix ADC appliance to log all the **Answer** sections in the DNS responses that the appliance sends to the client. DNS Answer Section logging is useful when the Citrix ADC is configured as a DNS resolver, or in GLSB use cases.

Following is an example of a DNS answer section log:

```
DNS DNS_RESPONSE 6678 0 :  U:100.100.100.210#32776:100.100.100.10#
53/61373/Q/(RD,AA,RA,R)/NO/1/1/2/4#n1.citrix.com1./
28#ANS#AAAA/120/1111:2345:6789:ffab:abcd:effa:1234:3212##
<!--NeedCopy-->
```

DNS extended logging

To configure a Citrix ADC appliance to log Authority and **Additional** sections in the DNS responses, enable Extended logging with Answer Section logging.

Note: If errors occur during processing of either queries or responses, the errors are logged if this option is set in the DNS profile.

Following is an example of a message logged when the cache lookup is completed and the response is embedded in the packet:

```
DNS DNS_RESPONSE 2252 0 :  T:100.100.100.118#21411:100.100.100.10
#53/48537/Q/(RD,AA,CD,RA,R)/NO/1/1/2/6#a1.citrix.com1./1#ANS#A/
120/1.1.1.1##AUTH#citrix.com1/NS/120/n2.citrix.com1#n1.citrix.com1##ADD
   #n1.citrix.com1
/A/120/1.1.1.1#1.1.1.2##n1.citrix.com1/AAAA/120/
1111:2345:6789:ffab:abcd:effa:1234:3212##n2.citrix.com1/A/120/2.1.1.2
##n2.citrix.com1/AAAA/120/2222:faff:3212:8976:123:1241:64:ff9b##OPT
   /0/1280/DO##
<!--NeedCopy-->
```

DNS error logging

You can configure a Citrix ADC appliance to log the errors or failures that occur when it processes a DNS query or response. For these errors, the appliance logs the DNS header, **Question** sections and OPT records.

Following is an example of a message logged when an error occurs during processing of a DNS request or response:

```
DNS DNS_ERROR 149 0 :  U:10.102.27.70#27832:10.102.27.73#53/61153/Q/
(RD)/NO/1/0/0/0#test.com./1140#Packet Dropped
```



```
<!--NeedCopy-->
```

Policy based logging

You can configure custom logging based on DNS expressions by configuring the logAction on DNS policies, Rewrite, or Responder policies. You can specify that logging occurs only when a particular DNS policy evaluates to true. For more information, see Configure policy based logging for DNS.

Understand the Citrix ADC syslog log message format

Citrix ADC appliance log DNS requests and responses in the following Syslog format:

```
<transport> :<client IP>#<client ephemeral port>:<DNS endpoint IP>#<
    port>
 : <query id> /opcode/header flags/rcode/question section count/answer
     section count
/ auth section count / additional section count #<queried domain name>
/<queried type>#...
<!--NeedCopy-->
```

- **<transport>:**
  - T = TCP
  - U = UDP
- **<client IP>#< client ephemeral port >:** DNS client IP address and port number
- **<DNS endpoint IP>#<port>:** Citrix ADC DNS endpoint IP address and port number
- **<query id>:**
  Query ID
- **<opcode>:** Operation code. Supported Values:
  - Q: query
  - I: inverse query
  - S: status
  - X0: unassigned
  - N: notify
  - U: update
  - X1-10: unassigned values
- **<header flags>:** Flags. Supported Values:
  - RD: recursion desired
  - TC: truncated

  - AA: authoritative response
  - CD: check disabled
  - AD: authenticated data
  - Z: unassigned
  - RA: recursion available
  - R: response

- **<rcode>:** Response Code. Supported Values:
  - NO: no error
  - F format error
  - S: server failure
  - NX: non-existent domain
  - NI: not implemented
  - R: query refused
  - YX: Name Exists when it must not
  - YXR: RR Set Exists when it must not
  - NXR: RR Set that must exist does not
  - NAS: Server Not Authoritative for zone
  - NA: Not Authorized
  - NZ: Name not contained in zone
  - X1-5: unassigned
- **/question section count/answer section count/auth section count/additional section count:** Question section, Authority section count, and **Additional** section count in the DNS request
- **<queried domain name>/<queried type>:** Queried domain and queried type in the DNS request
- **#ANS#<record type>/<ttl>/.. #AUTH#<domain name>/<record type>/<ttl>.. #ADD#<domain name>/<record type>/<ttl>...:**

  In DNS responses:

  Answer Section is logged if answer section logging is enabled in the DNS profile. Authority and **Additional** sections are logged if extended logging is enabled in the DNS profile. The log format would differ depending on the type of record. For more information see Understanding the Record Logging Format.

  - ANS: answer section
  - AUTH: authority
  - ADD: **Additional** section
- **OPT/<edns version>/UDP max payload size/DO:** OPT record format in the DNS log

- **OPT/<EDNS version>/<UDP payload size>/<“DO”or empty based on whether DNSSEC OK bit is set or not>/<value of RDLEN>/ECS/<Q/R>/<option length>/<Family>/<Source Prefix-Length>/<Scope Prefix-Length>/<ECS Address>:**

  If the DNS query or response includes the EDNS Client Subnet (ECS) option, then that is also logged in the OPT record format in the DNS log file.

  When a DNS query with an ECS option that includes either an IPv4 or IPv6 address is sent, the ECS option is logged with either of the following options;

  - “ECS/Q” indicating that the values in the log are from the query
  - “ECS/R” indicating that the values in the log are from the response.

  The value of Scope Prefix-Length is also set appropriately. In the DNS Query, it is set to zero, and for response, it is set to the calculated value.

  The following table describes the logged details in various scenarios:

Scenario	ECS option set in the DNS Query	ECS option set in the DNS Response	Logged Details
Both query logging and extended logging enabled	Yes	Yes	ECS option is logged with the string “ECS/R/” and the Scope Prefix-Length is set to the calculated value.
Both query logging and extended logging enabled	Yes	No	ECS option is logged with the string “ECS/Q” and the Scope Prefix-Length is set to zero.
Query logging is enabled, but extended logging is not enabled	Yes	Yes	ECS option is logged with the string “ECS/Q/” and the Scope Prefix-Length is set to zero.
Query logging & extended logging are not enabled	Yes	Yes	ECS option is not logged.

Scenario	ECS option set in the DNS Query	ECS option set in the DNS Response	Logged Details
Query logging is enabled, but extended logging is not enabled	Yes	No	ECS option is logged with the string “ECS/Q/” and the Scope Prefix-Length is set to zero.
Query logging is not enabled, but extended logging is enabled	Yes	Yes	ECS option is logged with the string “ECS/R/” and the Scope Prefix-Length is set to the calculated value.
Query logging is not enabled, but extended logging is enabled	Yes	No	ECS option is not logged.

Understand the record logging format

Following is an example of the record logging format in a Syslog message:

```
<domainname>/<record type>/ <record ttl> / <resource record data>#<
   resource record data>#......##
<!--NeedCopy-->
```

Where:

Record Type	Sample Format	Resource Record Data / Format
Address (A) record	A/5/1.1.1.1#1.1.1.2#1.1.1.3##	IPv4 address
AAAA record	AAAA/5/1::1#1::2#1::3##	IPv6 address
SOA record	SOA/3600/ns1.dnslogging.test./	Origin server, contact, and other details. Resource record format is: <originServer>/<contact>/<serial number>/<refresh rate>/<retry>/<expire>/<minimum>##

Record Type	Sample Format	Resource Record Data / Format
NS record	NS/5/ns1.dnslogging.test	Host name of the nameserver.
MX record	#MX/5/10/host1.dnslogging.test	Preference followed by mail exchange server host name
CNAME record logging	CNAME/5/host1.dnslogging.test.##	Canonical name
SRV record	SRV/5/1/2/3/host1.dnslogging.t .##	Resource record format: <priority>/<weight>/<port>/<target>#
TXT record	TXT/5/dns+logging##	Data comprises all the texts.
NAPTR record	NAPTR/5/10/11////dnslogging#. ##	Resource record format: <order>/<preference>/<flags>/<services>/<re expression>/<replacement string>#
DNSKEY record	DNSKEY/5/1/3/5/AwEAAanP0K+i5... I2Ccx6JZgiDBZhSON	Resource record format: <flags>/<protocol>/<algorithm>/<public key in base64 encoding>#
PTR record	PTR/3600/test.com.#test4.com.	Domain name

Limitations of DNS logging

DNS logging has the following limitations:

- If response logging is enabled, only the following record types are logged:
  - Address (A) record
  - AAAA record
  - SOA record
  - NS record
  - MX record
  - CNAME record
  - SRV record
  - TXT record
  - NAPTR record
  - DNSKEY record
  - PTR record

  For all other record types, only L3/L4 parameters, DNS Header, and Question section are logged.
- RRSIG records are not logged even if response logging is enabled.

- DNS64 is not supported.
- DNS proactive update requests or responses are logged according to the settings in the default profile.
- On the virtual server, if sessionless option and response logging is enabled, L3/L4 parameters, DNS Header, and DNS Question section are logged instead of the response.
- The maximum size of the syslog message is 1024 bytes.
- If you have set a DNS profile for a DNS policy with action type Rewrite Response, the Citrix ADC appliance does not log the query or the manipulated responses. To log the required information you must use an audit message action in the DNS policy.
- DNS transactions that are due to DNS monitoring traffic are not logged.

Configuring DNS logging

Following is an overview of configuring DNS logging:

1. Create a Syslog action and enable DNS in the action.
2. Create a Syslog policy and specify the Syslog action in the policy.
3. Globally bind the Syslog policy to enable logging of all Citrix ADC system events. Or, bind the Syslog policy to a specific load balancing virtual server.
4. Create a DNS profile and define any of the following types of logging that you want to enable:
   - DNS Query Logging
   - DNS Answer Section Logging
   - DNS Extended Logging
   - DNS Error Logging
5. Configure any of the following, based on your requirement:
   - DNS service and virtual server for DNS
   - ADNS service
   - Citrix ADC as a forwarder
   - Citrix ADC as a resolver
6. Set the created DNS profile to one of the DNS entities.

Configure DNS logging for Citrix ADC configured as DNS Proxy by using the CLI

1. Add a syslog action and enable DNS in the action. At the command prompt, type:

```
add audit syslogAction <name> (<serverIP> | -lbVserverName <string
    >) [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat <
    dateFormat>] [-logFacility <logFacility>] [-tcp ( NONE | ALL )]
     [-acl ( ENABLED | DISABLED )] [-timeZone ( GMT_TIME |
    LOCAL_TIME )] [-userDefinedAuditlog ( YES | NO )] [-
```



```
      appflowExport ( ENABLED |DISABLED )] [-lsn ( ENABLED | DISABLED
       )] [-alg ( ENABLED | DISABLED )] [-transport ( TCP | UDP )] [-
      tcpProfileName <string>] [-maxLogDataSizeToHold <
      positive_integer>] [-dns ( ENABLED | DISABLED)]
2 <!--NeedCopy-->
```

Example:

```
add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
LOCAL_TIME -dns ENABLED
```

2. Create a syslog policy and specify the created syslog action in the policy. At the command prompt, type:

   ```
   add audit syslogPolicy <name> <rule> <action>
   ```

   Example:

   ```
   add audit syslogPolicy syslogpol1 ns_true nssyslogact1
   ```

3. Bind the syslog policy globally. At the command prompt, type:

   ```
   bind system global [<policyName> [-priority <positive_integer>]]
   ```

   Example:

   ```
   bind system global syslogpol1
   ```

4. Create a DNS profile and enable any of the following type of logs that you want to configure:

   - DNS Query Logging
   - DNS Answer Section Logging
   - DNS Extended Logging
   - DNS Error Logging

   At the command prompt, type:

   ```
   add dns profile <dnsProfileName> [-dnsQueryLogging ( ENABLED | DISABLED
    )] [-dnsAnswerSecLogging ( ENABLED | DISABLED )] [-dnsExtendedLogging
    (ENABLED | DISABLED )] [-dnsErrorLogging ( ENABLED | DISABLED )] [-
   cacheRecords ( ENABLED | DISABLED )] [-cacheNegativeResponses ( ENABLED
    | DISABLED )]
   ```

   Example:

   ```
   add dns profile dnsprofile1 -dnsQueryLogging ENABLED
   ```

5. Configure service of type DNS. At the command prompt, type:

   ```
   add service <name> <serverName> <serviceType> <port>
   ```


Example:

```
add service svc1 10.102.84.140 dns 53
```

6. Configure a load balancing virtual server of service type DNS.

   ```
   add lb vserver <name> <serviceType> <ip> <port>
   ```

   Example:

   ```
   add lb vserver lb1 dns 100.100.100.10 53
   ```

7. Bind the service to the virtual server. At the command prompt, type:

   ```
   bind lb vserver <name> <serviceName>
   ```

   Example:

   ```
   bind lb vserver lb1 svc1
   ```

8. Set the created DNS profile to the virtual server. At the command prompt, type:

   ```
   set lb vserver <name> [ - dnsProfileName <string>]
   ```

   Example:

   ```
   set lb vserver lb1 -dnsProfileName dnsprofile1
   ```

Sample DNS logging configuration for Citrix ADC appliance configured as DNS proxy

```
1  > add audit syslogAction nssyslogact1 10.102.151.136 -logLevel
2  CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -
       timeZone
3  LOCAL_TIME -dns ENABLED
4  Done
5  > add audit syslogPolicy syslogpol1 ns_true nssyslogact1
6  Done
7  > bind system global syslogpol1
8  Done
9  > add dns profile dnsprofile1 -dnsqueryLogging ENABLED
10 Done
11 > add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
12 Done
13 > add service svc1 10.102.84.140 dns 53
14 Done
15 > bind lb vserver lb1 svc1
16 Done
17 <!--NeedCopy-->
```


Sample DNS logging configuration for Citrix ADC appliance configured as ADNS

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

Sample DNS logging configuration for Citrix ADC appliance configured as a forwarder

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add dns nameserver 8.8.8.8  - dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

Sample DNS logging configuration for Citrix ADC appliance configured as a resolver

```
> add audit syslogAction nssyslogact1 10.102.151.136
-logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -
    logFacility LOCAL4
```



```
-timeZone LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> set dns parameter -recursion enABLED
Done
> add nameserver 1.1.1.100 -local dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

Configure policy based logging for DNS

Policy based logging enables you to specify a format for log messages. The contents of a log message are defined by using a default syntax expression. When the message action specified in the policy is performed, the Citrix ADC appliance constructs the log message from the expression and writes the message to the log file. You can configure the appliance to log only when a particular DNS policy evaluates to True.

> **Note**
>
> If you have set a DNS policy with a DNS profile for the request side, the Citrix ADC appliance logs only the query.

To configure policy based logging for a DNS policy, you must first configure an audit message action. For more information about configuring an audit message action, see Configure the NetScaler appliance for audit logging. After configuring the audit message action, specify the message action in a DNS policy.

Configure policy based logging for a DNS policy by using the CLI

At the command prompt, type the following commands to configure policy based logging for a DNS policy and verify the configuration:

```
-  add dns action <actionName> <actionType> [-IPAddress <ip_addr|
   ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
   ...] [-TTL <secs>] [-dnsProfileName <string>]
-  set dns policy <name> [<rule>] [-actionName <string>] [-logAction <
   string>]
-  show dns policy [<name>]
<!--NeedCopy-->
```


Example 1:

In a GSLB deployment, if you want to respond with different IP addresses to the client requests coming from a particular subnet, instead of responding with IP addresses used for general purposes (such as the IP addresses of internal users), you can configure a DNS policy with the action type as DNS view. In this case, you can configure DNS logging on the specified DNS action such that you can log the specific responses.

```
> add dns profile dns_prof1 -dnsqueryLogging enABLED -
   dnsanswerSecLogging enABLED
 Done
> add dns view dns_view1
 Done
> add dns action dns_act1 viewName -view dns_view1  - dnsprofilename
   dns_prof1
 Done
> add dns policy dns_pol1 "CLIENT.IP.SRC.APPLY_MASK(255.255.255.0).EQ
   (100.100.100.0)”
dns_act1
 Done
> bind dns global dns_pol1 100 -gotoPriorityExpression END -type
   REQ_DEFAULT
 Done
> bind gslb service site_1_svc -viewName dns_view1 123.1.1.1
 Done
> bind gslb service site_5_svc -view dns_view1 132.1.1.1
 Done
<!--NeedCopy-->
```

Note: In the preceding configuration, if you query for the domain configured on a GSLB virtual server, for example, *sampletest.com*, all the internal users of subnet 100.100.100.0/24 are served with the DNS view IP addresses, and the responses are logged. Client requests for other subnets are not logged.

Example 2:

If you want to log only the queries for the domain *example.com*, you can create a DNS profile with query logging enabled and set the DNS profile to a DNS action with the action type **NOOP**, and then create a DNS policy and set the DNS action. For example:

```
>add dns profile query_logging -dnsqueryLogging ENABLED
Done
>add dns action dns_act1 NOOP -dnsprofileName query_logging
Done
>add dns policy dns_pol1 DNS.REQ.QUESTION.DOMAIN.EQ("example.com")
   dns_act1
Done
```



```
<!--NeedCopy-->
```

Configuring DNS suffixes

September 14, 2021

You can configure DNS suffixes that enable the Citrix ADC appliance to complete non-fully qualified domain names during name resolution. For example, while resolving a not fully qualified domain name abc, if a DNS suffix example.com is configured, the appliance appends the suffix to the domain name. Then it resolves the domain name. In this case, it would resolve abc.example.com. If DNS suffixes are not configured, the appliance appends a period to the non-fully qualified domain names and resolves the domain name.

Create DNS suffixes

DNS suffixes have significance and are valid only when the Citrix ADC is configured as an end resolver or forwarder. You can specify a suffix of up to 127 characters.

Note: The order of DNS suffixes is important. The ADC appliance tries the configured suffixes in a serial order and stops when it gets a successful response for a suffix.

Create DNS suffixes by using the CLI

At the command prompt, type the following commands to create a DNS suffix and verify the configuration:

```
-  add dns suffix <dnsSuffix>
-  show dns suffix <dnsSuffix>
<!--NeedCopy-->
```

Example:

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)      Suffix: example.com
Done

<!--NeedCopy-->
```

To remove a DNS suffix by using the Citrix ADC command line, at the command prompt, type the `rm dns suffix` command and the name of the DNS suffix.

Create DNS suffixes by using the GUI

Navigate to **Traffic Management > DNS > DNS Suffix** and create DNS suffixes.

DNS ANY query

September 14, 2021

An ANY query is a type of DNS query that retrieves all records available for a domain name. The ANY query must be sent to a name server that is authoritative for a domain.

Behavior in ADNS mode

In the ADNS mode, the Citrix ADC appliance returns the records held in its local cache. If there are no records in the cache, the appliance returns the NXDOMAIN (negative) response.

If the Citrix ADC can match the domain delegation records, it returns the NS records. Otherwise, it returns the NS records of the root domain.

Behavior in DNS proxy mode

In proxy mode, the Citrix ADC appliance checks its local cache. If there are no records in the cache, the appliance passes the query to the server.

Behavior for Global Server Load Balancing (GSLB) domains

If a GSLB domain is configured on the ADC appliance and an ANY query is sent for the GSLB (site) domain, the appliance returns the IP address of the GSLB service. It selects this service through a load balancing decision. If the multiple IP response (MIR) option is enabled, the IP addresses of all GSLB services are sent.

For the Citrix ADC to return these records when it responds to the ANY query, all records corresponding to a GSLB domain must be configured on the Citrix ADC.

> **Note**
>
> If records for a domain are distributed between the Citrix ADC and a server, only records configured on the Citrix ADC are returned.

The Citrix ADC provides the option to configure DNS views and DNS policies. These views and policies are used for performing global server load balancing. For more information, see Global Server Load Balancing.

Configure negative caching of DNS records

September 14, 2021

The Citrix ADC appliance supports caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called negative caching. Negative caching helps speed up responses to queries about a domain.

> **Note:**
>
> Negative caching is supported only when the back-end server is configured as an authoritative DNS (ADNS) server for the queried domain.

A negative response can be one of the following:

- NXDOMAIN error message — The authoritative DNS servers respond with the NXDOMAIN error message when the queried domain name does not have any records configured on the server. This message implies that the queried domain is an invalid or a non-existent domain name.
- NODATA error message — If the domain name in the query is valid but records of the given type are not available, the appliance sends a NODATA error message.

When negative caching is enabled, the appliance caches the negative response from the DNS server and serves the future requests from the cache only. This action helps speed up responses to queries and also to reduce the back-end DNS traffic. Negative caching can be used in all deployments, that is, when a Citrix ADC appliance is serving as a proxy, as an end resolver, or as a forwarder.

You can enable or disable negative caching using a DNS profile, for more information see, DNS profiles. By default, negative caching is enabled in the default DNS profile (**default**-dns-profile) that are bound by default to a DNS virtual server or in the newly created DNS profile.

Enable or disable negative caching by using the CLI

At the command prompt, type the following commands to enable or disable negative caching and verify the configuration:

```
1 -  add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED
     )] [-cacheNegativeResponses (ENABLED | DISABLED )]
2 -  show dns profile [<dnsProfileName>]
3 <!--NeedCopy-->
```

Example of a default DNS profile:

```
1 > sh dns profile default-dns-profile
2     1)   default-dns-profile
```



```
3         Query logging : DISABLED        Answer section logging :
             DISABLED
4         Extended logging : DISABLED     Error logging : DISABLED
5         Cache Records : ENABLED     Cache Negative Responses: ENABLED
6  Done
7  <!--NeedCopy-->
```

Example of a newly created DNS profile:

```
1  > add dnsprofile dns_profile1 -cacheRecords ENABLED -
      cacheNegativeResponses ENABLED
2  Done
3  > show dns profile dns_profile1
4      1)   dns_profile1
5         Query logging : DISABLED        Answer section logging :
             DISABLED
6         Extended logging : DISABLED     Error logging : DISABLED
7         Cache Records : ENABLED     Cache Negative Responses: ENABLED
8  Done
9  <!--NeedCopy-->
```

Specify service or virtual server level DNS parameters by using the CLI

At the command prompt, perform the following:

1. Configure the DNS profile.

   ```
   add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED )]
    [-cacheNegativeResponses (ENABLED | DISABLED )]
   ```

2. Bind the DNS profile to the service or virtual server.

   To bind the DNS profile to the service:

   ```
   set service <name> [-dnsProfileName <string>]
   ```

Example:

```
1  >set service service1 -dnsProfileName dns_profile1
2  Done
3  <!--NeedCopy-->
```

To bind the DNS profile to the virtual server:

```
set lb vserver <name> [-dnsProfileName <string>]
```

Example:


```
>set lb vserver lbvserver1 -dnsProfileName dns_profile1
Done
<!--NeedCopy-->
```

Specify service or virtual server level DNS parameters by using the GUI

1. Configure the HTTP profile.

   Navigate to **System** > **Profiles**> **DNS Profile**, and create the DNS profile.

2. Bind the HTTP profile to the service or virtual server.

   Navigate to **Traffic Management** > **Load Balancing**> **Services/Virtual Servers**, and create the DNS profile, that must be bound to the service or the virtual server.

Rate limiting negative response served by the appliance

You can set a threshold for negative responses being served by the Citrix ADC appliance from the cache. When the threshold is set, the appliance serves the response from the cache until the threshold is reached. Once the threshold is reached, the appliance drops the requests instead of responding with an NXDOMAIN response.

Setting a rate limit for negative responses has the following advantages.

- Save the resources on the Citrix ADC appliance.
- Prevent any malicious queries for non-existent domain names.

> **Note:** You can set a threshold for negative responses only for the domains for which the ADC appliance is configured as an authoritative domain name server. You cannot set a threshold for cached records received from the authoritative back-end name servers.

Rate limiting negative response served by the cache by using the CLI

At the command prompt, type

```
set dns parameter -NXDOMainRateLimitThreshold <positive-integer>
<!--NeedCopy-->
```

Example:

```
set dns parameter -NXDOMainRateLimitThreshold 1000
<!--NeedCopy-->
```


NXDOMainRateLimitThreshold: When this parameter is set to a positive integer value, responses are served from the cache until this threshold (in seconds) is reached. Once the threshold exceeds, the requests are dropped. The threshold configured is per packet engine.

Rate limiting negative response served by the cache by using the GUI

1. Navigate to **Traffic Management > DNS** and click **Change DNS Settings**.
2. In the **Configure DNS parameters** page, in the **NXDOMAIN Rate Limit Threshold** field, enter the threshold value until which the responses must be served from the cache.

Note: The value in the **NXDOMAIN Threshold Crossed** displays the number of times the requests are dropped after the threshold is reached.

Cache EDNS0 client subnet data when the Citrix ADC appliance is in proxy mode

September 14, 2021

In Citrix ADC Proxy mode, if a back-end server that supports an EDNS0 Client Subnet (ECS) sends a response containing the ECS option, the Citrix ADC appliance does the following:

- It forwards the response as-is to the client and
- Stores the response in the cache, along with the client subnet information.

DNS requests that are from the same subnet of the same domain, and for which the server would send the same response, are then served from the cache.

Note:

- ECS caching is disabled by default. Enable caching of EDNS0 client-subnet data in the associated DNS profile.
- The number of subnets that you can cache for a domain is limited to the available subnet IDs, that is, 1270 in the Citrix ADC appliance. Optionally, you can set the limit to a lower number (minimum value: 1 ipv4/ipv6).

Enable caching of ECS responses by using the CLI

At the command prompt, type:

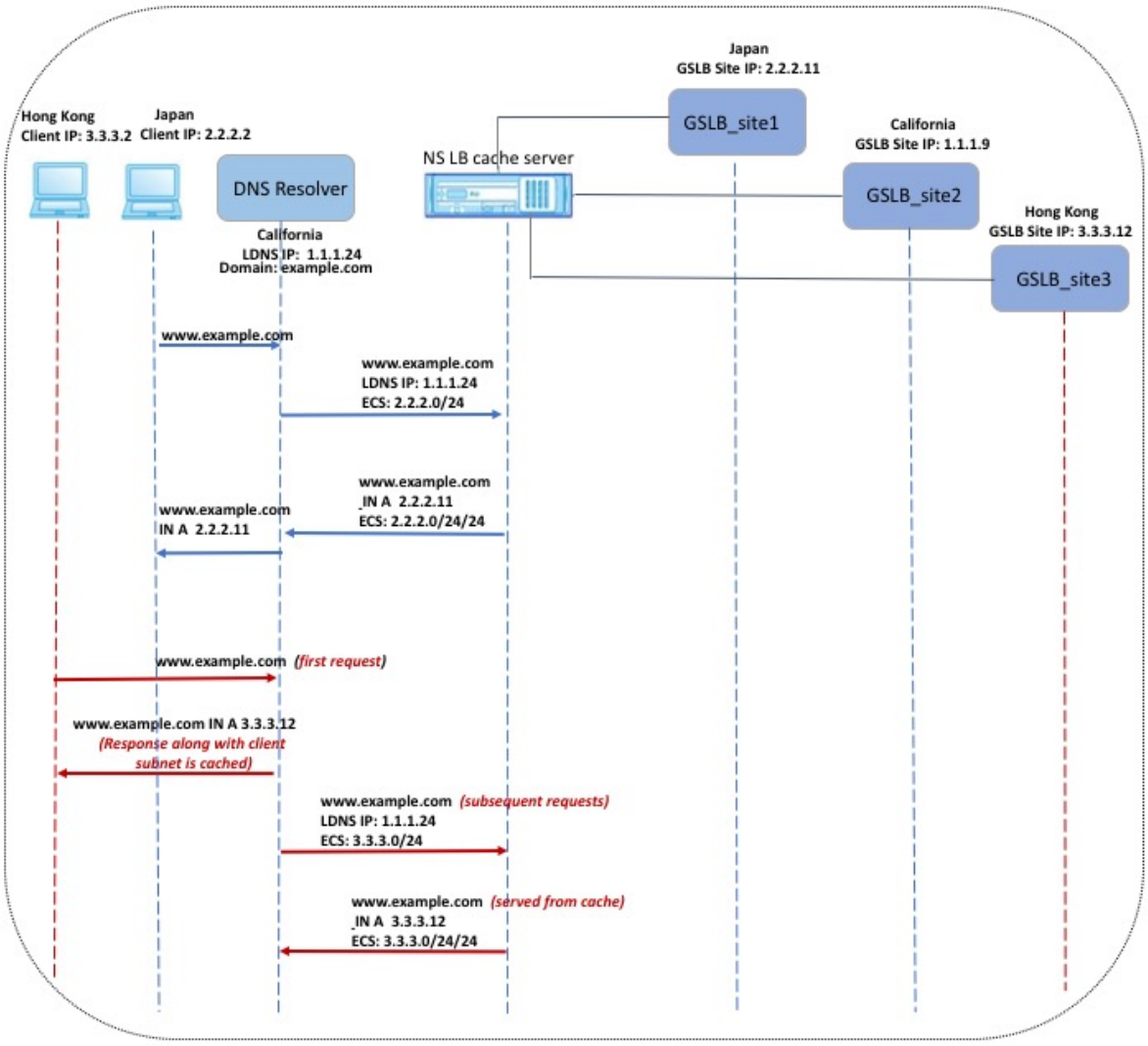
```
set dns profile <dnsProfileName> -cacheECSSubnet ( ENABLED | DISABLED )
```


Limit the number of subnets that can be cached per domain by using the CLI

At the command prompt, type:

```
set dns profile <dnsProfileName> -maxSubnetsPerDomain <positive_integer>
```

Example:



In the example shown in the preceding figure, the client at IP address 2.2.2.2 sends a query for www.example.com to the DNS resolver. The DNS resolver sends the following response:

www.example.com IN A, IP is 2.2.2.11, and ECS 2.2.2.0/24/24

At this point, the response and the client-subnet identifier (2.2.2.0/24) are cached. Further requests from the same subnet and domain are served from the cache.

For example, if the client's IP address is 2.2.2.100 and the query is for www.example.com, the query is served from the cache instead of being sent to the back-end server.

Domain name system security extensions

September 14, 2021

DNS Security Extensions (DNSSEC) is an Internet Engineering Task Force (IETF) standard. It aims to provide data integrity and data origin authentication in communications between name servers and clients while still transmitting UDP responses in clear text. DNSSEC specifies a mechanism that uses asymmetric key cryptography and a set of new resource records that are specific to its implementation.

The DNSSEC specification is described in:

- RFC 4033, "DNS Security Introduction and Requirements"
- RFC 4034, "Resource Records for the DNS Security Extensions"
- RFC 4035, "Protocol Modifications for the DNS Security Extensions"

The operational aspects of implementing DNSSEC within DNS are discussed in RFC 4641, "DNSSEC Operational Practices."

You can configure DNSSEC on the Citrix ADC. You can generate and import keys for signing DNS zones. You can configure DNSSEC for zones for which the Citrix ADC is authoritative. You can configure the ADC as a DNS proxy server for signed zones hosted on a farm of back-end name servers. If the ADC is authoritative for a subset of the records belonging to a zone for which the ADC is configured as a DNS proxy server, you can include the subset of records in the DNSSEC implementation.

Configure DNSSEC

September 14, 2021

Perform the following steps to configure DNSSEC:

1. Enable DNSSEC on the Citrix ADC appliance.
2. Create a zone signing key and a key signing key for the zone.
3. Add the two keys to the zone.
4. Sign the zone with the keys.

The Citrix ADC appliance does not act as a DNSSEC resolver. DNSSEC on the ADC is supported only in the following deployment scenarios:

1. ADNS—Citrix ADC is the ADNS and generates the signatures itself.
2. Proxy—Citrix ADC acts as a DNSSEC proxy. It is assumed that the Citrix ADC is placed in front of the ADNS/LDNS servers in a trusted mode. The ADC acts only as a proxy caching entity and does not validate any signatures.

Enable and disable DNSSEC

Enable DNSSEC on the Citrix ADC for the ADC to respond to DNSSEC-aware clients. By default, DNSSEC is enabled.

You can disable the DNSSEC feature if you do not want the Citrix ADC to respond to clients with DNSSEC-specific information.

Enable or disable DNSSEC by using the CLI

At the command prompt, type the following commands to enable or disable DNSSEC and verify the configuration:

```
-  set dns parameter -dnssec ( ENABLED | DISABLED )
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
         .
         .
         .
         DNSEC Extension: ENABLED
        Max DNS Pipeline Requests: 255
 Done

<!--NeedCopy-->
```

Enable or disable DNSSEC by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, click Change DNS settings.
3. In the **Configure DNS Parameters** dialog box, select or clear the **Enable DNSSEC Extension** check box.

Create DNS keys for a zone

For each DNS zone that you want to sign, you must create two pairs of asymmetric keys. One pair, called the zone signing key (ZSK), is used to sign all the resource record sets in the zone. The second

pair is called the key signing key (KSK) and is used to sign only the DNSKEY resource records in the zone.

When the ZSK and the KSK are created, the `suffix.key` is appended to the names of the public components of the keys. The `suffix.private` is appended to the names of their private components. The appending happens automatically.

The Citrix ADC also creates a Delegation Signer (DS) record and appends the suffix .ds to the name of the record. If the parent zone is a signed zone, you must publish the DS record in the parent zone to establish the chain of trust.

When you create a key, the key is stored in the `/nsconfig/dns/` directory, but it is not automatically published in the zone. After you create a key by using the `create dns key` command, you must explicitly publish the key in the zone by using the `add dns key` command. The process of generating a key is separate from the process of publishing the key in a zone to enable you to use alternative means to generate keys. For example, you can import keys generated by other key-generation programs (such as `bind-keygen`) by using the Secure FTP (SFTP) and then publish the keys in the zone. For more information about publishing a key in a zone, see Publish a DNS key in a zone.

Perform the steps described in this topic to create a zone signing key and then repeat the steps to create a key signing key. The example that follows the command syntax first creates a zone signing key pair for the zone example.com. The example then uses the command to create a key signing key pair for the zone.

From release 13.0 build 61.x, the Citrix ADC appliance now supports stronger crypto algorithms, such as RSASHA256 and RSASHA512, to authenticate a DNS zone. Previously, only the RSASHA1 algorithm was supported.

Create a DNS key by using the CLI

At the command prompt, type:

```
create dns key -zoneName <string> -keyType <keyType> -algorithm <algorithm>
 -keySize <positive_integer> -fileNamePrefix <string>
```

Example:

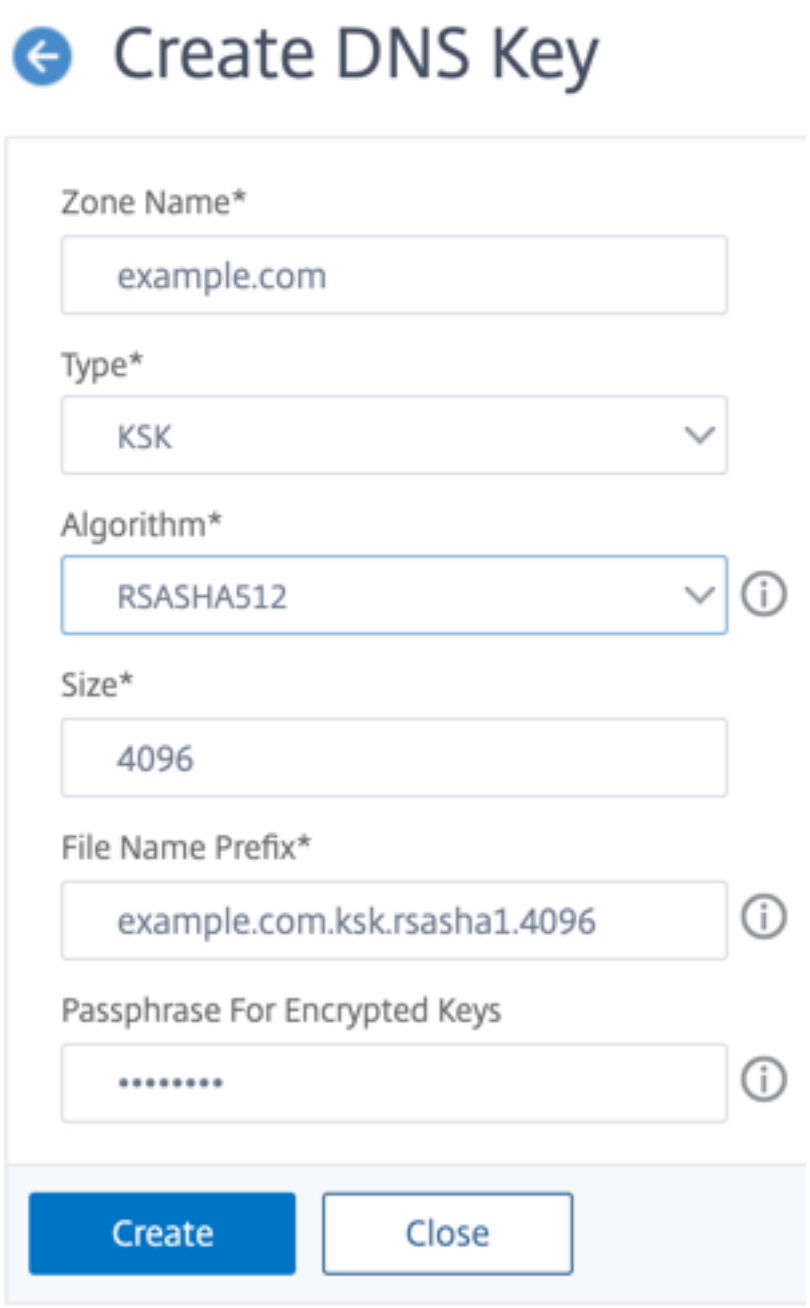
```
1 > create dns key -zoneName example.com -keyType zsk -algorithm
     RSASHA256 -keySize 1024 -fileNamePrefix example.com.zsk.rsasha1.1024
2 File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /
     nsconfig/dns/example.com.zsk.rsasha1.1024.private (private); /
     nsconfig/dns/example.com.zsk.rsasha1.1024.ds (ds)
3 This operation may take some time, Please wait...
4  Done
5 > create dns key -zoneName example.com -keyType ksk -algorithm
     RSASHA512 -keySize 4096 -fileNamePrefix example.com.ksk.rsasha1.4096
```



```
File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /
   nsconfig/dns/example.com.ksk.rsasha1.4096.private (private); /
   nsconfig/dns/example.com.ksk.rsasha1.4096.ds (ds)
This operation may take some time, Please wait...
 Done
<!--NeedCopy-->
```

Create a DNS key by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details area, click **Create DNS Key**.
3. Enter values for the different parameters and click **Create**.



Note: To modify the file name prefix of an existing key:

- Click the arrow next to the **Browse** button.
- Click either **Local** or **Appliance** (depending on whether the existing key is stored on your local computer or in the `/nsconfig/dns/` directory on the appliance)
- Browse to the location of the key, and then double-click the key.
  The **File Name Prefix** box is populated with only the prefix of the existing key. Modify the prefix accordingly.

Publish a DNS key in a zone

A key (zone signing key or key signing key) is published in a zone by adding the key to the ADC appliance. A key must be published in a zone before you sign the zone.

Before you publish a key in a zone, the key must be available in the **/nsconfig/dns/** directory. If you created the DNS key on another computer (for example, by using the `bind-keygen` program), ensure that the key is added to the `/nsconfig/dns/` directory. Then publish the key in the zone. Use the ADC GUI to add the key to the `/nsconfig/dns/` directory. Or, use some other program to import the key to the directory, such as the Secure FTP (SFTP).

Use the `add dns key` command for each public-private key pair that you want to publish in a given zone. If you created a ZSK pair and a KSK pair for a zone, use the `add dns key` command to first publish one of the key pairs in the zone. Repeat the command to publish the other key pair. For each key that you publish in a zone, a DNSKEY resource record is created in the zone.

The example that follows the command syntax first publishes the zone signing key pair (that was created for the example.com zone) in the zone. The example then uses the command to publish the key signing key pair in the zone.

Publish a key in a zone by using the CLI

At the command prompt, type the following command to publish a key in a zone and verify the configuration:

```
1 -  add dns key <keyName> <publickey> <privatekey> [-expires <
     positive_integer> [<units>]] [-notificationPeriod <positive_integer>
      [<units>]] [-TTL <secs>]
2 -  show dns zone [<zoneName> | -type <type>]
3 <!--NeedCopy-->
```

Example:

```
1 > add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.
     com.zsk.rsasha1.1024.private
2  Done
3 > add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.
     com.ksk.rsasha1.4096.private
4  Done
5 > show dns zone example.com
6          Zone Name : example.com
7          Proxy Mode : NO
8          Domain Name : example.com
9                  Record Types : NS SOA DNSKEY
10         Domain Name : ns1.example.com
```



```
                  Record Types : A
          Domain Name : ns2.example.com
                  Record Types : A
 Done
<!--NeedCopy-->
```

Publish a key in a DNS zone by using the GUI

Navigate to **Traffic Management > DNS > Keys**.

Note: For Public Key and Private Key, to add a key that is stored on your local computer, click the arrow next to the **Browse** button, click **Local**, browse to the location of the key, and then double-click the key.

Configure a DNS key

You can configure the parameters of a key that has been published in a zone. You can modify the key's expiry time period, notification period, and time-to-live (TTL) parameters. If you change the expiry time period of a key, the appliance automatically re-signs all the resource records in the zone with the key. The re-signing happens if the zone is signed with the particular key.

Configure a key by using the CLI

At the command prompt, type the following command to configure a key and verify the configuration:

```
-  set dns key <keyName> [-expires <positive_integer> [<units>]] [-
   notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
-  show dns key [<keyName>]
<!--NeedCopy-->
```

Example:

```
> set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3
   DAYS -TTL 3600
 Done
> show dns key example.com.ksk
1)      Key Name: example.com.ksk
        Expires: 30 DAYS        Notification: 3 DAYS    TTL: 3600
        Public Key File: example.com.ksk.rsasha1.4096.key
        Private Key File: example.com.ksk.rsasha1.4096.private
 Done
<!--NeedCopy-->
```


Configure a key by using the GUI

1. Navigate to **Traffic Management > DNS > Keys**.
2. In the details pane, click the key that you want to configure, and then click Open.
3. In the Configure DNS Key dialog box, modify the values of the following parameters as shown:
   - Expires—expires
   - Notification Period—notificationPeriod
   - TTL—TTL
4. Click OK.

Sign and unsign a DNS zone

To secure a DNS zone, you must sign the zone with the keys that have been published in the zone. When you sign a zone, the Citrix ADC creates a Next Secure (NSEC) resource record for each owner name. Then, it uses the key signing key to sign the DNSKEY resource record set. Finally, it uses the ZSK to sign all the resource record sets in the zone, including the DNSKEY resource record sets and NSEC resource record sets. Each sign operation results in a signature for the resource record sets in the zone. The signature is captured in a new resource record called the RRSIG resource record.

After you sign a zone, save the configuration.

Sign a zone by using the CLI

At the command prompt, type the following command to sign a zone and verify the configuration:

```
-  sign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
-  save config
<!--NeedCopy-->
```

Example:

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY RRSIG NSEC
         Domain Name : ns1.example.com
                 Record Types : A RRSIG NSEC
         Domain Name : ns2.example.com
```



```
                  Record Types : A RRSIG
          Domain Name : ns2.example.com
                  Record Types : RRSIG NSEC
Done
> save config
 Done
<!--NeedCopy-->
```

Unsign a zone by using the CLI

At the command prompt, type the following command to unsign a zone and verify the configuration:

```
-  unsign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
<!--NeedCopy-->
```

Example:

```
> unsign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
          Zone Name : example.com
          Proxy Mode : NO
          Domain Name : example.com
                  Record Types : NS SOA DNSKEY
          Domain Name : ns1.example.com
                  Record Types : A
          Domain Name : ns2.example.com
                  Record Types : A
 Done
<!--NeedCopy-->
```

Sign or unsign a zone by using the GUI

1. Navigate to **Traffic Management > DNS > Zones**.
2. In the details pane, click the zone that you want to sign, and then click Sign/Unsign.
3. In the Sign/Unsign DNS Zone dialog box, do one of the following:
   - To sign the zone, select the check boxes for the keys (zone signing key and key signing key) with which you want to sign the zone.
     You can sign the zone with more than one zone signing key or key signing key pair.
   - To unsign the zone, clear the check boxes for the keys (zone signing key and key signing key) with which you want to unsign the zone.

You can unsign the zone with more than one zone signing key or key signing key pair.

4. Click OK.

View the NSEC records for a given record in a zone

You can view the NSEC records that the Citrix ADC automatically creates for each owner name in the zone.

View the NSEC record for a given record in a zone by using the CLI

At the command prompt, type the following command to view the NSEC record for a given record in a zone:

```
show dns nsecRec [<hostName> | -type (ADNS | PROXY | ALL)]
```

Example:

```
> show dns nsecRec example.com
1)      Domain Name : example.com
        Next Nsec Name: ns1.example.com
        Record Types : NS SOA DNSKEY RRSIG NSEC
 Done
<!--NeedCopy-->
```

View the NSEC record for a given record in a zone by using the GUI

1. Navigate to **Traffic Management > DNS > Records > Next Secure Records**.
2. In the details pane, click the name of the record for which you want to view the NSEC record. The NSEC record for the record you select is displayed in the Details area.

Remove a DNS key

Remove a key from the zone in which it is published when the key has expired or if the key has been compromised. When you remove a key from the zone, the zone is automatically unsigned with the key. Removing the key with this command does not remove the key files present in the /nsconfig/dns/ directory. If the key files are no longer needed, they have to be explicitly removed from the directory.

Remove a key from the Citrix ADC by using the CLI

At the command prompt, type the following command to remove a key and verify the configuration:


```
-  rm dns key <keyName>
-  show dns key <keyName>
<!--NeedCopy-->
```

Example:

```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]

<!--NeedCopy-->
```

Remove a key from the Citrix ADC by using the GUI

1. Navigate to **Traffic Management > DNS > Keys**.
2. In the details pane, click the name of the key that you want to remove from the ADC, and then click Remove.

Configure DNSSEC when the Citrix ADC is authoritative for a zone

September 14, 2021

When the Citrix ADC is authoritative for a given zone, all the resource records in the zone are configured on the ADC. To sign the authoritative zone, you must create the zone signing and the key signing keys for the zone, add the keys to the ADC, and then sign the zone. For more information, see:

- Create DNS keys for a zone
- Publish a DNS key in a zone
- Sign and unsign a DNS zone.

If any GSLB domains configured on the ADC belong to the zone being signed, the GSLB domain names are signed along with the other records that belong to the zone.

After you sign a zone, responses to requests from DNSSEC-aware clients include the RRSIG resource records along with the requested resource records. DNSSEC must be enabled on the ADC. For more information about enabling DNSSEC, see Enable and disable DNSSEC.

Finally, after you configure DNSSEC for the authoritative zone, you must save the Citrix ADC configuration.

Configure DNSSEC for a zone for which the Citrix ADC is a DNS proxy server

September 14, 2021

The procedure for signing a zone for which the Citrix ADC is configured as a DNS proxy server depends on whether the ADC owns a subset of the zone information owned by the back-end name servers. If it does, the configuration is considered a partial zone ownership configuration. If the ADC does not own a subset of the zone information, the Citrix ADC configuration for managing the back-end servers is considered a zone-less DNS proxy server configuration. The basic DNSSEC configuration tasks for both Citrix ADC configurations are the same. However, signing the partial zone on the Citrix ADC requires some additional configuration steps.

Note: The terms zone-less proxy server configuration and partial zone are used only in the context of the Citrix ADC appliance.

Important: When configured in proxy mode, the ADC does not perform signature verification on DNSSEC responses before updating the cache.

If you configure the ADC as a DNS proxy to load balance DNSSEC aware resolvers (servers), you must set the Recursion Available option while configuring the DNS virtual server. If a DNSSEC query arrives with Checking Disabled (CD) bit set, the query is passed on to the server with the CD bit retained. The response from the server is not cached.

Configure DNSSEC for a zone-less DNS proxy server configuration

For a zone-less DNS proxy server configuration, zone signing must be performed on the back-end name servers. On the Citrix ADC, you configure the ADC as a DNS proxy server for the zone. Create a load balancing virtual server of protocol type DNS. Configure services on the ADC to represent the name servers. Then bind the services to the load balancing virtual server. For more information about these configuration tasks, see Configure the NetScaler as a DNS proxy server.

When a client sends the ADC a DNS request with the DNSSEC OK (DO) bit set, the ADC checks its cache for the requested information. If the resource records are not available in its cache, the ADC forwards the request to one of the DNS name servers. Then, it relays the response from the name server to the client. Also, the ADC caches the RRSIG resource records along with the response from the name server. Subsequent requests from DNSSEC-aware clients are served from the cache (including the RRSIG resource records), subject to the time-to-live (TTL) parameter. If a client sends a DNS request without setting the DO bit, the ADC responds with only the requested resource records. It does not include the RRSIG resource records that are specific to DNSSEC.

Configure DNSSEC for a partial zone ownership configuration

In some ADC configurations, even though the authority for a zone lies with the back-end name servers, a subset of the resource records belonging to the zone might be configured on the ADC. The ADC owns (or is authoritative for) only this subset of records. Such a subset of records can be considered to constitute a *partial zone* on the ADC. The ADC owns the partial zone. All other records are owned by the back-end name servers.

A typical partial zone configuration on the Citrix ADC is seen when:

- Global Server Load Balancing (GSLB) domains are configured on the ADC
- The GSLB domains are a part of a zone for which the back-end name servers are authoritative.

Signing a zone that includes only a partial zone on the ADC involves:

- Including the partial zone information in the back-end name server zone files
- Signing the zone on the back-end name servers
- Signing the partial zone on the ADC.

The same key set must be used to sign the zone on the name servers and the partial zone on the ADC.

Sign the zone on the back-end name servers

1. Include the resource records that are contained in the partial zone, in the zone files of the name servers.
2. Create keys and use the keys to sign the zone on the back-end name servers.

Sign the partial zone on the Citrix ADC

1. Create a zone with the name of the zone owned by the back-end name servers. When configuring the partial zone, set the proxyMode parameter to YES. This zone is the partial zone that contains the resource records owned by the ADC.

   For example, if the name of the zone that is configured on the back-end name servers is example.com, you must create a zone named example.com on the ADC. Set the proxyMode parameter to YES. For more information about adding a zone, see Configure a DNS zone.

   > **Note**
   >
   > Do not add SOA and NS records for the zone. These records must exist on the ADC for a zone for which the ADC is authoritative.

2. Import the keys (from one of the back-end name servers) to the ADC and then add them to the /nsconfig/dns/ directory. For more information about how you can import a key and add it to the ADC, see Publish a DNS key in a zone.

3. Sign the partial zone with the imported keys. When you sign the partial zone with the keys, the ADC generates RRSIG and NSEC records for the resource record sets and individual resource records in the partial zone, respectively. For more information about signing a zone, see sign and unsign a DNS zone.

Configure DNSSEC for global server load balancing (GSLB) domain names

September 14, 2021

If GSLB is configured on the Citrix ADC and the ADC is authoritative for the zone to which the GSLB domain names belong, all GLSB domain names are signed when the zone is signed. For more information about signing a zone for which the ADC is authoritative, see Configure DNSSEC when the Citrix ADC appliance is authoritative for a zone.

If the GSLB domains belong to a zone for which the back-end name servers are authoritative, you must:

- First sign the zone on the name servers.
- Then sign the partial zone on the ADC to complete the DNSSEC configuration for the zone.

For more information, see Configure DNSSEC for a partial zone ownership configuration.

Zone maintenance

September 14, 2021

From a DNSSEC perspective, zone maintenance involves rolling over Zone Signing Keys and Key Signing Keys when key expiry is imminent. These zone maintenance tasks must be performed manually. The zone is re-signed automatically and does not require any manual intervention.

Re-sign an updated zone

When a zone is updated (add a record or modify an existing record), the appliance automatically re-signs the new (or modified) record. If a zone contains multiple zone signing keys, the appliance re-signs the new (or modified) record with the key used to sign the zone.

Roll over DNSSEC keys

Note: Manually roll over the DNSSEC keys (KSK, ZSK) before they expire.

On the Citrix ADC, you can use the prepublish and double signature methods to perform a rollover of the Zone Signing Key and Key Signing Key. More information about these two rollover methods is available in RFC 4641, "DNSSEC Operational Practices."

The following topics map commands on the ADC to the steps in the rollover procedures discussed in RFC 4641.

The key expiry notification is sent through an SNMP trap called dnskeyExpiry. Three MIB variables, dnskeyName, dnskeyTimeToExpire, and dnskeyUnitsOfExpiry are sent along with the dnskeyExpiry SNMP trap. For more information, see *Citrix NetScaler SNMP OID Reference* at NetScaler 12.0 SNMP OID Reference.

Prepublish key rollover

RFC 4641, "DNSSEC Operational Practices" defines four stages for the prepublish-key rollover method: initial, new DNSKEY, new RRSIGs, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the ADC. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both Key Signing Keys and Zone Signing Keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.

  Example:

  Consider the key, example.com.zsk1, with which the zone example.com is signed. The zone contains only those RRSIGs generated by the example.com.zsk1 key, which is due for expiry. The Key Signing Key is example.com.ksk1.

- **Stage 2: New DNSKEY.** A new key is created and published in the zone. That is, the key is added to the ADC, but the zone is not signed with the new key until the pre-roll phase is complete. In this stage, the zone contains the old key, the new key, and the RRSIGs generated by the old key. Publishing the new key for the complete duration of the pre-roll phase gives the DNSKEY resource record corresponding to the new key time to propagate to the secondary name servers.

  Example:

  A new key example.com.zsk2 is added to the example.com zone. The zone is not signed with example.com.zsk2 until the pre-roll phase is complete. The example.com zone contains DNSKEY resource records for both example.com.zsk1 and example.com.zsk2.

  Citrix ADC commands:

  Perform the following tasks on the ADC:

  - Create a DNS key by using the `create dns key` command.

    For more information about creating a DNS key, including an example, see Create DNS keys for a zone.

  - Publish the new DNS key in the zone by using the `add dns key` command.

    For more information about publishing the key in the zone, including an example, see Publish a DNS key in a zone.

- **Stage 3: New RRSIGs.** The zone is signed with the new DNS key and then unsigned with the old DNS key. The old DNS key is not removed from the zone and remains published until the RRSIGs generated by the old key expire.

  Example:

  The zone is signed with example.com.zsk2 and then unsigned with example.com.zsk1. The zone continues to publish example.com.zsk1 until the RRSIGs generated by example.com.zsk1 expire.

  Citrix ADC commands:

  Perform the following tasks on the ADC:

  - Sign the zone with the new DNS key by using the `sign dns zone` command.
  - Unsign the zone with the old DNS key by using the `unsign dns zone` command.

  For more information about signing and unsigning a zone, including examples, see Sign and unsign a DNS zone.

- **Stage 4: DNSKEY Removal.** When the RRSIGs generated by the old DNS key expire, the old DNS key is removed from the zone.

  Example:

  The old DNS key example.com.zsk1 is removed from the example.com zone.

  Citrix ADC commands

  On the ADC, you remove the old DNS key by using the `rm dns key` command. For more information about removing a key from a zone, including an example, see Remove a DNS key.

Double signature key rollover

RFC 4641, “DNSSEC Operational Practices” defines three stages for double signature key rollover: initial, new DNSKEY, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the ADC. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both Key Signing Keys and Zone Signing Keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.

  Example:

  Consider the key, example.com.zsk1, with which the zone example.com is signed. The zone contains only those RRSIGs generated by the example.com.zsk1 key, which is due for expiry. The Key Signing Key is example.com.ksk1.

- **Stage 2: New DNSKEY.** The new key is published in the zone and the zone is signed with the new key. The zone contains the RRSIGs that are generated by the old and the new keys. The minimum duration for which the zone must contain both sets of RRSIGs is the time required for all the RRSIGs to expire.

  Example:

  A new key example.com.zsk2 is added to the example.com zone. The zone is signed with example.com.zsk2. The example.com zone now contains the RRSIGs generated from both keys.

  Citrix ADC commands

  Perform the following tasks on the ADC:

  - Create a DNS key by using the `create dns key` command.

    For more information about creating a DNS key, including an example, see Create DNS keys for a zone.

  - Publish the new key in the zone by using the `add dns key` command.

    For more information about publishing the key in the zone, including an example, see Publish a DNS key in a zone.

  - Sign the zone with the new key by using the `sign dns zone` command.

    For more information about signing a zone, including examples, see Sign and unsign a DNS zone.

- **Stage 3: DNSKEY Removal.** When the RRSIGs generated by the old DNS key expire, the old DNS key is removed from the zone.

  Example:

  The old DNS key example.com.zsk1 is removed from the example.com zone.

  Citrix ADC commands:

  On the ADC, you remove the old DNS key by using the `rm dns key` command.

  For more information about removing a key from a zone, including an example, see Remove a DNS key.

Offload DNSSEC operations to the Citrix ADC

September 14, 2021

For DNS zones for which your DNS servers are authoritative, DNSSEC operations can be offloaded to the ADC appliance. In a DNSSEC offloading deployment, a DNS server sends unsigned responses. The ADC signs the response dynamically before relaying it to the client. The ADC also caches the signed response. Apart from reducing the load on the DNS servers, offloading DNSSEC operations to the ADC gives you the following benefits:

- You can sign records that the DNS servers generate programmatically. Such records cannot be signed by routine zone signing operations performed on the DNS servers.
- You can serve signed responses to clients even if you have not implemented DNSSEC on your servers.

For setting up DNSSEC offloading, you must configure a DNS load balancing virtual server, configure services that represent the DNS servers, and then bind the services to the virtual server. For information about configuring a DNS load balancing virtual server, configuring services, and binding the services to the virtual server, see Configure a DNS zone.

Create a zone entity on the ADC for each DNS zone whose DNSSEC operations you want to offload. For each DNS zone, you must enable the Proxy Mode and DNSSEC Offload parameters. You can optionally configure NSEC record generation for an offloaded zone. To create a DNS zone entity for DNSSEC offloading, follow the instructions in this topic.

To complete the configuration, you must generate DNS keys for the zone, add the keys to the zone, and then sign the zone with the keys. This process is the same as for normal DNSSEC. For information about creating keys, adding keys to a zone, and signing the zone, see Domain name system security extensions.

After you configure DNS offloading, you must flush the DNS cache on the Citrix ADC. Flushing the DNS cache ensures that any unsigned records in the cache are removed and then replaced by signed records. For information about flushing the DNS cache, see Flush DNS records.

Enable DNSSEC offloading for a zone by using the CLI

At the command line, type the following commands to enable DNSSEC offloading for a zone and verify the configuration:

```
1 -  add dns zone <zoneName> -proxyMode YES -dnssecOffload ENABLED [-nsec
      ( ENABLED | DISABLED )
2 -  show dns zone
3 <!--NeedCopy-->
```


Example:

```
> add dns zone example.com -proxyMode YES -dnssecOffload ENABLED nsec
    ENABLED
 Done
> show dns zone example.com
     Zone Name : example.com
     Proxy Mode : YES
     DNSSEC Offload: ENABLED    NSEC: ENABLED
 Done
<!--NeedCopy-->
```

Enable DNSSEC offloading for a zone by using the GUI

1. Navigate to **Traffic Management > DNS > Zones**.
2. In the details pane, do one of the following:
   - To create a zone on the Citrix ADC, click Add.
   - To configure DNSSEC offloading for an existing zone, double-click the zone.
3. In the Create DNS Zone or Configure DNS Zone dialog box, select the Proxy Mode and DNSSEC Offload check boxes.
4. Optionally, if you want the Citrix ADC to generate NSEC records for the zone, select the NSEC check box.

Admin partition support for DNSSEC

September 14, 2021

In a partitioned Citrix ADC appliance, the DNS keys that are generated are stored in the following locations:

- Default partition: /nsconfig/dns/
- Non-default partition: /nsconfig/partitions/<partitionname>/dns/

You can now add a password to the DNS key. To add a password to the DNS key, you must first add the password in the `create dns key` command. Then provide the same password in the `add dns key` command when adding the DNS key to the ADC appliance. For example:

```
create dns key -zoneName com -keytype ksK -algorithm rsASHA1 -keysize 4096
- fileNamePrefix com.ksk.rsasha1.4096 -password 1jsfd3Wa

add dns key com.zsk.4096 /nsconfig/dns/com.zsk.rsasha1.4096.private -
password 1jsfd3Wa
```


Note:

- For a default partitioned environment, the keys are read from the default location/nsconfig/dns/. However, if the keys are stored in a different location, the path name has to be provided in the `add dns key -private` command. Example, `add dns key -private <path name>`.
- For a non-default partitioned environment, the keys are read from the default location `/nsconfig/partitions/<partitionname>/dns/`.

Supporting wildcard DNS domains

September 14, 2021

Wildcard DNS domains are used to handle requests for nonexistent domains and subdomains. In a zone, use wildcard domains to redirect queries for all nonexistent domains or subdomains to a particular server, instead of creating a separate Resource Record (RR) for each domain. The most common use of a wildcard DNS domain is to create a zone that can be used to forward mail from the internet to some other mail system.

In DNS resolution, wildcard RRs support the wildcard domain. The wildcard RRs are used to synthesize the responses to queries for a nonexistent domain name. For example, if you queried `http://image.example.com`, and the subdomain "image" did not exist, you might be redirected to example.com.

A wildcard record has an asterisk (*) character as the leftmost label of a domain name. For example, `*.example.com`. An asterisk at any other place in the domain name does signify a wildcard DNS record. For example, **`new`**`.*.example.com` is not a valid wildcard DNS record.

> **Note**
>
> - Wildcard domain is supported only when the Citrix ADC appliance is authoritative for the zone and is configured as an ADNS or a DNS proxy server.
> - Wildcard domain is not supported for NS and SOA records.
> - Wildcard domain cannot be applied when the query is in another zone.
> - Wildcard domain cannot be applied when the QNAME or a name between the wildcard domain and the QNAME is known to exist.

Example configuration

```
1 add dns soaRec example.com -originServer n1.example.com -contact admin.
     example.com
2
3 add dns nsRec example.com n1.example.com
```



```
4
5  add dns nsRec example.com n2.example.com
6
7  add dns zone example.com  -proxyMode no
8
9  add dns addrec www.example.com 2.2.2.2
10
11 add dns addrec *.example.com 10.10.10.10
12
13 add dns addrec *.example.com 10.10.10.11
14
15 add dns aaaarec *.example.com 2001::1
16 <!--NeedCopy-->
```

In the example, a wildcard domain name is added for an A and AAAA record.

When a query is received for a domain name that exists in the zone, the Citrix ADC appliance responds with the corresponding response. For example, for `www.example.com`, the appliance responds with 2.2.2.2 in the example.

For a nonexistent domain name that matches with a wildcard type, a synthesized response is delivered.

In the example, the Citrix ADC appliance responds with 10.10.10.10 and 10.10.10.11 for a domain name nonexist.example.com or xyz.example.com.

Wildcard synthesis is not applicable for a domain name that exists in the zone.

For example, for the query `www.example.com` and type AAAA, the Citrix ADC appliance does not synthesize with wildcard, because `www.example.com` exists with type A. In the example, the Citrix ADC appliance responds with a NODATA response.

For a query say abc.example.com and type AAAA, the Citrix ADC appliance responds with a synthesized response. For example, for `www.example.com`, the appliance responds with 2001::1 in the example.

Mitigate DNS DDoS attacks

September 14, 2021

DNS servers are one of the most critical components of a network, and they must be defended against attacks. One of the most basic types of DNS attacks is the DDoS attack. Attacks of this type are on the rise and can be destructive. You can do the following to mitigate DDoS attacks:

- Flush negative records.
- Restrict the time to live (TTL) of negative records.

- Preserve Citrix ADC memory by limiting the memory consumed by the DNS cache.
- Retain DNS records in the cache.
- Enable DNS cache bypass.

Flush negative records

A DNS attack fills the cache with negative records (NXDOMAIN and NODATA). As a result, responses to legitimate requests are not cached, so new requests are sent to a back-end server for DNS resolution. Responses are therefore delayed.

You can now flush the negative DNS records from the Citrix ADC appliance's DNS cache.

Flush negative cache records by using the CLI

At the command prompt, type:

```
flush dns proxyrecords -type (dnsRecordType | negRecType)NXDOMAIN | NODATA
```

Example:

```
flush dns proxyrecords -negRecType NODATA
```

Flush of negative cache records by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **DNS** > **Records**.
2. In the details pane, click **Flush Proxy Records**.
3. In the **Flush Type** box, select **Negative Records**.
4. In the **Negative Records Type** box, select either **NXDOMAIN** or **NODATA**.

Protection against random subdomain and NXDOMAIN attacks

To prevent random subdomain and NXDOMAIN attacks, you can restrict the DNS cache memory, and you can adjust the TTL values for negative records.

To limit the amount of memory consumed by the DNS cache, specify the maximum cache size (in MB), and also the cache size (in MB) for storing negative responses. When either limit is reached, no more entries are added to the cache. Also, syslog messages are logged and, if you have configured SNMP traps, SNMP traps are generated. If these limits are not set, caching continues until the system memory is exhausted.

A higher TTL value for negative records can result in storing records that are not valuable for a long time. A lower TTL value results in sending more requests to the back-end server.

The TTL of the negative record is set to a value that can be the lesser of the TTL value or the "Expires" value of the SOA record.

Note:

- This limitation is added per packet engine. For example, if the maxCacheSize is set to 5 MB and the appliance has 3 packet engines, the total cache size is 15 MB.
- The cache size for the negative records must be less than or equal to the maximum cache size.
- If you reduce the DNS cache memory limit to a value lower than the amount of data already cached, the cache size remains above the limit until the data ages out. That is, exceeds its TTL0 or is flushed (`flush dns proxyrecords` command, or Flush Proxy Records in the Citrix ADC GUI).
- To configure SNMP traps, see Configuring the NetScaler to Generate SNMP Traps.

Limit the memory consumed by the DNS Cache by using the CLI

At the command prompt, type:

```
set dns parameter -maxCacheSize <MBytes> -maxNegativeCacheSize <MBytes>
```

Example:

```
set dns parameter - maxCacheSize 100 -maxNegativeCacheSize 25
```

Limit the memory consumed by the DNS Cache by using the GUI

Navigate to **Configuration** > **Traffic Management** > **DNS**, click **Change DNS Settings**, and set the following parameters:

- Max Cache Size in MB
- Max Negative Cache Size in MB

Restrict the TTL of negative records by using the CLI

At the command prompt, type:

```
set dns parameter -maxnegcacheTTL <secs>
```

Example:

```
set dns parameter -maxnegcacheTTL 360
```

Restrict the TTL of negative records by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **DNS**.
2. Click **Change DNS Settings** and set the **Max Negative Cache TTL in sec** parameter.

Retain DNS records in the cache

An attack can flood the DNS cache with non-important entries but can cause flushing of the already cached legitimate records to make room for the new entries. To prevent attacks from filling the cache with invalid data, you can retain the legitimate records even after they exceed their TTL values.

If you enable the cacheNoExpire parameter, the records currently in the cache are retained until you disable the parameter.

Note:

- This option can be used only when the maximum cache size is specified (maxCacheSize parameter).
- If maxnegcacheTTL is configured and cacheNoExpire is enabled, cacheNoExpire takes priority.

Retain DNS records in the cache by using the CLI

At the command prompt, type:

```
set dns parameter -cacheNoExpire ( ENABLED | DISABLED)
```

Example:

```
set dns parameter -cacheNoExpire ENABLED
```

Retain DNS records in the cache by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **DNS** and click **Change DNS Settings**.
2. Select **Cache No Expire**.

Enable DNS cache bypass

For greater visibility and control of DNS requests, set the cacheHitBypass parameter to forward all requests to the back-end servers and allow the cache to be built but not used. After the cache is built, you can disable the parameter so that requests are served from the cache.

Enable DNS cache bypass by using the CLI

At the command prompt, type:

```
set dns parameter -cacheHitBypass ( ENABLED | DISABLED )
```

Example:

```
set dns parameter -cacheHitBypass ENABLED
```


Enable DNS cache bypass by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **DNS** and click **Change DNS Settings**.
2. Select **Cache Hit Bypass**.

Prevent the `Slowloris` attack

A DNS query spanning multiple packets, presents the potential threat of a `Slowloris` attack. The Citrix ADC appliance can silently drop DNS queries that are split into multiple packets.

You can set the `splitPktQueryProcessing` parameter to ALLOW or DROP a DNS query if the query is split into multiple packets.

Note: This setting is applicable only for DNS TCP.

Limit the DNS queries to a single packet by using the CLI

At the command prompt, type:

```
set dns parameter -splitPktQueryProcessing ( ALLOW | DROP )
```

Example:

```
set dns parameter -splitPktQueryProcessing DROP
```

Limit DNS queries to a single packet by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **DNS** and click **Change DNS Settings**.
2. In the **Split Packet Query Processing** box, choose **ALLOW** or **DROP**.

Collect statistics of the DNS responses served from the cache

You can collect statistics of the DNS responses served from the cache. Then use these statistics to create a threshold beyond which more DNS traffic is dropped, and enforce this threshold with a bandwidth based policy. Previously, bandwidth calculation for a DNS load balancing virtual server was not accurate, because the number of requests served from the cache was not reported.

In proxy mode, the statistics for Request bytes, Response bytes, Total Packets received, and Total Packets sent statistics are continuously updated. Previously, these statistics were not always updated, particularly for a DNS load balancing virtual server.

Proxy mode also now enables you to determine the number of DNS responses served from the cache. To collect these statistics, the following options have been added to the `stat lb vserver <DNSvirtualServerName>` command:

- **Requests** – Total number of requests received by the DNS or DNS_TCP virtual server. Includes the requests forwarded to the back end and the requests answered from the cache.
- **Vserver hits** –Total number of requests forwarded to the back end. The number of requests served from the cache is the difference between the total number of requests and the number of requests served from the virtual server.
- **Responses** – Total number of responses sent by this virtual server. For example, if a DNS LB virtual server received 5 DNS requests, forwarded 3 of them to the back end, and served 2 of them from the cache, the corresponding value of each of these statistics would be as follows:
  - **Vserver hits**: 3
  - **Requests**: 5
  - **Responses**: 5

Firewall Load Balancing

September 14, 2021

Firewall load balancing distributes traffic across multiple firewalls, providing fault tolerance and increased throughput. Firewall load balancing protects your network by:

- Dividing the load between the firewalls, which eliminates a single point of failure and allows the network to scale.
- Increasing high availability.

Configuring a Citrix ADC appliance for firewall load balancing is similar to configuring load balancing, with the exception that the recommended service type is ANY, recommended monitor type is PING, and the load balancing virtual server mode is set to MAC.

You can set up firewall load balancing in a sandwich, an enterprise, or multiple-firewall environment configuration. The sandwich environment is used for load balancing traffic entering the network from outside and traffic leaving the network to the internet and involves configuring two Citrix ADC appliances, one on each side of a set of firewalls. You configure an enterprise environment for load balancing traffic leaving the network to the internet. The enterprise environment involves configuring a single Citrix ADC appliance between the internal network and the firewalls that provide access to the Internet. The multiple-firewall environment is used for load balance traffic coming from another firewall. Having firewall load balancing enabled on both the sides of Citrix ADC appliance improves the traffic flow in both the egress and ingress direction and ensures faster processing of the traffic. The multiple-firewall environment involves configuring a Citrix ADC appliance sandwiched between two firewalls.

Important: If you configure static routes on the Citrix ADC appliance for the destination IP address and enable L3 mode, the Citrix ADC appliance uses its routing table to route the traffic instead of sending

the traffic to the load balancing vserver.

Note: For FTP to work, an additional virtual server or service should be configured on the Citrix ADC appliance with IP address and port as * and 21 respectively, and the service type specified as FTP. In this case, the Citrix ADC appliance manages the FTP protocol by accepting the FTP control connection, modifying the payload, and managing the data connection, all through the same firewall.

Firewall Load Balancing supports only some of the load balancing methods supported on the Citrix ADC appliance. Also, you can configure only a few types of persistence and monitors.

Firewall Load Balancing Methods

The following load balancing methods are supported for firewall load balancing.

- Least Connections
- Round Robin
- Least Packets
- Least Bandwidth
- Source IP Hash
- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port hash
- Least Response Time Method (LRTM)
- Custom Load

Firewall Persistence

Only SOURCEIP, DESTIP, and SOURCEIPDESTIP based persistence are supported for firewall load balancing.

Firewall Server Monitoring

Only PING and transparent monitors are supported in firewall load balancing. You can bind a PING monitor (default) to the backend service that represents the firewall. If a firewall is configured not to respond to ping packets, you can configure transparent monitors to monitor hosts on the trusted side through individual firewalls.

Sandwich Environment

September 14, 2021

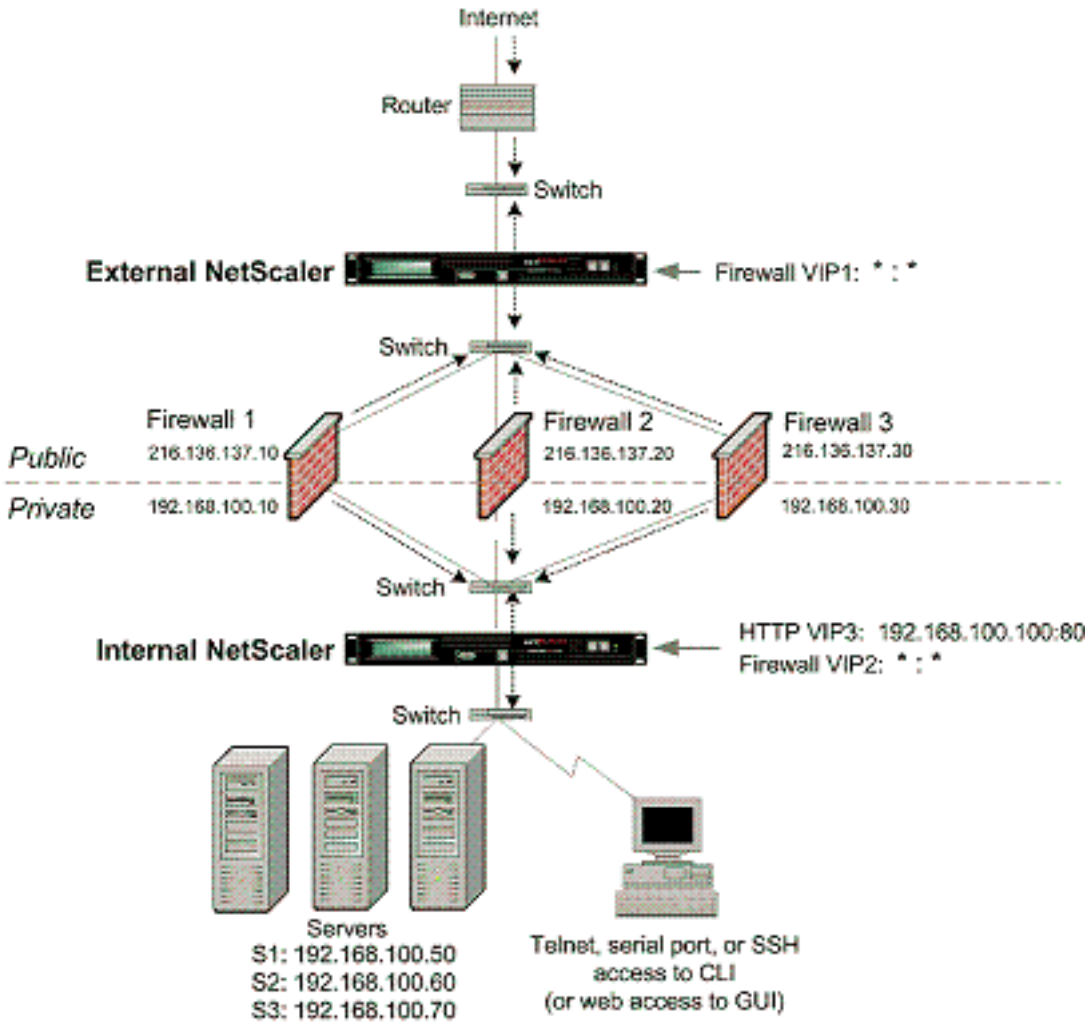
A Citrix ADC deployment in a sandwich mode can load balance network traffic through firewalls in both directions: ingress (traffic entering the network from the outside, such as the internet) and egress (traffic leaving the network to the internet).

In this setup, a Citrix ADC is located on each side of a set of firewalls. The Citrix ADC placed between the firewalls and the Internet, called the external Citrix ADC that handles ingress traffic selects the best firewall, based on the configured method. The Citrix ADC between the firewalls and the private network, called the internal Citrix ADC tracks the firewall from which the initial packet for a session is received. It then makes sure that all subsequent packets for that session are sent to the same firewall.

The internal Citrix ADC can be configured as a regular traffic manager to load balance traffic across the private network servers. This configuration also allows traffic originating from the private network (egress) to be load balanced across the firewalls.

The following diagram shows the sandwich firewall load balancing environment.

Figure 1. Firewall Load Balancing (Sandwich)



The service type ANY configures the Citrix ADC to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the External Citrix ADC in a Sandwich Environment

Perform the following tasks to configure the external Citrix ADC in a sandwich environment

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server for traffic coming from the Internet.
- Configure the virtual server in MAC rewrite mode.
- Bind services to the wildcard virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```


To enable load balancing by using the configuration utility

Navigate to **System > Settings** and, in **Configure Basic Features**, select **Load Balancing**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services** and add a service. Specify **ANY** in the **Protocol** field and * in the Port field.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the Citrix ADC appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor overrides the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
To bind a PING monitor, type the following command:
bind monitor PING fw-svc1
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Monitors**, and then create and bind a transparent monitor.

Configure a wildcard virtual server for traffic coming from the Internet

To configure a wildcard virtual server for traffic coming from the Internet by using the command line interface

At the command prompt, type:

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Virtual Servers** and create a wildcard virtual server. Specify **ANY** in the **Protocol** field and * in the Port field.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1).
2. Edit the **Basic Settings** section, and click **more**.
3. From the **Redirection Mode** drop-down list, select **MAC Based**.

Bind services to the wildcard virtual server

To bind a service to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a service to the wildcard virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers** and select the virtual server for which you want to bind the service.
2. Click in the **Services** section and select a service to bind.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. Ensure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
save ns config
show vserver
<!--NeedCopy-->
```

Example:

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 06:40:14 2010
        Time since last state change: 0 days, 00:00:11.240
        Effective State: UP  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: SRCIPDESTIPHASH
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: *) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
        fw-svc1 (10.102.29.251:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 10:04:50 2010
        Time since last state change: 0 days, 00:00:38.120
        Server Name: 10.102.29.251
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
```



```
30          Access Down Service: NO
31          TCP Buffering(TCPB): YES
32          HTTP Compression(CMP): NO
33          Idle timeout: Client: 120 sec   Server: 120 sec
34          Client IP: DISABLED
35          Cacheable: NO
36          SC: OFF
37          SP: OFF
38          Down state flush: ENABLED
39
40  1)      Monitor Name: monitor-HTTP-1
41                  State: DOWN     Weight: 1
42                  Probes: 5       Failed [Total: 5 Current: 5]
43                  Last response: Failure - Time out during TCP connection
                        establishment stage
44                  Response Time: 2000.0 millisec
45  2)      Monitor Name: ping
46                  State: UP       Weight: 1
47                  Probes: 3       Failed [Total: 0 Current: 0]
48                  Last response: Success - ICMP echo reply received.
49                  Response Time: 1.415 millisec
50   Done
51  <!--NeedCopy-->
```

Configuring the Internal Citrix ADC in a Sandwich Environment

Perform the following tasks to configure the internal Citrix ADC in a sandwich environment

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled. But they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

To enable load balancing by using the configuration utility

Navigate to **System > Settings** and, in Configure Basic Features, select **Load Balancing**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```


Example:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services** and add a service. Specify **ANY** in the **Protocol** field and * in the Port field.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the Citrix ADC appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor overrides the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```


To create and bind a transparent monitor by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors** and create a monitor.
2. In the **Create Monitor** dialog box, enter the required parameters, and select **Transparent**.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type:

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers** and create a wildcard virtual server.
2. Specify **ANY** in the Protocol field and ***** in the Port field.

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
4. In Protocol, select ANY, and in IP Address and Port, select *.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1).
2. Edit the **Basic Settings** section, and click **more**.
3. From the **Redirection Mode** drop-down list, select **MAC Based**.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

To configure a service for each virtual server by using the command line interface

At the command prompt, type:


```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

To configure a service for each virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Services**, and configure a service for each virtual server.
2. Specify **HTTP** in the **Protocol** field, and select **HTTP** under **Available Monitors**.

To configure a service for each virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
   - Port—port
4. In Protocol, specify HTTP. Under Available Monitors, select HTTP.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each service

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

Example:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```


To bind a monitor to a service by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services**, double-click a service, and add a monitor.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the command line interface

At the command prompt, type:

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Services**, and configure an HTTP virtual server.
2. Specify **HTTP** in the **Protocol** field.

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
   - IP Address—IP Address
     Note: If the virtual server uses IPv6, select the
     IPv6 check box and enter the address in IPv6 format (for example,
     1000:0000:0000:0000:0005:0600:700a:888b).
   - Port—port
4. Under Protocol, select HTTP.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- `save ns config`
- `show vserver`

Example:

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
```



```
32          TCP Buffering(TCPB): NO
33          HTTP Compression(CMP): NO
34          Idle timeout: Client: 120 sec   Server: 120 sec
35          Client IP: DISABLED
36          Cacheable: NO
37          SC: OFF
38          SP: OFF
39          Down state flush: ENABLED
40
41  1)      Monitor Name: monitor-HTTP-1
42                  State: DOWN     Weight: 1
43                  Probes: 9       Failed [Total: 9 Current: 9]
44                  Last response: Failure - Time out during TCP connection
                        establishment stage
45                  Response Time: 2000.0 millisec
46  2)      Monitor Name: ping
47                  State: UP       Weight: 1
48                  Probes: 3       Failed [Total: 0 Current: 0]
49                  Last response: Success - ICMP echo reply received.
50                  Response Time: 1.275 millisec
51   Done
52  <!--NeedCopy-->
```

To save and verify the configuration by using the configuration utility

1. In the **Details** pane, click **Save**.
2. In the **Save Config** dialog box, click **Yes**.
3. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
4. In the **Details** pane, select the virtual server that you created in step 5.
5. Verify that the settings displayed in the **Details** pane are correct.
6. Navigate to **Traffic Management > Load Balancing > Services**.
7. In the **Details** pane, select the services that you created in step 5.
8. Verify that the settings displayed in the **Details** pane are correct.

Monitoring a Firewall Load Balancing Set up in a Sandwich Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the Citrix ADC appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the Citrix ADC, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

Example:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
         0/s
Two                          *     0          TCP         DOWN      0/s
         0/s
Three                        *  2598          TCP         DOWN      0/s
         0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
         0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
         0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
         0/s
 Done

<!--NeedCopy-->
```


To display virtual server statistics by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers > Statistics**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Services > Statistics**.
2. If you want to display the statistics for only one service, select the service, and click **Statistics**.

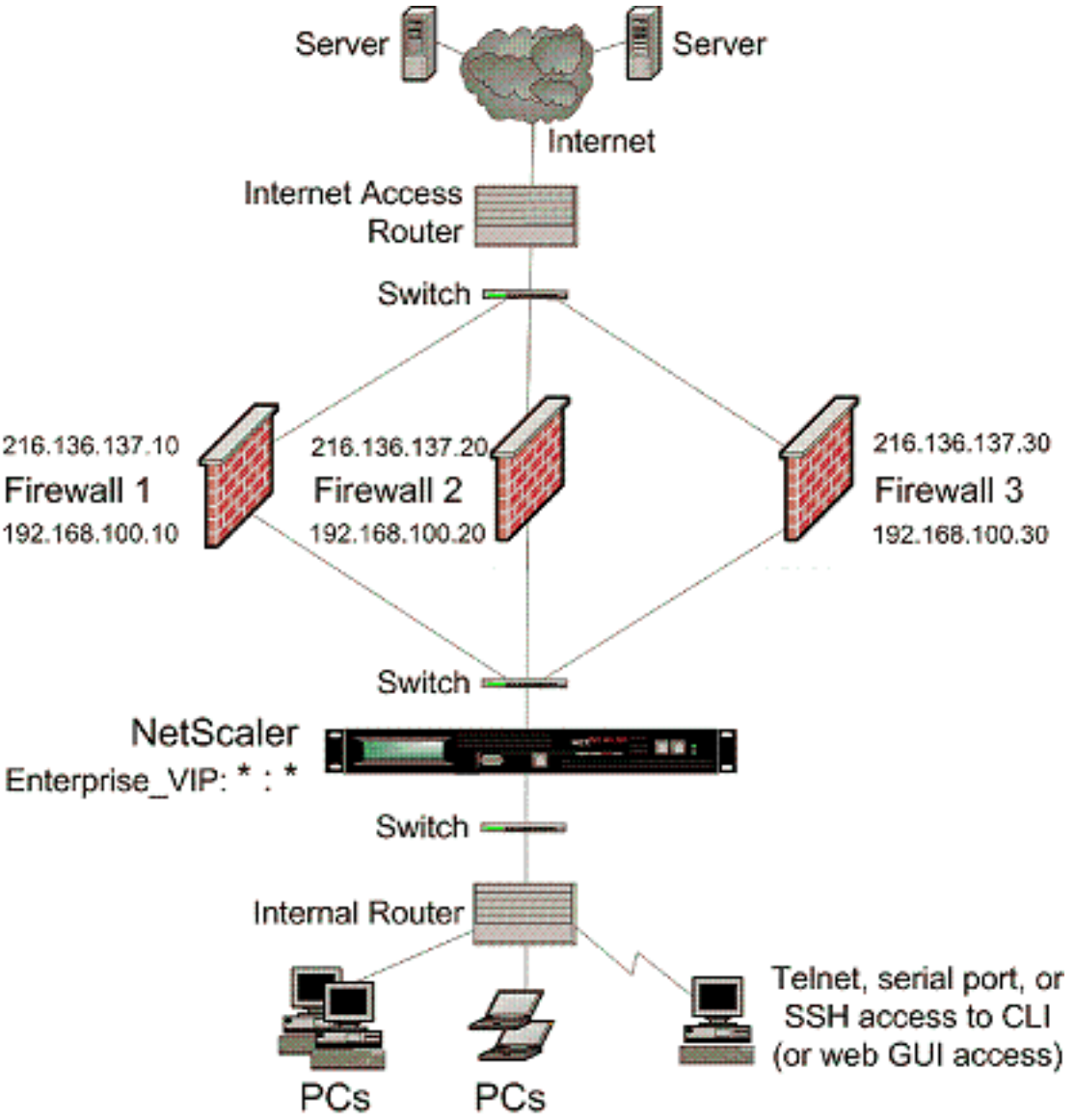
Enterprise Environment

September 14, 2021

In the enterprise setup, the Citrix ADC is placed between the firewalls connecting to the public Internet and the internal private network and handles egress traffic. The Citrix ADC selects the best firewall based on the configured load balancing policy.

The following diagram shows the enterprise firewall load balancing environment

Figure 1. Firewall Load Balancing (Enterprise)



The service type ANY configures the Citrix ADC to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and vserver with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the Citrix ADC in an Enterprise Environment

Perform the following tasks to configure a Citrix ADC in an enterprise environment.

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.

- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```


To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
4. In Protocol, select ANY, and in Port, select *.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the Citrix ADC appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the details pane, click Add.
3. In the Create Monitor dialog box, specify values as shown:
   - Name*
   - Type*—type
   - Destination IP
   - Transparent

   -* A required parameter
4. Click Create, and then click Close. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type:


```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
4. In Protocol, select ANY, and in IP Address and Port, select *.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1), and then click Open.
3. On the Advanced tab, under Redirection Mode, click MAC-Based.
4. Click OK.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

To configure a service for each virtual server by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

To configure a service for each virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
   - Port—port
4. In Protocol, specify HTTP. Under Available Monitors, select HTTP.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each service

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

Example:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and add a monitor.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the command line interface

At the command prompt, type:

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name

- IP Address—IPAddress
  Note: If the virtual server uses IPv6, select the
  IPv6 check box and enter the address in IPv6 format (for example,
  1000:0000:0000:0000:0005:0600:700a:888b).
- Port—port

4. Under Protocol, select HTTP.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Bind HTTP services to the HTTP virtual server

To bind HTTP services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind HTTP services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example:


```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
```



```
                        establishment stage
45                  Response Time: 2000.0 millisec
46 2)      Monitor Name: ping
47                  State: UP       Weight: 1
48                  Probes: 3       Failed [Total: 0 Current: 0]
49                  Last response: Success - ICMP echo reply received.
50                  Response Time: 1.275 millisec
51  Done
52 <!--NeedCopy-->
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in an Enterprise Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the Citrix ADC appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the Citrix ADC appliance, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

Example:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
        0/s
Two                          *     0          TCP         DOWN      0/s
        0/s
Three                        *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done


<!--NeedCopy-->
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

Updated: 2013-08-28

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

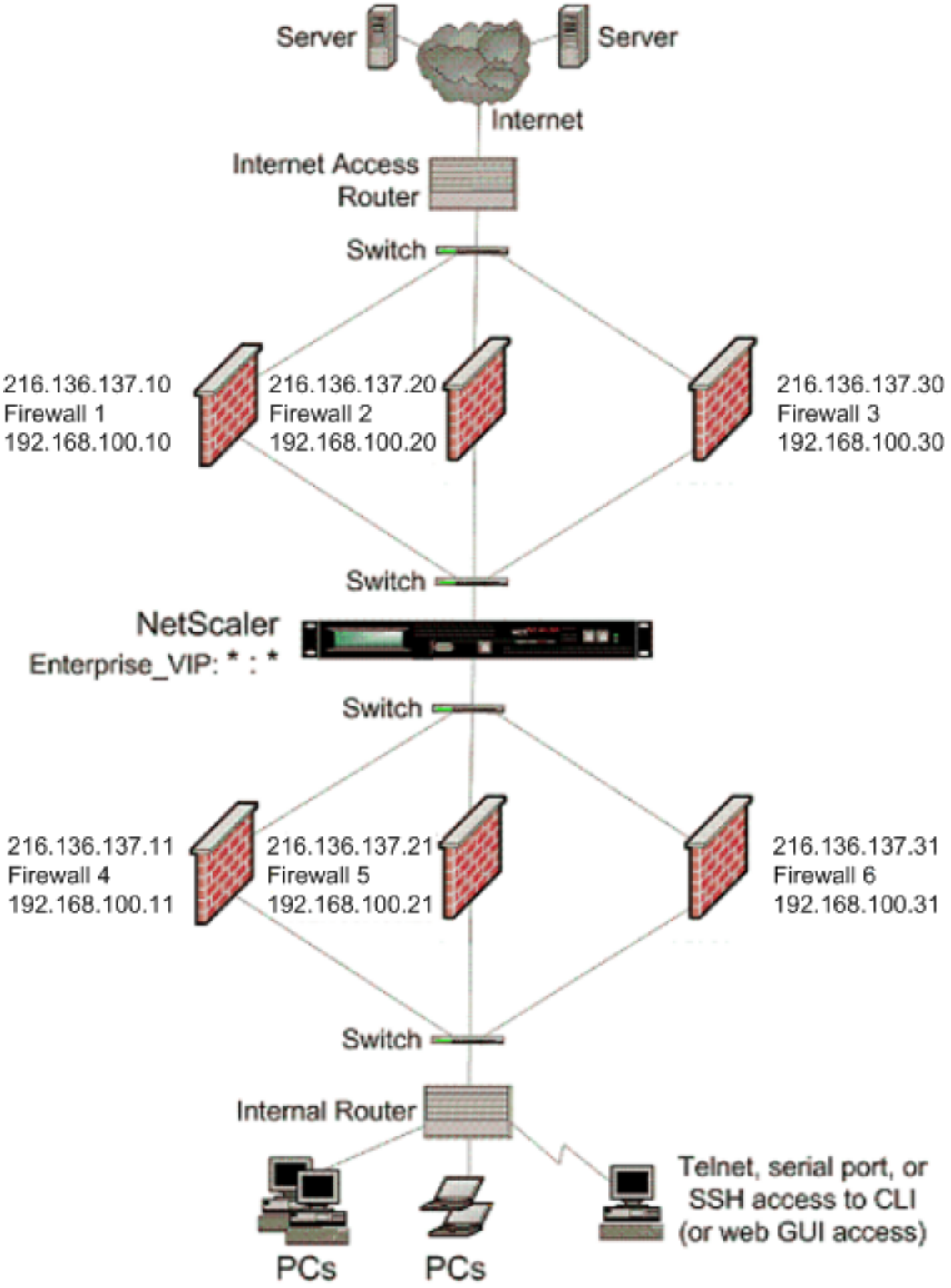
Multiple-Firewall Environment

September 14, 2021

In a multiple-firewall environment, the Citrix ADC appliance is placed between two sets of firewalls, the external set connecting to the public Internet, and the internal set connecting to the internal private network. The external set typically handles the egress traffic. These firewalls mainly implement access control lists to allow or deny access to external resources. The internal set typically handles the ingress traffic. These firewalls implement security to safeguard the intranet from malicious attacks apart from load-balancing the ingress traffic. The multiple-firewall environment allows you to load-balance traffic coming from another firewall. By default, the traffic coming from a firewall is not load balanced on the other firewall across a Citrix ADC appliance. Having firewall load balancing enabled on both the sides of Citrix ADC improves the traffic flow in both the egress and ingress direction and ensures faster processing of the traffic.

The following figure shows a multiple-firewall load balancing environment

Figure 1. Firewall Load Balancing (multiple-firewall)



With a configuration like the one shown in Figure 1, you can configure the Citrix ADC to load balance the traffic through the an internal firewall even if it is load balanced by an external firewall. For example,

with this feature configured, the traffic coming from the external firewalls (firewalls 1, 2, and 3) is load balanced on the internal firewalls (firewalls 4, 5, and 6) and vice versa.

Firewall load balancing is supported only for MAC mode LB virtual server.

The service type ANY configures the Citrix ADC to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the Citrix ADC in a Multiple-Firewall Environment

To configure a Citrix ADC appliance in a multiple-firewall environment, you have to enable the load balancing feature, configure a virtual server to load balance the egress traffic across the external firewalls, configure a virtual server to load balance the ingress traffic across the internal firewalls, and enable firewall load balancing on the Citrix ADC appliance. To configure a virtual server to load balance traffic across a firewall in the multiple-firewall environment, you need to:

1. Configure a wildcard service for each firewall
2. Configure a monitor for each wildcard service
3. Configure a wildcard virtual server to load balance the traffic sent to the firewalls
4. Configure the virtual server in MAC rewrite mode
5. Bind firewall services to the wildcard virtual server

Enabling the load balancing feature

To configure and implement load balancing entities such as services and virtual servers, you need to enable the load balancing feature on the Citrix ADC device.

To enable load balancing by using the CLI:

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature <featureName>
show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature LoadBalancing
Done
show ns feature
Feature Acronym Status
------- ------- ------
```



```
1) Web Logging WL OFF
2) Surge Protection SP ON
3) Load Balancing LB ON
.
.
.
24) NetScaler Push push OFF
Done
<!--NeedCopy-->
```

To enable load balancing by using the GUI:

1. In the navigation pane, expand System, and then click Settings.
2. In the Settings pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click Ok.

Configuring a wildcard service for each firewall

To accept traffic from all the protocols, you need to configure wildcard service for each firewall by specifying support for all the protocols and ports.

To configure a wildcard service for each firewall by using the CLI:

At the command prompt, type the following command to configure support for all the protocols and ports:

```
add service <name>@ <serverName> <serviceType> <port_number>
<!--NeedCopy-->
```

Example:

```
add service fw-svc1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Services dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName

   -* A required parameter

4. In Protocol, select Any and in Port, select *.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configuring a monitor for each service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the Citrix ADC appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the CLI:

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

The Citrix ADC appliance learns the server L2 parameters from the monitor that is bound to the service. For UDP-ECV monitors, configure a receive string to enable the appliance to learn the L2 parameters of the server. If the receive string is not configured and the server does not respond, then the appliance does not learn the L2 parameters but the service is set to UP. The traffic for this service is blackholed.

To configure a receive string by using the CLI:

At the command prompt, type the following command:


```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )] [-send <string>] [-recv <string>]
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-udp-1 udp-ecv -destip 10.10.10.11 -transparent
    YES  -send "test message"  -recv "site_is_up"
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the details pane, click Add.
3. In the Create Monitor dialog box, specify values for the following parameters as shown:
   - Name*
   - Type*—type
   - Destination IP
   - Transparent

   -* A required parameter
4. Click Create, and then click Close. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring a virtual server to load balance the traffic sent to the firewalls

To load balance any kind of traffic, you need to configure a wildcard virtual server specifying the protocol and port as any value.

To configure a virtual server to load balance the traffic sent to the firewalls by using the CLI:

At the command prompt, type the following command:

```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a virtual server to load balance the traffic sent to the firewalls by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In Protocol, select Any, and in IP Address and Port, select *.
4. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configuring the virtual server to MAC rewrite mode

To configure the virtual server to use MAC address for forwarding the incoming traffic, you need to enable the MAC rewrite mode.

To configure the virtual server in MAC rewrite mode by using the CLI:

At the command prompt, type the following command:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB1), and then click Open.
3. On the Advanced tab, under the Redirection Mode mode, click Open.
4. Click Ok.

Binding firewall services to the virtual server

To access a service on Citrix ADC appliance, you need to bind it to a wildcard virtual server.

To bind firewall services to the virtual server by using the CLI:

At the command prompt, type the following command:

```
bind lb vserver <name>@ <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```


To bind firewall services to the virtual server by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Services tab, select the Active check box next to the service that you want to bind to the virtual server(for example, Service-HTTP-1).
4. Click Ok.

Configuring the multiple-firewall load balancing on the Citrix ADC appliance

To load balance traffic on both the sides of a Citrix ADC using firewall load balancing, you need to enable mulitpl-firewall load balancing by using the vServerSpecificMac parameter.

To configure multiple-firewall load balancing by using the CLI:

At the command prompt, type the following command:

```
set lb parameter -vServerSpecificMac <status>
<!--NeedCopy-->
```

Example:

```
set lb parameter -vServerSpecificMac ENABLED
<!--NeedCopy-->
```

To configure multiple-firewall load balancing by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Configure Load Balancing parameters).
3. In the Set Load Balancing Parameters dialog box, select the Virtual Server Specific MAC check box.
4. Click Ok.

Saving and Verifying the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the CLI:

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example:

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

```



```
1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

To save and verify the configuration by using the GUI:

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in a Multiple-Firewall Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the Citrix ADC appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received

- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the Citrix ADC appliance, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

Example:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
        0/s
Two                          *     0          TCP         DOWN      0/s
        0/s
Three                        *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done


<!--NeedCopy-->
```

To display virtual server statistics by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the CLI:

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

Global Server Load Balancing

September 14, 2021

> **Notes:**
>
> - From release 13.0 build 41.x, global server load balancing (GSLB) deployments using the Citrix ADC appliance are fully compliant with DNS flag day 2019.
> - The GSLB feature is included with the Citrix ADC Advance and Premium edition licenses. The Citrix ADC option license is supported with the Standard edition.

Citrix ADC appliances configured for GSLB provide disaster recovery and ensure continuous availability of applications by protecting against points of failure in a WAN. GSLB balances the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers if there is an outage.

In a typical configuration, a local DNS server sends client requests to a GSLB virtual server, to which are bound GSLB services. A GSLB service identifies a load balancing or content switching virtual server, which can be at the local site or a remote site. If the GSLB virtual server selects a load balancing or content switching virtual server at a remote site, it sends the virtual server's IP address to the DNS server. The DNS server sends it to the client. The client then resends the request to the new virtual server at the new IP.

The GSLB entities that you must configure are the GSLB sites, the GSLB services, the GSLB virtual servers, load balancing or content switching virtual servers, and authoritative DNS (ADNS) services. You must also configure MEP. You can also configure DNS views to expose different parts of your network to clients accessing the network from different locations.

> **Note:**
>
> To take full advantage of GSLB features, use ADC appliances for load balancing or content switching at each data center, so that your GSLB configuration can use the proprietary MEP to exchange site metrics.

How GSLB works

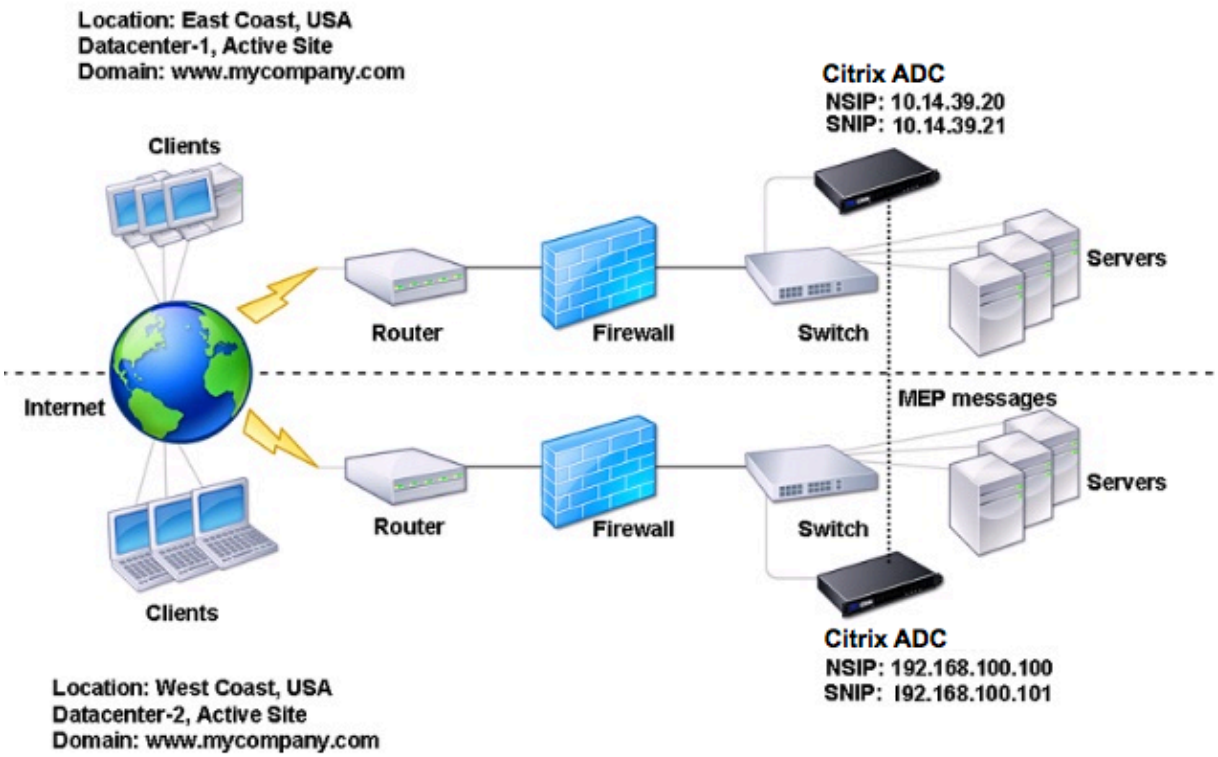
With ordinary DNS, when a client sends a domain name system (DNS) request, it receives a list of IP addresses of the domain or service. Generally, the client chooses the first IP address in the list and initiates a connection with that server. The DNS server uses a technique called DNS round robin to rotate through the IPs on the list. It sends the first IP address to the end of the list and promotes the others after it responds to each DNS request. This technique ensures equal distribution of the load, but it does not support disaster recovery, load balancing based on load or proximity of servers, or persistency.

When you configure GSLB on ADC appliances and enable MEP, the DNS infrastructure is used to connect the client to the data center that best meets the set criteria. The criteria can designate the following:

- Least loaded data center
- Closest data center
- Data center that responds most quickly to requests from the client's location
- A combination of those metrics and SNMP metrics.

An appliance tracks the location, performance, load, and availability of each data center. It uses these factors to select the data center to send the client request.

The following figure illustrates a basic GSLB topology.



A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include GSLB sites, GSLB services, GSLB service groups, GSLB virtual servers, load balancing servers, content switching servers, and ADNS services.

GSLB deployment types

September 14, 2021

Citrix ADC appliances configured for global server load balancing (GSLB) provide for disaster recovery and ensure continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in the event of an outage.

The following are some of the typical GSLB deployment types:

- Active-active site deployment
- Active-passive site deployment
- Parent-child topology deployment

Active-active site deployment

September 14, 2021

An active-active site consists of multiple active data centers. Client requests are load balanced across active data centers. This deployment type can be used when you have a need for global distribution of traffic in a distributed environment.

All the sites in an active-active deployment are active, and all the services for a particular application/domain are bound to the same GSLB virtual server. Sites exchange metrics through the Metrics Exchange Protocol (MEP). Site metrics exchanged between the sites include the status of each load balancing and content switching virtual server, current number of connections, current packet rate, and current bandwidth usage. The Citrix ADC appliance needs this information to perform load balancing across the sites.

An active-active deployment can include a maximum of 32 GSLB sites, because MEP cannot synchronize more than 32 sites. No backup sites are configured in this deployment type.

The Citrix ADC appliance sends client requests to the appropriate GSLB site as determined by the GSLB method specified in the GSLB configuration.

For an active-active deployment, you can configure the following GSLB methods.

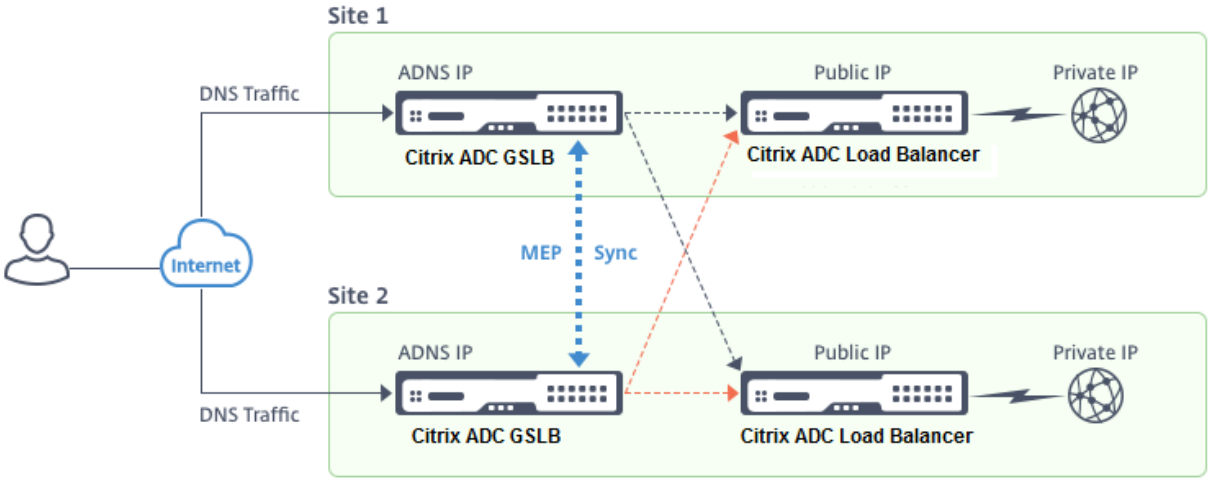
- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

> Note:
>
> - If MEP is disabled, the following GSLB methods default to the Round Robin method.
>   - RTT
>   - Least Connections
>   - Least Bandwidth
>   - Least Packets
>   - Least Response Time
> - In the static proximity GLSB method, the appliance sends the request to the IP address of the site that best matches the proximity criteria.

- In the Round Trip Time method, the dynamic round trip time (RTT) values are to select the IP address of the best performing site. RTT is a measure of the delay in the network between the client's local DNS server and a data resource.

GSLB active-active data center topology

In the diagram, Site 1 and Site 2 are active GLSB sites.



When the client sends a DNS request, it lands in one of the active sites.

If Site 1 receives the client request, the GSLB virtual server in Site 1 selects a load balancing or content switching virtual server and sends the virtual server's IP address to the DNS server, which sends it to the client. The client then resends the request to the new virtual server at the new IP address.

As both sites are active, the GSLB algorithm evaluates the services at both sites when making a selection as determined by the configured GSLB method.

Active-passive site deployment

September 14, 2021

An active-passive site consists of an active and a passive data center. This deployment type is ideal for disaster recovery.

In this type of deployment, some of the sites (remote sites) are reserved only for disaster recovery. These sites do not participate in any decision making until all the active sites are DOWN. A passive site does not become operational unless a disaster event triggers a failover.

Once you have configured the primary data center, replicate the configuration for the backup data center and designate it as the passive GSLB site by designating a GSLB virtual server at that site as the backup virtual server.

An active-passive deployment can include a maximum of 32 GSLB sites, because MEP cannot synchronize more than 32 sites.

For an active-passive deployment, you can configure the following GSLB methods.

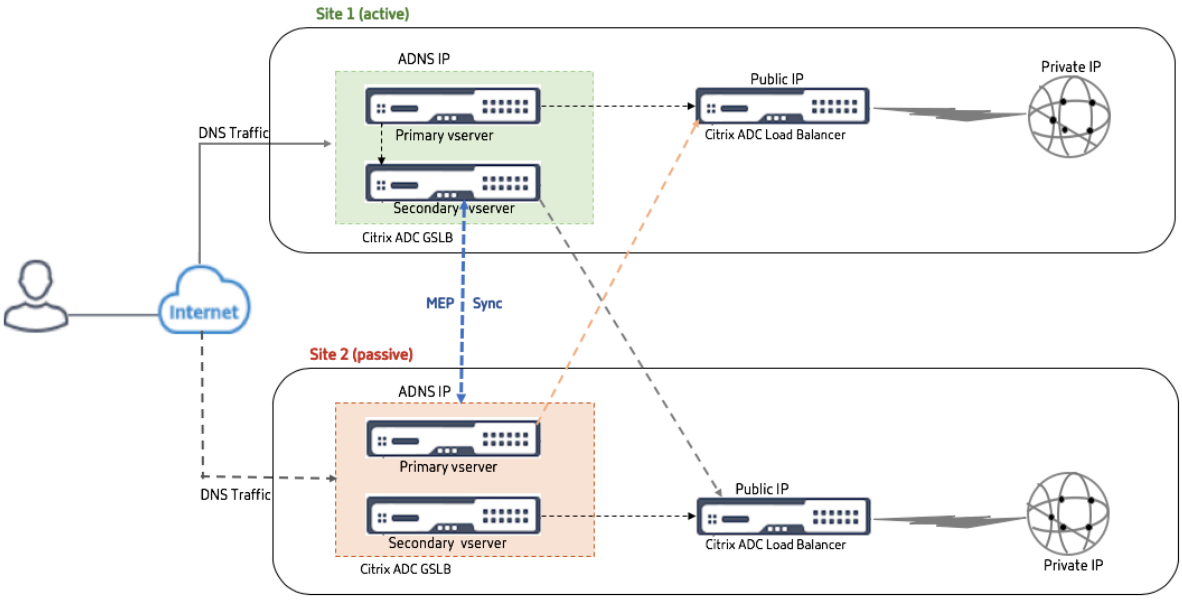
- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

Note:

- If MEP is disabled, the following algorithm methods default to Round Robin.
  - RTT
  - Least Connections
  - Least Bandwidth
  - Least Packets
  - Least Response Time
- In the static proximity GLSB method, the appliance sends the request to the IP address of the site that best matches the proximity criteria.
- In the Round Trip Time method, the dynamic round trip time (RTT) values are to select the IP address of the best performing site. RTT is a measure of the delay in the network between the client's local DNS server and a data resource.

GSLB active-passive datacenter topology

In the diagram, Site 1 is an active site and Site 2 is a passive site, which has the same configuration as that of Site 1.



If Site 1 goes DOWN, Site 2 becomes operational.

When the client sends a DNS request, the request can land in any of the sites. However, the services are selected only from the active site (Site1) as long as it is UP.

Services from the passive site (Site 2) are selected only if the active site (Site 1) is DOWN.

Parent-child topology deployment using the MEP protocol

September 14, 2021

Citrix ADC GSLB provides global server load balancing and disaster recovery by creating mesh connections between all the involved sites and making intelligent load balancing decisions. Each site communicates with the others to exchange server and network metrics through Metric Exchange Protocol (MEP), at regular intervals. However, with the increase in number of peer sites, the volume of MEP traffic increases exponentially because of the mesh topology. To overcome this, you can use a parent-child topology. The parent-child topology also supports larger deployments. In addition to the 32 parent sites, you can configure 1024 child sites.

The GSLB parent-child topology is a two-level hierarchical design with the following characteristics:

- At the top level are parent sites, which have peer relationships with other parents.
- Each parent can have multiple child sites.
- Each parent site exchanges health information with its child sites and with other parent sites.
- A child site communicates only with its parent site.

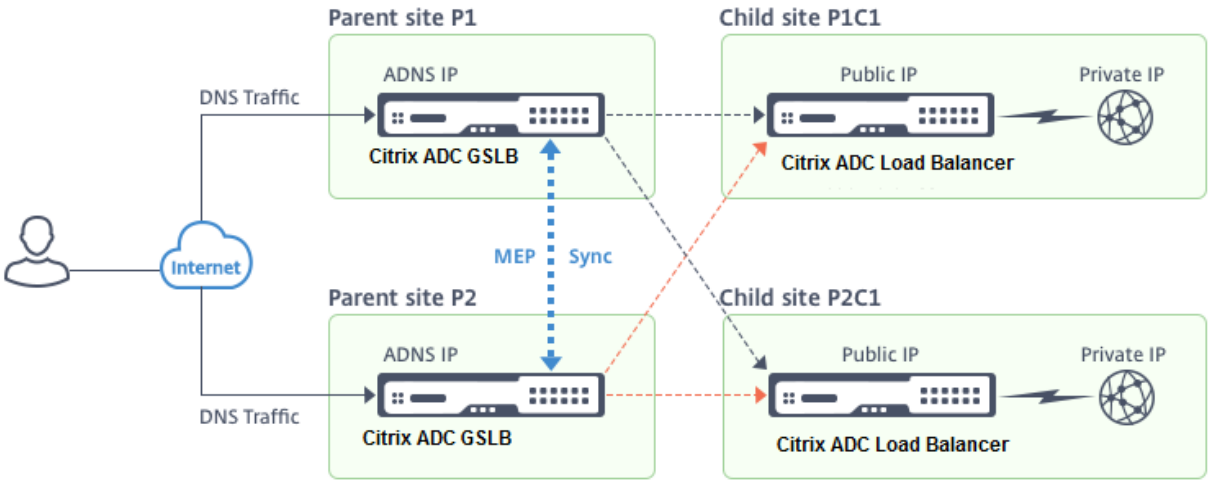
- In a parent-child relationship for GSLB, only the parent site responds to ADNS queries. The child sites act as normal load balancing sites.
- Configure an ADNS service or DNS load balancing virtual server only in the parent site.
- A parent site can have a normal GSLB configuration, that is, services from local and all remote sites, but a child site can have local services only. Also, only the parent sites have GSLB virtual servers configured.

> **Note**
>
> - In a parent-child topology, the exchange of site metrics is initiated from the lower of two IP addresses. However, from Citrix ADC release 11.1 build 51.x, the parent sites initiate connections to the child sites, and not the opposite way. Because the parent sites have information about all the child sites in the GSLB setup.

- In a parent-parent connection, the exchange of site metrics is still initiated from the lower IP of two IP addresses.
- In a parent-child topology, GSLB services are not always required to be configured on a child site. However, if you have more configuration such as client authentication, client IP address insertion, or other SSL-specific requirement, you must add an explicit GSLB service on the child site and configure it accordingly.
- In a parent-child topology, the parent site and the child site can be on differentCitrix ADC software versions. However, to use the GSLB automaticConfigSync option to synchronize the configuration across the parent sites, all parent sites must be on the sameCitrix ADC software versions. If you are not using the automaticConfigSync option, then the parent site and the child site can be on differentCitrix ADC software versions but make sure that you are not using any of the new features in the latest release. This is also applicable, in general, to two Citrix ADC nodes participating in GSLB.

Basic parent-child topology

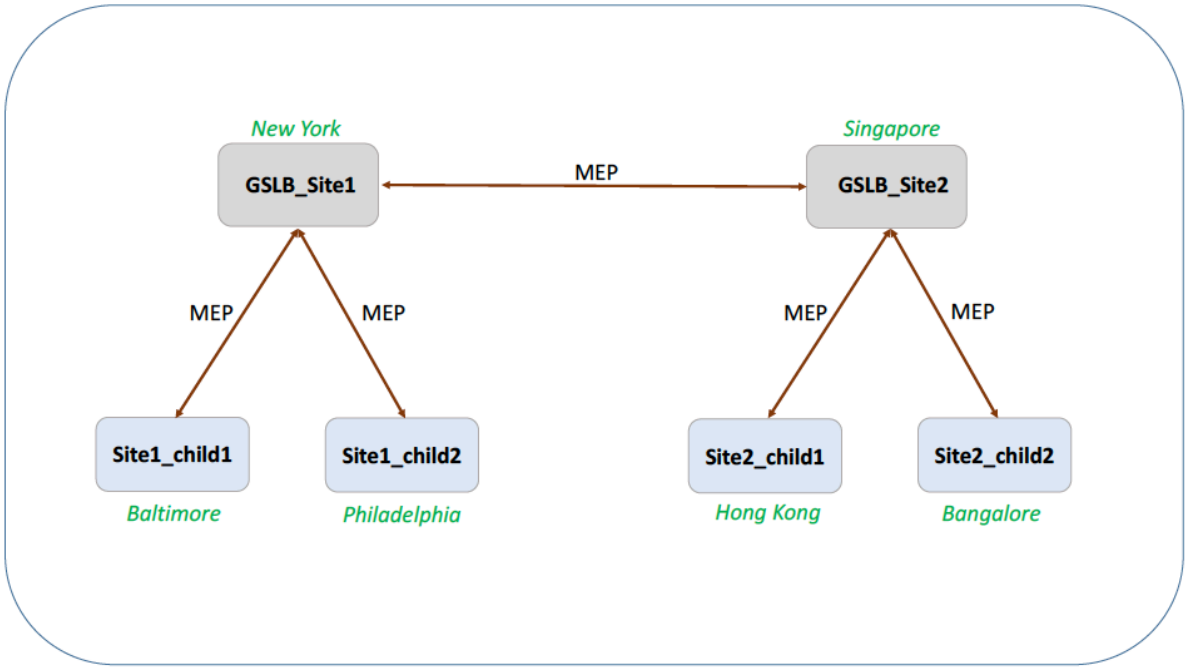
In the diagram, SiteP1 and SiteP2 are parent sites in a peer relationship. Site P1C1 and P2C1 are the child sites of P1 and P2 respectively.



Setting up a parent-child configuration for GSLB

If you have a firewall configured at a GSLB site, make sure that port 3011 is open.

The following diagram displays a sample parent-child configuration.



- The configuration of a child site includes the child site and its parent site, but no other parent or child sites.
- Network metrics, such as RTT and persistence session information, are synchronized only across the parent sites. Therefore, parameters such as nwMetricExchange and sessionExchange are disabled by default on all the child sites.
- To verify correct parent-child configuration, check the states of all the GSLB services bound to the parent sites.

To set up a parent-child configuration for GSLB by using the CLI:

1. On each parent site, enter the following commands:

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
   ipv6_addr|*>]

add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
   ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

   Example:

```
add gslb site GSLB_Site1 10.1.1.1 - publicIP 10.1.1.1

add gslb site Site1_child1 1 10.1.1.2 -publicIP 10.1.1.2 -
   parentSite GSLB_Site1
<!--NeedCopy-->
```

2. On each child site, enter the following commands:

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
   ipv6_addr|*>]

add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
   ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

   Example:

```
add gslb site GSLB_Site1 10.1.1.1 - publicIP 10.1.1.1

add gslb site Site1_child1 1 10.1.1.2 -publicIP 10.1.1.2 -
   parentSite GSLB_Site1
<!--NeedCopy-->
```

For a complete example of a parent-child configuration, using the command line interface, see Example of a Complete Parent-Child Configuration, Using the CLI

> **Note**
>
> If the load balancing virtual server IP address is a private IP address and the public IP address is different from this IP address, you need to configure a GSLB service for the local load balancing virtual server on the child site. This is required for statistics collection between the parent and the child site.

> On the child site, at the command prompt, type:
>
> ```
> add gslb service <name> <private IP/lb vserver IP> http 80 -sitename <
> childsite name> -publicip <public IP of LB vserver>
> ```
>
> **Example:**
>
> add gslb service Service-GSLB 192.168.1.3 http 80 -GSLB_Site11 site 11_lb1 172.16.1.1
>
> Where 192.168.1.3 is a private IP address of the load balancing virtual server and 172.16.1.1 is a public IP address of the load balancing virtual server.

Backing up a parent site

Note: This feature was introduced in Citrix ADC release 11.1 build 51.x. To use the backup parent site topology, make sure that the parent site and the child sites are on Citrix ADC 11.1 build 51.x and later.

The backup parent site topology is useful in scenarios wherein many child sites are associated with a parent site. If this parent site goes DOWN, all of its child sites become unavailable. To prevent this, you can now configure a backup parent site to which the child sites can connect if the original parent site is DOWN. The parent site sends the backup parent list to the child sites through the MEP messages.

When a parent site is DOWN, the other parent sites in the GSLB get to know that a particular parent site is DOWN through MEP because MEP to that parent site is DOWN. The other parent sites in the GSLB setup look up the backup chain of the peer parent. The parent site with the highest preference adopts the child sites of the parent that went DOWN. The new parent then initiates a connection with the child site. A child site can accept or reject the connection after evaluating its existing connections and the information in the backup list. It takes a few seconds for the backup parent to adopt the child sites.
When the original parent site is back UP, it tries to establish connections with its child sites that have migrated to a different parent. When a connection attempt is successful, the child site is reassigned to its original parent site.

Note:

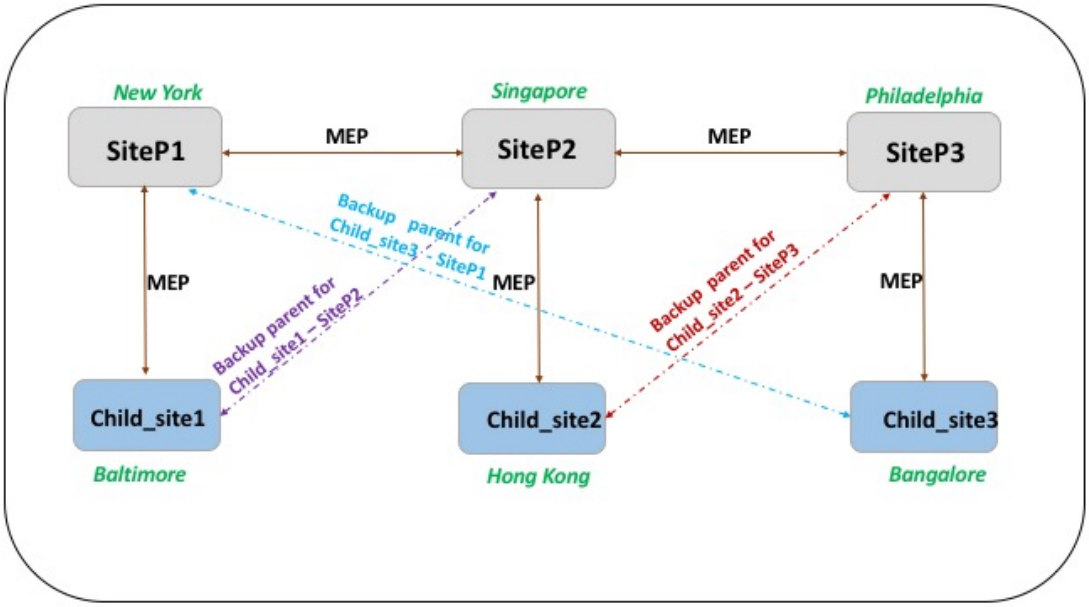
- Only parent sites can be configured as backups, and this configuration can only be done at the parent site.
- All child sites use the backup parent set.
- Synchronization is done only on the parent sites. GSLB child sites' configuration is not affected by synchronization. This is because the parent site and the child site configurations are not identical. The child sites configuration consists only of its own and its parent site's details. Also, GSLB services are not always required to be configured in the child sites.

Consider the configuration shown in the following figure, in which:

- SiteP1, SiteP2, and SiteP3 are the parent sites.

- Child_site1, Child_site2, and Child_site3 are the child sites of SiteP1, SiteP2, and SiteP3 respectively.
- Backup parent sites;
  - SiteP1 backup parents - SiteP2 (higher preference) and SiteP3
  - SiteP2 backup parents – SiteP3 (higher preference) and SiteP1
  - SiteP3 backup parents – SiteP1 (higher preference) and SiteP2

Note: For illustration purposes, the figure shows only one backup parent for each parent site.



The following list summarizes the behavior of the parent and child sites under various scenarios:

- Scenario 1: SiteP1 goes DOWN.
  - SiteP2 and SiteP3 detect that SiteP1's MEP connection is DOWN. SiteP2 is higher in the preference list of backup parents for SiteP1, so it tries to initiate a connection to Child_site1. SiteP3 assumes that Child_site1 is now the child site of parent SiteP2.
  - SiteP2 sends Child_Site1 the list of SiteP1's backup parents (SiteP2 and SiteP3) to Child_site1. Child_site1 uses the list to decide whether to accept or reject the connection from SiteP2. It accepts the connection and becomes a child of SiteP2.
  - When SiteP1 is back UP, it sends Child_site1 a connection request. The new request takes precedence and Child_site 1 migrates to SiteP1.
- Scenario 2: Only the MEP connection between SiteP1 and SiteP2 has gone DOWN. Child_site1 rejects SiteP2's connection request, because its parent, SiteP1, is still UP.
- Scenario 3: SiteP3 and Child_Site1 detect that SiteP1 is DOWN, and the MEP connection between SiteP3 and SiteP2 is also DOWN. However, SiteP2 detects that SiteP1 is UP, and the MEP

connection between SiteP1 and SiteP2 is UP.

- SiteP2 does not take any action.
- SiteP3 checks SiteP1's backup list and finds that SiteP2 has a higher preference than SiteP3. But SiteP2 is DOWN, so SiteP3 tries to establish a connection with Child_site1. Child_site1 has detected that SiteP1 is DOWN, so it accepts SiteP3's connection request.
- Now the connection between SiteP1 and SiteP2 goes DOWN. SiteP2 checks SiteP1's backup list and finds itself as the most preferred backup, so it tries to connect to Child_site1. Child_site1 evaluates the new connection request based on SiteP1's list and finds SiteP2 as the most preferred backup, so it migrates to SiteP2 from SiteP3.

To configure a backup parent site by using the command line interface

At the command prompt type:

```
set gslb site <sitename> -backupParentlist <bkp_site1> <bkp_site2> … <
   bkp_site5>
<!--NeedCopy-->
```

<sitename> is the current parent site.

Example:

For the parent site (SiteP1), the sites (SiteP2 and SiteP3) are configured as the backup parent sites.

```
set gslb site SiteP1 -backupParentlist SiteP2 SiteP3
<!--NeedCopy-->
```

Note:

- You cannot add a new site as a backup parent. You must first add all the sites, and then configure the site as a backup parent.
- To remove a backup parent, you must use the unset command, which unsets all the sites that were previously configured as backup parent sites.

To configure a backup parent site by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Sites**.
2. Add a new site or select an existing site.
3. Choose the **Backup Parent Sites** option box while creating or configuring the GSLB site.

GSLB configuration entities

September 14, 2021

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include the following:

- GSLB Sites
- GSLB Services
- GSLB Virtual Servers
- Load Balancing or Content Switching Virtual Servers
- ADNS Services
- DNS VIPs

GSLB Sites

A typical GSLB setup consists of data centers, each of which has various network appliances that may or may not be Citrix ADC appliances. The data centers are called GSLB sites. Each GSLB site is managed by a Citrix ADC appliance that is local to that site. Each of these appliances treats its own site as the local site and all other sites, managed by other appliances, as remote sites.

If the appliance that manages a site is the only Citrix ADC appliance in that data center, the GSLB site hosted on that appliance acts as a bookkeeping placeholder for auditing purposes, because no metrics can be collected. Typically, this happens when the appliance is used only for GSLB, and other products in the data center are used for load balancing or content switching.

Relationships among GSLB Sites

The concept of sites is central to Citrix ADC GSLB implementations. Unless otherwise specified, sites form a peer relationship among themselves. This relationship is used first to exchange health information and then to distribute load as determined by the selected algorithm. In many situations, however, a peer relationship among all GSLB sites is not desirable. Reasons for not having an all-peer implementation could be;

- To clearly separate GSLB sites. For example, to separate sites that participate in resolving DNS queries from the traffic management sites.
- To reduce the volume of Metric Exchange Protocol (MEP) traffic, which increases exponentially with an increasing number of peer sites.

These goals can be achieved by using parent and child GSLB sites.

GSLB Services

A GSLB service is usually a representation of a load balancing or content switching virtual server, although it can represent any type of virtual server. The GSLB service identifies the virtual server's IP address, port number, and service type. GSLB services are bound to GSLB virtual servers on the Citrix ADC appliances managing the GSLB sites. A GSLB service bound to a GSLB virtual server in the same data center is local to the GSLB virtual server. A GSLB service bound to a GSLB virtual server in a different data center is remote from that GSLB virtual server.

> **Note**
>
> Sites and services are inherently linked to indicate proximity between the two. That is, all services must belong to a site, and are assumed to be in the same location as the GSLB site for proximity purposes. Likewise, services and virtual servers are linked, so that the logic is linked to the resources that are available.

GSLB Virtual Servers

A GSLB virtual server has one or more GSLB services bound to it, and load balances traffic among those services. It evaluates the configured GSLB methods (algorithms) to select the appropriate service to which to send a client request. Because the GSLB services can represent either local or remote servers, selecting the optimal GSLB service for a request has the effect of selecting the data center that should serve the client request.

The domain for which global server load balancing is configured must be bound to the GSLB virtual server, because one or more services bound to the virtual server will serve requests made for that domain.

Unlike other virtual servers configured on a Citrix ADC appliance, a GSLB virtual server does not have its own virtual IP address (VIP).

Load Balancing or Content Switching Virtual Servers

A load balancing or content switching virtual server represents one or many physical servers on the local network. Clients send their requests to the load balancing or content switching virtual server's virtual IP (VIP) address, and the virtual server balances the load across the physical servers. After a GSLB virtual server selects a GSLB service representing either a local or a remote load balancing or content switching virtual server, the client sends the request to that virtual server's VIP address.

For more information about load balancing or content switching virtual servers and services, see Load Balancing or Content Switching.

ADNS Services

An ADNS service is a special kind of service that responds only to DNS requests for domains for which the Citrix ADC appliance is authoritative. When an ADNS service is configured, the appliance owns that IP address and advertises it. Upon reception of a DNS request by an ADNS service, the appliance checks for a GSLB virtual server bound to that domain. If a GSLB virtual server is bound to the domain, it is queried for the best IP address to which to send the DNS response.

DNS VIPs

A DNS virtual IP is a virtual IP (VIP) address that represents a load balancing DNS virtual server on the Citrix ADC appliance. DNS requests for domains for which the Citrix ADC appliance is authoritative can be sent to a DNS VIP.

GSLB methods

September 14, 2021

Unlike traditional DNS servers that simply respond with the IP addresses of the configured servers, a Citrix ADC appliance configured for GSLB responds with the IP addresses of the services, as determined by the configured GSLB method. By default, the GSLB virtual server is set to the least connection method. If all GSLB services are down, the appliance responds with the IP addresses of all the configured GSLB services.

GSLB methods are algorithms that the GSLB virtual server uses to select the best-performing GSLB service. After the host name in the Web address is resolved, the client sends traffic directly to the resolved service IP address.

The Citrix ADC appliance provides the following GSLB methods:

- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

For GSLB methods to work with a remote site, either MEP must be enabled or explicit monitors must be bound to the remote services. If MEP is disabled, RTT, Least Connections, Least Bandwidth, Least

Packets and Least Response Time methods default to Round Robin.

The Static Proximity and RTT load balancing methods are specific to GSLB.

Specifying a GSLB method other than static proximity or dynamic RTT

For information about the Round Robin, Least Connections, Least Response Time, Least Bandwidth, Least Packets, Source IP Hash, or Custom Load method, see Load Balancing.

To change the GSLB method by using the CLI

At the command prompt, type:

```
set gslb vserver <name> -lbMethod GSLBMethod
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
<!--NeedCopy-->
```

To change the GSLB method by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. In the details pane, select a GSLB virtual server and click **Open**.
3. In the Configure GSLB Virtual Server dialog box, on the Method and Persistence tab, under Method, select a method from the Choose Method list.
4. Click **OK**, and verify that the method you selected appears under Details at the bottom of the screen.

GSLB algorithms

September 14, 2021

The following algorithm are supported for GSLB.

- **Round Robin**: When a GSLB virtual server is configured to use the round robin method, it continuously rotates a list of the services that are bound to it. When the virtual server receives a request, it assigns the connection to the first service in the list, and then moves that service to the bottom of the list.

- **Least Response Time**: When the GSLB virtual server is configured to use the least response time method, it selects the service with the lowest value. Where, lowest value = current active connections X average response time.

  You can configure this method for HTTP and Secure Sockets Layer (SSL) services only. The response time (also called Time to First Byte, or TTFB) is the time interval between sending a request packet to a service and receiving the first response packet from the service. The NetScaler appliance uses response code 200 to calculate TTFB.

- **Least Connections**: When a GSLB virtual server is configured to use the least connection GSLB algorithm (or method), it selects the service with the fewest active connections. This is the default method, because, in most circumstances, it provides the best performance.

- **Least Bandwidth**: A GSLB virtual server configured to use the least bandwidth method selects the service that is currently serving the least amount of traffic, measured in megabits per second (Mbps).

- **Least Packets**: A GSLB virtual server configured to use the least packets method selects the service that has received the fewest packets in the last 14 seconds.

- **Source IP Hash**: A GSLB virtual server configured to use the source IP hash method uses the hashed value of the client IPv4 or IPv6 address to select a service. To direct all requests from source IP addresses that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

- **Custom Load**: Custom load balancing is performed on server parameters such as CPU usage, memory, and response time. When using the custom load method, the Citrix ADC appliance usually selects a service that is not handling any active transactions. If all of the services in the GSLB setup are handling active transactions, the appliance selects the service with the smallest load. A special type of monitor, known as a load monitor, calculates the load on each service in the network. The load monitors do not mark the state of a service, but they do take services out of the GSLB decision when those services are not UP.

For more details, see Load Balancing.

Static proximity

September 14, 2021

The static proximity method for GSLB uses an IP-address based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The Citrix ADC appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the Citrix ADC appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

For the static proximity method to work, you must either configure the Citrix ADC appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the GSLB method.

For details about configuring static proximity, see Configuring Static Proximity.

Dynamic round trip time method

September 14, 2021

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the Citrix ADC appliance probes the client's local DNS server and gathers RTT metric information. The appliance then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value.

When a client's DNS request for a domain comes to the Citrix ADC appliance configured as the authoritative DNS for that domain, the appliance uses the RTT value to select the IP address of the best performing site to send it as a response to the DNS request.

The Citrix ADC appliance uses different mechanisms, such as ICMP echo request or reply (PING), UDP, and TCP to gather the RTT metrics for connections between the local DNS server and participating sites. The appliance first sends a ping probe to determine the RTT. If the ping probe fails, a DNS UDP probe is used. If that probe also fails, the appliance uses a DNS TCP probe.

These mechanisms are represented on the Citrix ADC appliance as Load Balancing Monitors and are easily identified due to their use of the "ldns" prefix. The three monitors, in their default order, are:

- `ldns-ping`
- `ldns-dns`
- `ldns-tcp`

These monitors are built into the appliance and are set to safe defaults. But they are customizable like any other monitor on the appliance.

You can change the default order by setting it explicitly as a GSLB parameter. For example, to set the order to be the DNS UDP query followed by the PING and then TCP, type the following command:

```
1 set gslb parameter -ldnsprobeOrder DNS PING TCP
```



```
<!--NeedCopy-->
```

Unless they have been customized, the Citrix ADC appliance performs UDP and TCP probing on port 53, however unlike regular load balancing monitors the probes need not be successful to provide valid RTT information. ICMP port unavailable messages, TCP Resets and DNS error responses, which would usually constitute a failure are all acceptable for calculating the RTT value.

Once the RTT data has been compiled, the appliance uses the proprietary metrics exchange protocol (MEP) to exchange RTT values between participating sites. After calculating RTT metrics, the appliance sorts the RTT values to identify the data center with the best (smallest) RTT metric."

If RTT information is not available (for example, when a client's local DNS server accesses the site for the first time), the Citrix ADC appliance selects a site by using the round robin method and directs the client to the site.

To configure the dynamic method, you configure the site's GSLB virtual server for dynamic RTT. You can also set the interval at which local DNS servers are probed to a value other than the default.

Configure a GSLB virtual server for dynamic RTT

To configure a GSLB virtual server for dynamic RTT, you specify the RTT load balancing method.

The Citrix ADC appliance regularly validates the timing information for a given local server. If a change in latency exceeds the configured tolerance factor, the appliance updates its database with the new timing information and sends the new value to other GSLB sites by performing a MEP exchange. The default tolerance factor is 5 milliseconds (ms).

The RTT tolerance factor must be the same throughout the GSLB domain. If you change it for a site, you must configure identical RTT tolerance factors on all Citrix ADC appliances deployed in the GSLB domain.

To configure a GSLB virtual server for dynamic RTT by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -lbMethod RTT -tolerance <value>
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
<!--NeedCopy-->
```


To configure a GSLB virtual server for dynamic RTT by using the configuration utility

Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the virtual server.

Set the probing interval of local DNS servers

The Citrix ADC appliance uses different mechanisms, such as ICMP echo request or reply (PING), TCP, and UDP to obtain RTT metrics for connections between the local DNS server and participating GSLB sites. By default, the appliance uses a ping monitor and probes the local DNS server every 5 seconds. The appliance then waits 2 seconds for the response. If a response is not received in that time, it uses the TCP DNS monitor for probing.

However, you can modify the time interval for probing the local DNS server to accommodate your configuration.

To modify the probing interval by using the command line interface

At the command prompt, type:

```
set lb monitor <monitorName> <type> -interval <integer> <units> -
    resptimeout <integer> <units>
<!--NeedCopy-->
```

Example:

```
set lb monitor ldns-tcp LDNS-TCP -interval 10 sec -resptimeout 5 sec
<!--NeedCopy-->
```

To modify the probing interval by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Monitors**, and double-click the monitor that you want to modify (for example, ping).

API method

September 14, 2021

You can use the API method to determine the best performing GSLB service. The API method for GSLB uses a REST API to determine the best performing GSLB service.

In the API method, when GSLB receives a DNS request from a client, it evaluates the request against the specified rule. If GSLB encounters the HTTP callout expression, SYS.HTTP_CALLOUT(<name>), it

invokes a REST API request to an HTTP callout agent. GSLB uses the response from the HTTP callout agent to decide the best performing service. In the DNS response, GSLB returns the IP address of the best performing service, back to the client.

To configure a GSLB API method by using the CLI

Perform the following to configure the GSLB API method:

1. Configure an HTTP callout.

   For more information, see Configuring an HTTP callout.

   At the command prompt, type:

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-
    port <port>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod (GET | POST)] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)
    > ...] [-bodyExpr <string>] [-fullReqExpr <string>] [-scheme (
    http | https)] [-resultExpr <string>] [-cacheForSecs <secs>] [-
    comment <string>]
<!--NeedCopy-->
```

   Example:

```
add policy httpCallout GSLB_Method_API -IPAddress 208.111.39.237 -
    port 443 -returnType TEXT -hostExpr  "\" hopx.gslb.com\ ""  -
    urlStemExpr  "\" /zones/1/customers/92395/apps/6/decision\ ""
    -headers Authorization( "Basic 19fbe6db-4332-4e3f-a8bc-
    ee47bdc726f8") -parameters ip(DNS.REQ.OPT.ECS.IP.
    TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme
    https -resultExpr  "HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
    XPATH_JSON(xp%/providers/Val[1]/provider%)"  -cacheForSecs 30
<!--NeedCopy-->
```

2. Specify the API method for load balancing. GSLB evaluates the DNS request against the specified rule.

   At the command prompt, type:

```
add gslb vserver <name> <serviceType>  [-lbMethod <lbMethod>] [-
    backupLBMethod <backupLBMethod>] -rule <expression>
<!--NeedCopy-->
```

   Example:


```
1 add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN
      -rule  “sys.http_callout(GSLB_Method_API)”
2 <!--NeedCopy-->
```

Sample configuration for integrating GSLB and ITM using API as the LB method

This configuration allows GSLB to use the internet visibility aspects of the Citrix intelligent traffic management (ITM) to determine the best performing GSLB service.

```
1  /* Enable ns features */
2
3  enable ns feature lb gslb cs
4
5  /* This is a named expression that is used in the HTTP callout, used
      for result expression. */
6
7  add policy expression exp1 "HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
      XPATH_JSON(xp%/providers/Val[1]/provider%)"
8
9  /* This is a named expression that is used in HTTP callout, used for
      host expression. */
10
11 add policy expression exp2 "\"hopx.cedexis.com\""
12
13 /* This is the HTTP callout configured to request the ITM for the GSLB
      decision. */
14
15 add policy httpCallout ITM_OpenMix_API -IPAddress 208.111.39.237 -port
      80 -returnType TEXT -hostExpr exp2 -urlStemExpr "\"/zones/1/
      customers/61770/apps/3/decision\"" -headers Authorization("Basic
      a310697a-1d69-48bf-8f36-55742a8e894e") -parameters ip(DNS.REQ.OPT.
      ECS.IP.TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme
      http -resultExpr exp1 -cacheForSecs 30
16
17 /* Add service 1 */
18 add service sg1 98.136.103.24 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
       -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
      svrTimeout 360 -CKA NO -TCPB NO -CMP NO
19
20 /* Add service 2 */
21 add service sg2 172.217.194.113 HTTP 80 -gslb NONE -maxClient 0 -maxReq
       0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180
      -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
```



```
22
23 /* Add ADNS service */
24
25 add service adns1 10.102.217.106 ADNS 53 -gslb NONE -maxClient 0 -
      maxReq 0 -cip DISABLED -usip NO -useproxyport NO -sp OFF -cltTimeout
       120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO
26
27 /* Add lb vserver 1 for service 1 */
28 add lb vserver lbvs1 HTTP 10.102.217.116 80 -persistenceType NONE -
      cltTimeout 180
29
30 /* Add lb vserver 2 for service 2 */
31 add lb vserver lbvs2 HTTP 10.102.217.117 80 -persistenceType NONE -
      cltTimeout 180
32
33 /* Bind service 1 to lb vserver 1 */
34
35 bind lb vserver lbvs1 sg1
36
37 /* Bind service 2 to lb vserver 2 */
38
39 bind lb vserver lbvs2 sg2
40
41 /* Configure API GSLB method on GSLB virtual server to call the HTTP
      callout. This HTTP callout requests the ITM for the GSLB decision
      and returns GSLB service name, which should serve the request. */
42
43 add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN -
      rule "sys.http_callout(ITM_OpenMix_API)" -tolerance 0
44
45 /* Add GSLB site */
46
47 add gslb site site1 10.102.217.106 -publicIP 10.102.217.106
48
49 /* Add GSLB service 1 */
50
51 add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai_1
      10.102.217.116 HTTP 80 -publicIP 10.102.217.116 -publicPort 80 -
      maxClient 0 -siteName site1 -sitePersistence HTTPRedirect -
      sitePrefix gs2. -cltTimeout 180 -svrTimeout 360 -downStateFlush
      ENABLED
52
53 /* Add GSLB service 2 */
54
55 add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai 10.102.217.117
```



```
    HTTP 80 -publicIP 10.102.217.117 -publicPort 80 -maxClient 0 -
    siteName site1 -sitePersistence HTTPRedirect -sitePrefix gs1. -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
56
57 /* Bind the GSLB service 1 to GSLB server 1 */
58 bind gslb vserver vs1 -serviceName
    aws_ec2_ap_south_1_asia_pacific_mumbai_1
59
60 /* Bind the GSLB service 2 to GSLB server 2 */
61 bind gslb vserver vs1 -serviceName
    aws_ec2_ap_south_1_asia_pacific_mumbai
62
63 /* Bind a domain name to the GSLB virtual server */
64 bind gslb vserver vs1 -domainName testruchit104.com -TTL 5
65
66 <!--NeedCopy-->
```

Configure static proximity

September 14, 2021

For the static proximity method to work, you must either configure the Citrix ADC appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the GSLB method.

This document includes the following information:

- Adding a Location File to Create a Static Proximity Database
- Adding Custom Entries to a Static Proximity Database
- Setting the Location Qualifiers
- Specifying the Proximity Method
- Synchronizing GSLB Static Proximity Database

Add a location file to create a static proximity database

September 14, 2021

A static proximity database is a UNIX-based ASCII file. Entries added to this database from a location file are called static entries. Only one location file can be loaded on a Citrix ADC appliance. Adding a new location file overrides the existing file. The number of entries in the static proximity database is limited by the configured memory in the Citrix ADC appliance.

The static proximity database can be created in the default format or in a format derived from commercially configured third party databases (such as www.maxmind.com and www.ip2location.com).

The Citrix ADC appliance includes the following two IP geolocation database files. These are GeoLite2 files, published by MaxMind.

- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4
- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6

These database files are available in a format supported by the Citrix ADC appliance in the directory /var/netscaler/inbuilt_db.

You can use these IP geolocation databases as the location file for the static proximity based GSLB method, or in location based policies.

These databases vary in the details they provide. There is no strict enforcement of the database file format, except that the default file has format tags. The database files are ASCII files that use a comma as the field delimiter. There are differences in the structure of fields and the representation of IP addresses in the locations.

The format parameter describes the structure of the file to the Citrix ADC appliance. Specifying an incorrect value for the format option can corrupt the internal data.

> **Note**
>
> - After an upgrade, if the /var/netscaler/inbuilt_db/ directory contains the database file (Citrix_Netscaler_InBuilt_GeoIP_DB.csv) from the earlier Citrix ADC software versions, the file is retained.
> - The default location of the database file is /var/netscaler/locdb, and on a high availability (HA) setup, an identical copy of the file must be present in the same location on both Citrix ADC appliances.
> - If the location file is stored in a location other than the default location, then specify the path of the location file.
> - For admin partitions, the default path is: /var/partitions/<partitionName>/netscaler/locdb.
> - Some databases provide short country names according to ISO-3166 and long country names as well. The Citrix ADC uses short names when storing and matching qualifiers.
> - To create a static proximity database, log on to the UNIX shell of the Citrix ADC appliance and use an editor to create a file with the location details in one of the Citrix ADC supported formats.

To add a static location file by using the CLI

At the command prompt, type:

```
add locationFile <locationFile> [-format <format>]
-  show locationFile
<!--NeedCopy-->
```

Example:

```
add locationFile /var/netscaler/locdb/nsgeo1.0 -format netscaler
Done

show locationFile
Location File: /var/netscaler/locdb/nsgeo1.0
Format: netscaler
 Done
>
<!--NeedCopy-->
```

Example:

```
add locationFile /var/netscaler/inbuilt_db/
   Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4 -format netscaler

add locationFile6 /var/netscaler/inbuilt_db/
   Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6 -format netscaler
<!--NeedCopy-->
```

To add a static location file by using the GUI:

1. Navigate to **AppExpert** > **Location**, click the **Static Database** tab.
2. Click **Add** to add a static location file.

You can view an imported location file database by using the **View Database** dialog box in the configuration utility. There is no CLI equivalent.

To view a static location file by using the GUI:

1. Navigate to **AppExpert** > **Location**, click the **Static Database** tab.
2. Select a static location file, and from the **Action** list, click **View Database**.

To convert a location file into the Citrix ADC format:

By default, when you add a location file, it is saved in the Citrix ADC format. You can convert a location file of other formats into the Citrix ADC format.

> **Note**: The nsmap option can be accessed only from the command line interface. The conversion is possible only into the Citrix ADC format.

To convert the static database format, at the CLI prompt, type the following command:

```
nsmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
<!--NeedCopy-->
```

Example:

```
 nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.
    csv
<!--NeedCopy-->
```

Script to convert Maxmind GeoLite2 database format to Citrix ADC database format

MaxMind GeoIP database cannot be used directly in Citrix ADC. The MaxMind GeoIP database must be converted into Citrix ADC format and then loaded for IP location detection in GSLB static proximity method and other features like policies.
You can use a script to convert the GeoLite2 database format to Citrix ADC database format. This script can be used to convert both IPv4 and IPv6 files.
The script is available in the location: https://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Format

Steps to convert GeoIP2 database to Citrix ADC format

1. Download the GeoLite2 City or GeoLite2 Country database in .csv format from https://dev.maxmind.com/geoip/geoip2/geolite2/.
2. Copy the file in a Citrix ADC directory (say /var). Unzip the file using the following shell command, which would create a directory with the same name.

   ```
   tar -xf <filename>
   ```

3. Download the script Convert_GeoIPDB_To_Netscaler_Format.pl from https://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Format and copy it to the directory created in step #2.
4. To check the acceptable options for the script execution, run the following command:

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl -help
   ```

   Various options available are:

   - <filename> IPv4 output file. Default output file name: Netscaler_Maxmind_GeoIP_DB_IPv4.csv

- `-p <filename>` IPv6 output file. Default output file name: Netscaler_Maxmind_GeoIP_DB_IPv6.csv
- `-logfile <filename>` File containing list of events/messages
- `-debug` Prints all the messages to STDOUT

5. Execute the following command to convert the GeoLite2 database format to Citrix ADC database format.

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl
   ```

   > **Note:** The operation can take up to 5 minutes.

   The default file names used in the script are that of the Maxmind GeoLite2 City based database. If you have downloaded GeoLite2 Country database, you must provide the input file names accordingly as listed.

   - `-b <filename>` name of IPv4 block file to be converted. Default file name: GeoLite2-City-Blocks-IPv4.csv
   - `-i <filename>` name of IPv6 block file to be converted. Default file name: GeoLite2-City-Blocks-IPv6.csv
   - `-l <filename>` name of location file to be converted. Default file name: GeoLite2-City-Locations-en.csv

   Example:

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl -b GeoLite2-Country-
      Blocks-IPv4.csv -i GeoLite2-Country-Blocks-IPv6.csv -l
      GeoLite2-Country-Locations-en.cs
   <!--NeedCopy-->
   ```

   The following are the output files generated after running the script.

   - Netscaler_Maxmind_GeoIP_DB_IPv4.csv
   - Netscaler_Maxmind_GeoIP_DB_IPv6.csv

6. Once the conversion of the database into Citrix ADC format is complete, use the following command to start using it.

   ```
   add locationFile <locationFile>
   ```

Add a third-party static database file on a Citrix ADC appliance

Perform the following steps to add a third-party static database file on a Citrix ADC appliance.

1. Obtain the location database file from a third-party vendor, such as www.maxmind.com or www.ip2location.com.
2. Copy the location database file to the Citrix ADC appliance using the WinSCP utility.

> **Note**
>
> The default location of the database file on the appliance is **/var/netscaler/locdb**.

3. Execute the following command to add a static location file:

```
add location file <locationfile Name> -format LocationFormat
<!--NeedCopy-->
```

4. Execute the following command to ensure that the location database is loaded:

```
show location parameter
<!--NeedCopy-->
```

   This command displays the parameters, such as number of static entries. If the database is not loaded correctly, this command also displays an error message. A maximum of 3M-1 (3 million minus one) entries can be loaded.

5. Execute the following command to view the location of the GSLB site:

```
show gslb service
<!--NeedCopy-->
```

> **Note**
>
> - If the database is loaded correctly, the location of the GSLB sites is automatically populated in the database.
> - You can specify only one location file in the configuration on the appliance.
> - If the appliances are in a high availability setup, then one appliance must copy the database from the other appliance.
> - If no match is found for an incoming IP address, the request is processed using the Round Robin method.

6. Execute the following command to configure the GSLB method on the appliance:

```
set gslb vserver GSLBVserverName -lbMethod MethodType
<!--NeedCopy-->
```

Add custom entries to a static proximity database

September 14, 2021

Custom entries take precedence over static entries in the proximity database. You can add a maximum of 500 custom entries. For a custom entry, denote all omitted qualifiers with an asterisk (*) and, if

qualifiers have a period or space in the name, enclose the parameter in double quotation marks. The first 31 characters are evaluated for each qualifier. You can also provide the longitude and latitude of the geographical location of the IP address-range for selecting a service with the static proximity GSLB method.

To add custom entries by using the command line interface

At the command prompt, type the following commands to add a custom entry to the static proximity database and verify the configuration:

```
add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer
   >[-latitude <integer>]]
show location
<!--NeedCopy-->
```

Example:

```
>add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
<!--NeedCopy-->
```

```
>show location
<!--NeedCopy-->
```

Parameters for adding custom entries

- IPfrom

  First IP address in the range, in dotted decimal notation. This is a mandatory argument.

- IPto

  Last IP address in the range, in dotted decimal notation. This is a mandatory argument.

- preferredLocation

  String of qualifiers, in dotted notation, describing the geographical location of the IP address range. Each qualifier is more specific than the one that precedes it, as in continent.country.region.city.isp.organization. For example,”NA.US.CA.San Jose.ATT.citrix”.

  Note: A qualifier that includes a dot (.) or space ( ) must be enclosed in double quotation marks.

  This is a mandatory argument. Maximum Length: 197

- longitude

  Numerical value, in degrees, specifying the longitude of the geographical location of the IP address-range.

Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location.

Maximum value: 180

- latitude

  Numerical value, in degrees, specifying the latitude of the geographical location of the IP address-range.

  Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location.

  Maximum value: 180

To add custom entries by using the configuration utility

Navigate to **AppExpert** > **Location**, click the **Custom Entries** tab, and add the custom entries.

Set location parameters

September 14, 2021

The database used to implement static proximity contains the location of the GSLB sites. Each location contains an IP address range and up to six qualifiers for that range. The qualifiers are literal strings and are compared in a prescribed order at run time. Every location must have at least one qualifier. The qualifier labels define the meaning of the qualifiers (context), which are user defined. Citrix ADC has two built-in contexts:

Geographic context, which has the following qualifier labels:

- Qualifier 1 – “Continent”
- Qualifier 2 – “Country”
- Qualifier 3 – “State”
- Qualifier 4 – “City”
- Qualifier 5 – “ISP”
- Qualifier 6 – “Organization”

Custom entries, which have the following qualifier labels:

- Qualifier 1 – “Qualifier 1”
- Qualifier 2 – “Qualifier 2”
- Qualifier 3 – “Qualifier 3”

- Qualifier 4 – “Qualifier 4”
- Qualifier 5 – “Qualifier 5”
- Qualifier 6 – “Qualifier 6”

If the geographic context is set with no Continent qualifier, Continent is derived from Country. Even the built-in qualifier labels are based on the context, and the labels can be changed. These qualifier labels specify the locations mapped with the IP addresses used to make static proximity decisions.

To perform a static proximity-based decision, the Citrix ADC appliance compares the location attributes (qualifiers) derived from the IP address of the local DNS server resolver with the location attributes of the participating sites. If only one site matches, the appliance returns the IP address of that site. If there are multiple matches, the site selected is the result of a round robin on the matching GSLB sites. If there is no match, the site selected is a result of a round robin on all configured sites. A site that does not have any qualifiers is considered a match.

The GEO rules for location-based policy expression allow you to check wildcard matches. This feature checks whether wildcard qualifiers match any other qualifier including non-wildcard or not. The wildcard match is performed by using the `matchWildcardtoany` attribute that is added to the `set locationParameter` command.

The `matchWildcardtoany` attribute can be set to the following values:

- **Yes:** Wildcard qualifiers match any other qualifiers.
- **No:** Wildcard qualifiers do not match non-wildcard qualifiers but match other wildcard qualifiers. The default option is **No**.
- **Expression:** Wildcard qualifiers in an expression match any qualifier in an LDNS location but wildcard qualifiers in the LDNS location do not match non-wildcard qualifiers in an expression.

Example:

```
1 add dns policy policy1 "CLIENT.IP.SRC.MATCHES_LOCATION(\"Continent.
     country *.*.*.* \ “)”  <action>
2 <!--NeedCopy-->
```

To set the location parameters by using the CLI

At the command prompt, type:

```
1 set locationparameter -context <context> -q1label <string> [-q2label <
     string>] [-q3label <string>] [-q4label <string>] [-q5label <string>]
      [-q6label <string>] -matchWildcardtoany [Yes | No | Expression]
2 <!--NeedCopy-->
```

Example:


```
set locationparameter -context custom -q1label asia -matchWildcardtoany
    Yes
<!--NeedCopy-->
```

To set the location parameters by using the GUI

1. Navigate to **Traffic Management > GSLB > Database and Entries**.
2. Under **Settings**, click **Change Location Parameters**.
3. In the **Configure Location Parameters** page, set the location parameters.

Specify proximity method

September 14, 2021

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the command line interface

At the command prompt, type the following commands to configure static proximity and verify the configuration:

```
set gslb vserver <name> -lbMethod STATICPROXIMITY
show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
<!--NeedCopy-->
```

To specify static proximity by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server.
2. Click the **Method** section and from the **Choose Method** drop-down list, select **STATICPROXIMITY**.

Synchronize GSLB static proximity database

September 14, 2021

Synchronizing a global server load balancing (GSLB) static proximity database requires that one of the sites be identified as the master GSLB node. Any site in the topology can be designated as the master node. The rest of the GSLB nodes are automatically designated as slave nodes.

Synchronizing GSLB static proximity databases synchronizes the files in the /var/netscaler/locdb directory across the slave nodes. During the synchronization process, the master node fetches the running configuration from each of the slave nodes and compares it to the configuration on the master node. The master GSLB node uses the rsync program to synchronize the static proximity database across the slave nodes. To speed up the synchronization process, the rsync program makes only enough changes to eliminate the differences between the two files. The synchronization process cannot be rolled back.

The following example synchronizes Site2, which is a slave site, to master site Site1. The administrator enters the **sync gslb config** command on Site1:

```
sync gslb config -nowarn
Sync Time: Feb 24 2014 14:56:16
Retrieving local site info: ok
Retrieving all participating gslb sites info:
0 bytes in 0 blocks
ok
site1[Master]:
        Getting Config: ok
site2[Slave]:
        Syncing gslb static proximity database: ok
        Getting Config: ok
        Comparing config: ok
        Applying changes: ok
 Done
<!--NeedCopy-->
```

Configure site-to-site communication

September 14, 2021

GSLB site-to-site communication is between the remote procedure call (RPC) nodes that are associated with the communicating sites. A master GSLB site establishes connections with slave sites to synchronize GSLB configuration information and to exchange site metrics.

An RPC node is created automatically when a GSLB site is created, and is assigned an internally generated user name and password. The Citrix ADC appliance uses this user name and password to authenticate itself to remote GSLB sites during connection establishment. No configuration steps are necessary for an RPC node, but you can specify a password of your choice, enhance security by encrypting the information that GSLB sites exchange, and specify a source IP address for the RPC node.

The appliance needs a Citrix ADC owned IP address to use as the source IP address when communicating with other GSLB sites. By default, the RPC nodes use either a subnet IP (SNIP) address, but you might want to specify an IP address of your choice.

The following topics describe the behavior and configuration of RPC nodes on the Citrix ADC appliance:

Changing the password of an RPC node

Citrix recommends you to secure the communication between sites in your GSLB setup by changing the password of each RPC node. After you change the password for the RPC node of the local site, you must manually propagate the change to the RPC node at each of the remote sites.

The password is stored in encrypted form. You can verify that the password has changed by using the show rpcNode command to compare the encrypted form of the password before and after the change.

> **Note:** GSLB uses internal user account. For enhanced security, Citrix recommends that you change the internal user account password as well. Internal user account password is changed through RPC node password.

To change the password of an RPC node by using the command line interface

At the command line, type the following commands to change the password of an RPC node:

```
set ns rpcNode <IPAddress> {
 -password }

show ns rpcNode
<!--NeedCopy-->
```

Example:

```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
```



```
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
    SrcIP:  *               Secure:  OFF
Done
>

<!--NeedCopy-->
```

To unset the password of an RPC node by using the command line interface

To unset the password of an RPC node by using the CLI, type the unset rpcNode command, the IP address of the RPC node, and the password parameter, without a value.

To change the password of an RPC node by using the configuration utility

Navigate to System > Network > RPC, select the RPC node, and change the password.

Encrypt the exchange of site metrics

You can secure the information that is exchanged between GSLB sites by setting the secure option for the RPC nodes in the GSLB setup. With the secure option set, the Citrix ADC appliance encrypts all communication sent from the node to other RPC nodes.

To encrypt the exchange of site metrics by using the command line interface

At the command prompt, type the following commands to encrypt the exchange of site metrics and verify the configuration:

```
set ns rpcNode <IPAddress> [-secure ( YES | NO )]
show rpcNode
<!--NeedCopy-->
```

Example:

```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:
    192.0.2.3     Secure:  ON
```



```
 Done
>
<!--NeedCopy-->
```

To unset the secure parameter by using the command line interface

To unset the secure parameter by using the CLI, type the unset rpcNode command, the IP address of the RPC node, and the secure parameter, without a value.

To encrypt the exchange of site metrics by using the Citrix ADC configuration utility

1. Navigate to System > Network > RPC and double-click a RPC node.
2. Select the **Secure** option, and click **OK**.

Configure source IP address for an RPC node

By default, the Citrix ADC appliance uses a Citrix ADC owned subnet IP (SNIP) address as the source IP address for an RPC node, but you can configure the appliance to use a specific SNIP address. If a SNIP address is not available, the GSLB site cannot communicate with other sites. In such a scenario, you must configure either the NSIP address or a virtual IP (VIP) address as the source IP address for an RPC node. A VIP address can be used as the source IP address of an RPC node only if the RPC node is a remote node. If you configure a VIP address as the source IP address and remove the VIP address, the appliance uses a SNIP address.

> **Note**
>
> From NetScaler 11.0.64.x release onwards, you can configure the appliance to use GSLB Site IP address as the source IP address for an RPC node.

To specify a source IP address for an RPC node by using the command line interface

At the command prompt, type the following commands to change the source IP address for an RPC node and verify the configuration:

```
set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
show ns rpcNode
<!--NeedCopy-->
```

Example:

```
set rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
```



```
3 show rpcNode
4 <!--NeedCopy-->
```

```
1 IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3
     Secure: OFF
2 Done
3 <!--NeedCopy-->
```

To unset the source IP address parameter by using the command line interface

To unset the source IP address parameter by using the CLI, type the unset rpcNodecommand, the IP address of the RPC node, and the srcIP parameter, without a value.

To specify a source IP address for an RPC node by using the Citrix ADC configuration utility

1. Navigate to System > Network > RPC and double-click a RPC node.
2. In the Source IP Address field, enter the IP address that you want the RPC node to use as the source IP address and click OK.

> **Important**
>
> The source IP address cannot be synchronized across the sites participating in GSLB because the source IP address for a RPC node is specific to each Citrix ADC appliance. Therefore, after you force a synchronization (using the sync gslb config –forceSync command or by selecting the ForceSync option in the GUI), you have to manually change the source IP addresses on the other Citrix ADC appliances.

Configure metrics exchange protocol

September 14, 2021

The data centers in a GSLB setup exchange metric with each other through the metrics exchange protocol (MEP), which is a proprietary protocol for Citrix ADC appliance. The exchange of the metric information begins when you create a GSLB site. These metrics comprise load, network, and persistence information.

MEP is required for health checking of data centers to ensure their availability. A connection for exchanging network metric (round-trip time) can be initiated by either of the data centers involved in the exchange, but a connection for exchanging site metrics is always initiated by the data center with the lower IP address. By default, the data center uses a subnet IP address (SNIP) to establish a connection to the IP address of a different data center. However, you can configure a specific SNIP, virtual IP

(VIP) address, or the NSIP address, as the source IP address for metrics exchange. The communication process between GSLB sites uses TCP port 3011 or 3009, so this port must be open on firewalls that are between the Citrix ADC appliances.

Note: You can configure a SNIP or a GSLB site IP address as the source IP address for metrics exchange. For more information, see Configure source IP address for an RPC node.

If the source and target sites (the site that initiates a MEP connection and the site that receives the connection request, respectively) have both private and public IP addresses configured, the sites exchange MEP information by using the public IP addresses.

You can also bind monitors to check the health of remote services as described in "Monitoring GSLB Services." When monitors are bound, metric exchange does not control the state of the remote service. If a monitor is bound to a remote service and metric exchange is enabled, the monitor controls the health status. Binding the monitors to the remote service enables the Citrix ADC appliance to interact with a non-citrix ADC load balancing device. The Citrix ADC appliance can monitor non-citrix ADC devices but cannot perform load balancing on them unless monitors are bound to all GSLB services and only static load balancing methods (such as the round robin, static proximity, or hash-based methods) are used.

With NetScaler release 11.1.51.x or later, to avoid unnecessary disruption of services, you can set a time delay for marking GSLB services as DOWN when a MEP connection goes DOWN.

MEP state in a high availability setup

In a high availability setup, the primary node establishes connections with the remote sites and the MEP state is not synchronized from the primary node to secondary nodes. Therefore, the MEP state in secondary node remains DOWN. When the secondary node becomes primary, it establishes MEP connections with the new GSLB site and updates the MEP state accordingly.

Enable site metrics exchange

Site metrics exchanged between the GSLB sites include the status of each load balancing, or content switching virtual server, the current number of connections, the current packet rate, and current bandwidth usage information.

The Citrix ADC appliance needs this information to perform load balancing between the sites. The site metric exchange interval is 1 second. A remote GSLB service must be bound to a local GSLB virtual server to enable the exchange of site metrics with the remote service.

To enable or disable site metrics exchange by using the command line interface

At a command prompt, type the following commands to enable or disable site metric exchange and verify the configuration:

```
set gslb site <siteName> -metricExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To enable or disable site metric exchange by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and select the site.
2. In the **Configure GSLB Site** dialog box, select the **Metric Exchange** option.

Enable network metric exchange

If your GSLB sites use the round-trip time (RTT) load balancing method, you can enable or disable the exchange of RTT information about the client's local DNS service. This information is exchanged every 5 seconds.

For details about changing the GSLB method to a method based on RTT, see GSLB Methods.

To enable or disable network metric information exchange by using the command line interface

At the command prompt, type the following commands to enable or disable network metric information exchange and verify the configuration:

```
set gslb site <siteName> -nwmetricExchange (ENABLED|DISABLED)
show gslb site <<siteName>
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
```



```
<!--NeedCopy-->
```

To enable or disable network metric information exchange by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. In the **Configure GSLB Site** dialog box, select the **Network Metric Exchange** option.

Configuring a time delay for the GSLB services to be marked as DOWN when a MEP connection goes DOWN

If the status of a MEP connection to a remote site changes to DOWN, the status of every GSLB service on that remote site is marked as DOWN, although the site might not actually be DOWN.

You can now set a delay to allow some time for reestablishment of the MEP connection before the site is marked as DOWN. If the MEP connection is back UP before the delay expires, the services are not affected.

For example, if you set the delay 10, the GSLB services are marked as DOWN until the MEP connection has been DOWN for 10 seconds. If the MEP connection is back UP within 10 seconds, the GSLB services remain in the UP state.

Note: This delay is applicable only to services not bound to a monitor. The delay does not affect the trigger monitors.

To set a time delay by using the command line interface

At the command prompt, type the following command:

```
set gslb parameter** - GSLBSvcStateDelayTime <sec>
<!--NeedCopy-->
```

Example:

set gslb parameter - GSLBSvcStateDelayTime 10

> **Note**
>
> In a hierarchical deployment (parent-child topology), if you configure the GSLB service on both the parent and child sites, set the GSLB parameter on both the parent and child sites. If you do not configure the GSLB service on the child site, set the GSLB parameter only on the parent site.

To set a time delay by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Change GSLB Settings**.
2. In the **GSLB Service State Delay Time (secs)** box, type the time delay in seconds.

Configure a learning time for GSLB services when MEP connection status comes up to avoid flaps on GSLB services

When a node reboots or during HA failover, the system is initialized. Then, the node must learn current information about the configured local and child services to communicate the service state to remote nodes through MEP. The node takes some time to learn the correct information. Meanwhile, if a peer node connects to this node and requests for an update, the node might send an incorrect service state and statistics. This incorrect information might cause service flap and other functionality related issues on the remote peer nodes. To avoid this scenario, you can now set a learning time for the local and child GSLB service.

When a learning timeout is configured, the GSLB site gets some buffer time (learning timeout) to learn the correct statistics about its local and child services. When a service is in a learning phase, the remote GSLB site gets this information in MEP update, and does not honor the primary site state and statistics received through MEP for that service.

GSLB services enter the learning phase in any of the following scenarios.

- Citrix ADC appliance is rebooted
- High availability failover has occurred
- Owner node in a cluster GSLB setup is changed
- MEP is enabled on a local node
- GSLB site comes out of island scenario. A GSLB Site becomes island when it is not connected to any other site.

In a parent-child deployment, the backup parent (if configured) selectively moves the adopted child site's GSLB services to the learning phase when the primary parent goes DOWN.

To set a service state learning time by using the CLI

At the command prompt, type the following command:

```
set gslb parameter - SvcStateLearningTime <sec>
<!--NeedCopy-->
```

You can set "SvcStateLearningTime" in seconds. Default value is 0 and maximum value is 3600. This parameter is applicable only if monitors are not bound to GSLB services.

Example:


```
set gslb parameter - SvcStateLearningTime 10
<!--NeedCopy-->
```

To set a service state learning time by using the GUI

1. Navigate to **Configuration > Traffic Management > GSLB > Dashboard > Change GSLB Settings**.

   The **Set GSLB Parameters** page appears.

2. In the **GSLB Service State Learning Time (secs)** field, type the learning time in seconds.

Enable persistence information exchange

You can configure Citrix ADC appliance to provide persistent connections, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

You can enable or disable the exchange of persistence information at each site. This information is exchanged once every 5 seconds between Citrix ADC appliances participating in GSLB.

For details about configuring persistence, see Configuring Persistent Connections.

To enable or disable persistence information exchange by using the command line interface

At the command prompt, type the following commands to enable or disable persistence-information exchange and verify the configuration:

```
set gslb site <siteName> -sessionExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To enable or disable persistence information exchange by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and double-click the site.

2. In the **Configure GSLB Site** dialog box, select, or clear the **Persistence Session Entry Exchange** check box.

Configure GSLB by using a wizard

September 14, 2021

You can now use a wizard to configure the GSLB deployment types: active-active, active-passive, and parent-child.

This wizard is available in the GUI. To access the wizard, navigate to **Configuration** > **Traffic Management** > **GSLB** and click **Get Started**.

You can also access this wizard from the GSLB dashboard. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Dashboard** and click **Configure GSLB**.

Note: You can also configure the GSLB entities individually.

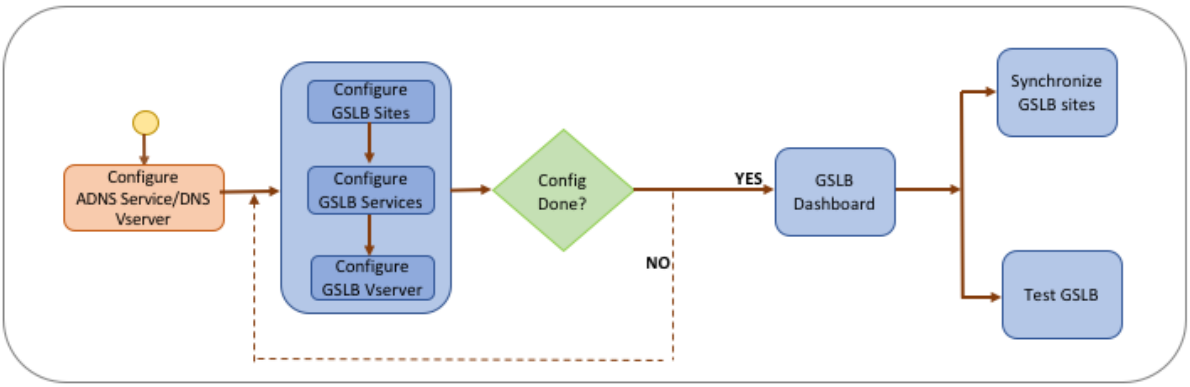
- Active-Active Site Configuration
- Active-Passive Site Configuraiton
- Parent-Child Topology Configuration

> **Important**
>
> This feature is supported in High Availability deployment and not in Admin Partition and Cluster deployments.

Configure active-active site

September 14, 2021

The following figure shows the workflow involved in a GSLB active-active site configuration.



Before you begin configuring an active-active site, make sure you have configured a standard load balancing setup for each server farm or data center.

Also, for synchronizing the GSLB configuration across the GSLB sites in the deployment, make sure that:

- Local GLSB Sites are configured on all the appliances in the GSLB configuration.
- You have enabled management access on all the GSLB Sites in the configuration.
- You have configured the firewall to accept the auto synchronization and MEP connections.
- The master and slave Citrix ADC appliances are running the same Citrix ADC software versions.
- All the Citrix ADC appliances participating as sites should have the same Citrix ADC software version (the sites are not in a master-slave relationship).
- The RPC node password is same across all the GSLB sites in the GSLB configuration.

To configure an active-active site by using the wizard

On the Configuration tab, do the following:

1. Navigate to **Traffic Management** > **GSLB**, and then click **Get Started**.
2. If you have not configured an ADNS service or a DNS virtual server for the site, you can do it now.
   a) Click **View** and then click **Add**.
   b) Enter the service name, IP address and select the protocol (ADNS/ADNS_TCP) through which the data is exchanged with the service.
3. Select **Active-Active Site**.
4. Enter the fully qualified domain name and specify the time period for which the record must be cached by DNS proxies.
5. Configure the GSLB sites. Each site must be configured with a local GSLB site, and each site's configuration must include all the other sites as remote GSLB sites. There can be only one local site and all other sites are remote sites.
   a) Enter the site details, such as the site name and site IP address.
   b) Select either REMOTE or LOCAL site type.
   c) Optionally, change the RPC password and, if necessary, secure it.
   d) If a monitor is to be bound to the GSLB service, select the condition under which the monitor is to monitor the service. This will be effective only after a monitor is bound to the services. The possible conditions are:
      - **ALWAYS**. Monitor the GSLB service at all times.
      - **MEP Fails**. Monitor the GSLB service only when the exchange of metrics through MEP fails.
      - **MEP Fails and Service ID Down**. Exchange of metrics through MEP is enabled but the status of the service, updated through metrics exchange, is DOWN.
6. Configure the GSLB services. To create an active-active site you must add at least two GSLB services.

a) Enter the service details, such as service name, service type, and port number.
b) Associate the service with a site (local or remote) by selecting the GSLB site where the GSLB service belongs.
c) Select the monitor that must be bound to the service when MEP fails, if required. The service can be an existing server, or you can create a new server or a virtual server.
d) To associate an existing server, select the server name. The service IP address is automatically populated.
   - If the public IP address is different from the server IP, which can happen in a NAT environment, enter the public IP address and the port number of the public port.
   - To associate a new server, create a server by entering the server IP details and its public IP address and the public port number.
   - To associate a virtual server, select an already existing virtual server, or click + and add a new virtual server. This vserver is the load balancing vserver with which this GSLB service will be associated.

7. Configure the GSLB virtual servers.
   a) Enter the name of the GSLB virtual server name and select the DNS record type.
   b) Click **>** in the **Select Service** box and choose the GSLB services to be bound to the GSLB virtual server.
   c) Click **>** in the **Domain Binding** box to select the domain to be bound to this GSLB virtual server.
   d) Choose the GSLB method for selecting the best-performing GSLB service. The default values for GSLB method, backup method, and dynamic weight are auto-populated, by default. You can change them if required.
      - If you choose the **Algorithm based** method, select the primary and backup methods and also specify the dynamic weight option.
      - If you choose the **Static Proximity** method, select the backup method and the dynamic weight method. Also, provide the location of the database file by clicking the > icon or add a new location by clicking **+** in the Select a location database box.
      - If you choose the **Dynamic Proximity (RTT)** method, select the backup method and specify the dynamic weight option and the round-trip time value based on which the best-performing service is to be selected.
8. Click **Done** if the configuration is complete. The GSLB dashboard appears.
9. If you have modified the GSLB site configuration, click **Auto Synchronize GSLB** in the dashboard to synchronize the configuration to other sites in the GSLB setup.
   - Before synchronization, make sure that the local site's configuration includes information about the remote sites. Also, for the synchronization to be successful, the local site must be configured on the other Citrix ADC appliances.
   - If real-time synchronization is enabled, you do not have to click **Auto Synchronize GSLB**. The synchronization happens automatically. To enable real time synchronization, do the

following:

a) Navigate to **Traffic Management** > **GSLB** > **Dashboard** and click **Change GSLB Settings**.

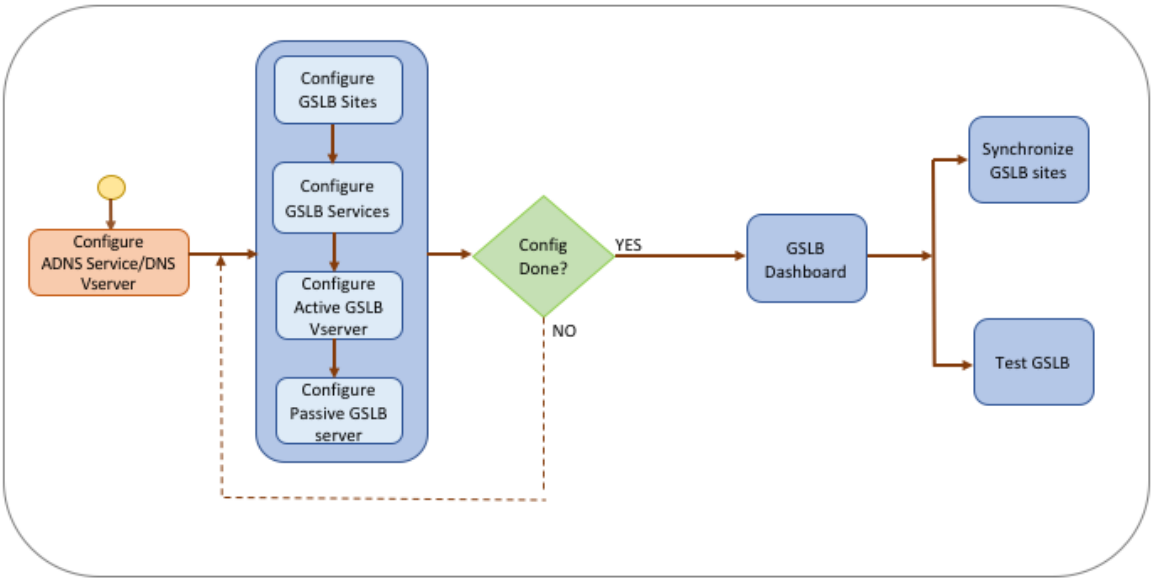
b) Select the **Automatic Config Sync** check box.

10. Click **Test GSLB Setup** to make sure that the ADNS services or the DNS servers are responding with the correct IP address for the domain name that is configured in the GSLB setup.

Configure active-passive site

September 14, 2021

The following figure shows the workflow involved in the active-passive site configuration.



Before you begin configuring an active-passive site, make sure you have configured a standard load balancing setup for each server farm or data center.

Also, for synchronizing the GSLB configuration across the GSLB sites in the deployment, make sure that:

- Local GLSB Sites are configured on all the appliances in the GSLB configuration.
- You have enabled management access on all the GSLB Sites in the configuration.
- You have configured the firewall to accept the auto synchronization and MEP connections.
- The master and slave Citrix ADC appliances are running the same Citrix ADC software versions.
- All the Citrix ADC appliances participating as sites should have the same Citrix ADC software version (the sites are not in a master-slave relationship).
- The RPC node password is same across all the GSLB sites in the GSLB configuration.

To configure an active-passive site by using the wizard

On the Configuration tab, do the following:

1. Navigate to **Traffic Management** > **GSLB**, and then click **Get Started**.
2. If you have not configured an ADNS service or a DNS virtual server for the site, you can do it now.
   a) Click **View** and then click **Add**.
   b) Enter the service name, IP address and select the protocol (ADNS/ADNS_TCP) through which the data is exchanged with the service.
3. Select **Active-Passive Site**.
4. Enter the fully qualified domain name and specify the time period for which the record must be cached by DNS proxies.
5. Configure the GSLB sites. Each site must be configured with a local GSLB site, and each site's configuration must include all the other sites as remote GSLB sites. There can be only one local site and all other sites are remote sites.
   a) Enter the site details, such as the site name and site IP address.
   b) Select either REMOTE or LOCAL site type.
   c) Optionally, change the RPC password and, if necessary, secure it.
   d) If a monitor is to be bound to the GSLB service, select the condition under which the monitor is to monitor the service. This will be effective only after a monitor is bound to the services. The possible conditions are:
      - **ALWAYS**. Monitor the GSLB service at all times.
      - **MEP Fails**. Monitor the GSLB service only when the exchange of metrics through MEP fails.
      - **MEP Fails and Service ID Down**. Exchange of metrics through MEP is enabled but the status of the service, updated through metrics exchange, is DOWN.
6. Configure the GSLB services.
   a) Enter the service details, such as service name, service type, and port number.
   b) Associate the service with a site (local or remote) by selecting the GSLB site where the GSLB service belongs.
   c) Select the monitor that must be bound to the service when MEP fails, if required. The service can be an existing server, or you can create a new server or a virtual server.
   d) To associate an existing server, select the server name. The service IP address is automatically populated.
      - If the public IP address is different from the server IP, which can happen in a NAT environment, enter the public IP address and the port number of the public port.
      - To associate a new server, create a server by entering the server IP details and its public IP address and the public port number.
      - To associate a virtual server, select an already existing virtual server, or click **+** and add a new virtual server. This vserver is the load balancing vserver with which this GSLB service will be associated.

7. Configure the GSLB backup virtual servers. The GSLB backup virtual servers become operational only when the primary GSLB virtual servers is inaccessible or it is marked DOWN for any reason.
   a) Enter the name of the GSLB virtual server name and select the DNS record type.
   b) Click **>** in **Service Binding**, and choose the GSLB services that must be bound to the GSLB virtual server.
   c) Choose the GSLB method for selecting the best-performing GSLB service. The default values for GSLB method, backup method, and dynamic weight are auto-populated, by default. You can change them if required.
      - If you choose the **Algorithm based** method, select the primary and backup methods.
      - If you choose the **Static Proximity** method, select the backup method and provide the location of the database file.
      - If you choose the **Dynamic Proximity (RTT)** method, select the backup method and specify the service weight and the RTT value based on which the best-performing service is to be selected.
8. Configure the GSLB virtual servers.
   a) Enter the name of the GSLB virtual server name and select the DNS record type.
   b) Click **>** in the **Select Service** box and choose the GSLB services to be bound to the GSLB virtual server.
   c) Click **>** in the **Domain Binding** box to select the domain to be bound to this GSLB virtual server.
   d) Choose the GSLB method for selecting the best-performing GSLB service. The default values for GSLB method, backup method, and dynamic weight are auto-populated, by default. You can change them if required.
      - If you choose the **Algorithm based** method, select the primary and backup methods and also specify the dynamic weight option.
      - If you choose the **Static Proximity** method, select the backup method and the dynamic weight method. Also, provide the location of the database file by clicking the **>** icon or add a new location by clicking **+** in the Select a location database box.
      - If you choose the **Dynamic Proximity (RTT)** method, select the backup method and specify the dynamic weight option and the round-trip time value based on which the best-performing service is to be selected.
9. Click **Done** if the configuration is complete. The GSLB dashboard appears.
10. If you have modified the GSLB site configuration, click **Auto Synchronize GSLB** in the dashboard to synchronize the configuration to other sites in the GSLB setup.
    - Before synchronization, make sure that the local site's configuration includes information about the remote sites. Also, for the synchronization to be successful, the local site must be configured on the other Citrix ADC appliances.
    - If real-time synchronization is enabled, you do not have to click **Auto Synchronize GSLB**.

The synchronization happens automatically. To enable real time synchronization, do the following:

a) Navigate to **Traffic Management** > **GSLB** > **Dashboard** and click **Change GSLB Settings**.

b) Select the **Automatic Config Sync** check box.

11. Click **Test GSLB Setup** to make sure that the ADNS services or the DNS servers are responding with the correct IP address for the domain name that is configured in the GSLB setup.

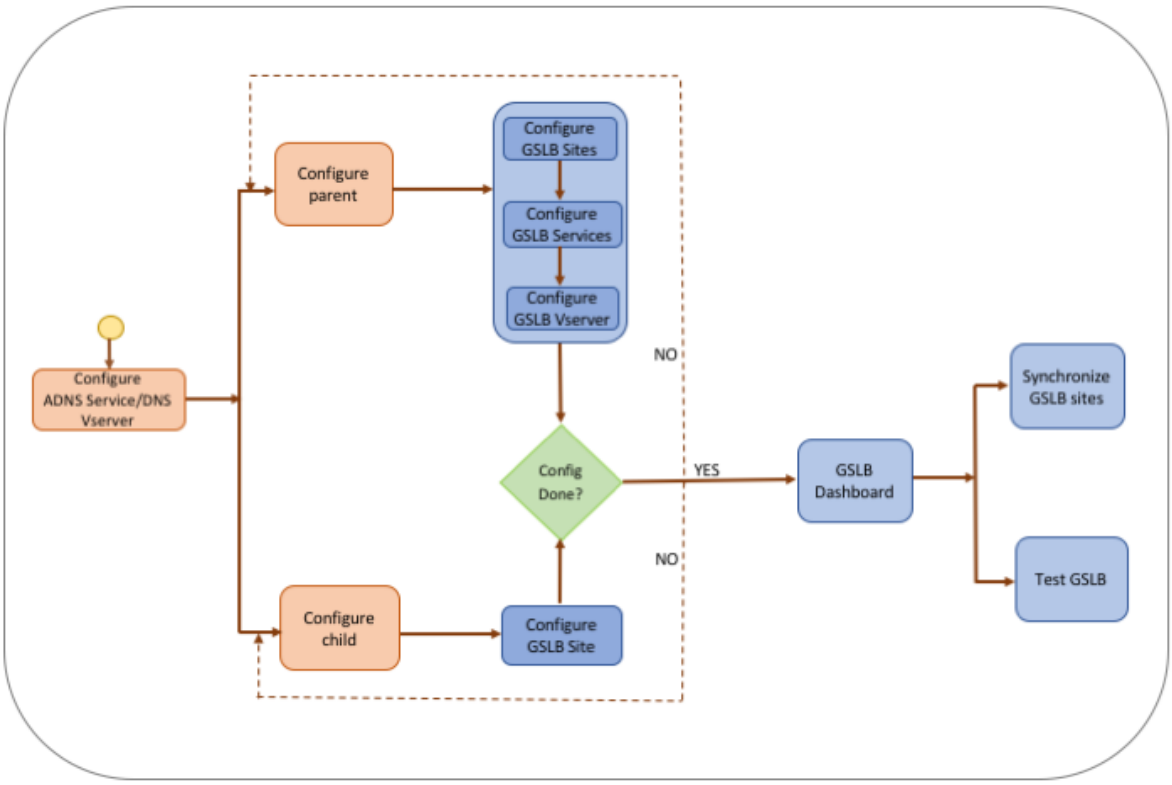
> **Note**
>
> For details about configuring GSLB entities of an active-passive GSLB setup for disaster recovery, see Configuring GSLB for Disaster Recovery.

Configure parent-child topology

September 14, 2021

In a parent-child topology, at the top level are parent sites, which have peer relationships with other parents. Each parent can have multiple child sites, and each parent site exchanges health information with its child sites and with other parent sites. However, a child site communicates only with its parent site.

The following figure shows the workflow involved in a GSLB parent-child topology configuration.



Before you begin configuring the parent-child topology deployment, make sure you have configured a standard load balancing setup for each server farm or data center.

Also, for synchronizing the GSLB configuration across the GSLB sites in the deployment, make sure that:

- Local GLSB Sites are configured on all the appliances in the GSLB configuration.
- You have enabled management access on all the GSLB Sites in the configuration.
- You have configured the firewall to accept the auto synchronization and MEP connections.
- All the Citrix ADC appliances participating as sites should have the sameCitrix ADC software version (the sites are not in a master-slave relationship).
- The RPC node password is same across all the GSLB sites in the GSLB configuration.

To configure a parent-child deployment by using the wizard

On the Configuration tab, do the following:

1. Navigate to **Traffic Management** > **GSLB**, and then click **Get Started**.
2. If you have not configured an ADNS server or a DNS virtual server for the site, you can do it now.
   a) Click **View** and then click **Add**.
   b) Enter the service name, IP address and select the protocol (ADNS/ADNS_TCP) through which the data is exchanged with the service.

3. Select **Parent-Child Topology**.
4. In the Select the site type field, choose;
   - **Parent** – When configuring the parent site, you must configure its associated child sites and also configure the other parent sites in the GSLB setup.
   - **Child** – When configuring the child site, you must configure only the child site and its parent site.

To configure a parent site

1. Enter the fully qualified domain name and specify the time period for which the record must be cached by DNS proxies.
2. Configure the GSLB sites. Each site must be configured with a local GSLB site, and each site's configuration must include all the other sites as remote GSLB sites. There can be only one local site. All other sites are remote sites. If the specified site IP address is owned by the appliance (for example, a MIP address or SNIP address), the site is a local site. Otherwise, it is a remote site.
3. Enter the site details, such as the site name and site IP address.
   a) Select the site type.
   b) Optionally, change the RPC password and, if necessary, secure it.
   c) If a monitor is to be bound to the GSLB service, select the condition under which the monitor is to monitor the service. This will be effective only after a monitor is bound to the services. The possible conditions are:
      - **Always**. Monitor the GSLB service at all times.
      - **MEP Fails**. Monitor the GSLB service only when the exchange of metrics through MEP fails.
      - **MEP Fails and Service is DOWN**. Exchange of metrics through MEP is enabled but the status of the service, updated through metrics exchange, is DOWN.
4. Configure the GSLB services.
   a) Enter the service details such as service name, service type, and port number.
   b) Associate the service with a site (local or remote) by selecting the GSLB site to which the GSLB service belongs.
   c) Select the monitor that must be bound to the service when MEP fails, if required. The service can be an existing server, or you can create a new server or a virtual server.
      - To associate an existing server, select the server name. The service IP address is auto-populated.
      - To associate a new server, create a server by entering the server IP details and its public IP address and the public port number.
      - To associate a virtual server, select an already existing virtual server or click **+** and add a new virtual server. This vserver is the load balancing vserver to which this GSLB service will be associated. If the public IP address is different from the server IP, which

can happen in a NAT environment, enter the public IP address and the public port number.

5. Configure the GSLB virtual servers.
   a) Enter the name of the GSLB virtual server name and select the DNS record type.
   b) Click **>** in the **Select Service** box and choose the GSLB services to be bound to the GSLB virtual server.
   c) Click **>** in the **Domain Binding** box to view the domain name that is bound to the GSLB virtual server.
   d) Choose the GSLB method for selecting the best-performing GSLB service. The default values for GSLB method, backup method, and dynamic weight are automatically populated by default. You can change them if required.
      - If you choose the **Algorithm based** method, select the primary and backup methods and also specify the dynamic weight option.
      - If you choose the **Static Proximity** method, select the backup method and the dynamic weight method. Also, provide the location of the database file by clicking the **>** icon or add a new location by clicking **+** in the Select a location database box.
      - If you choose the **Dynamic Proximity (RTT)** method, select the backup method and specify the service weight and the RTT value based on which the best-performing service is to be selected.
6. Click **Done** if the configuration is complete. The GSLB dashboard appears.
7. If you have modified the GSLB parent-site configuration, click **Auto Synchronize GSLB** to synchronize the configuration to the other parent sites in the GSLB setup. In a parent-child topology, synchronization for the child sites is skipped.
   - Before synchronization, make sure that the local site's configuration includes information about the remote sites.
   - If real-time synchronization is enabled, you do not have to click **Auto Synchronize GSLB**. The synchronization happens automatically. To enable real time synchronization, do the following:
     a) Navigate to **Traffic Management** > **GSLB** > **Dashboard** and click **Change GSLB Settings**.
     b) Select the **Automatic Config Sync** check box.
8. Click **Test GSLB Setup** to make sure that the ADNS services or the DNS servers are responding with the correct IP address for the domain name that is configured in the GSLB setup.

To configure a child site

1. Configure the GSLB sites.
   a) Enter the site details, such as the site name and site IP address.
   b) Select the site type.
   c) Optionally, change the RPC password and, if necessary, secure it. 4. If a monitor is bound

to the GSLB service, select the condition under which the monitor is to monitor the service. The possible conditions are:

- **Always**. Monitor the GSLB service at all times.
- **MEP Fails**. Monitor the GSLB service only when the exchange of metrics through MEP fails.
- **MEP Fails and Service is DOWN**. Exchange of metrics through MEP is enabled but the status of the service, updated through metrics exchange, is DOWN.

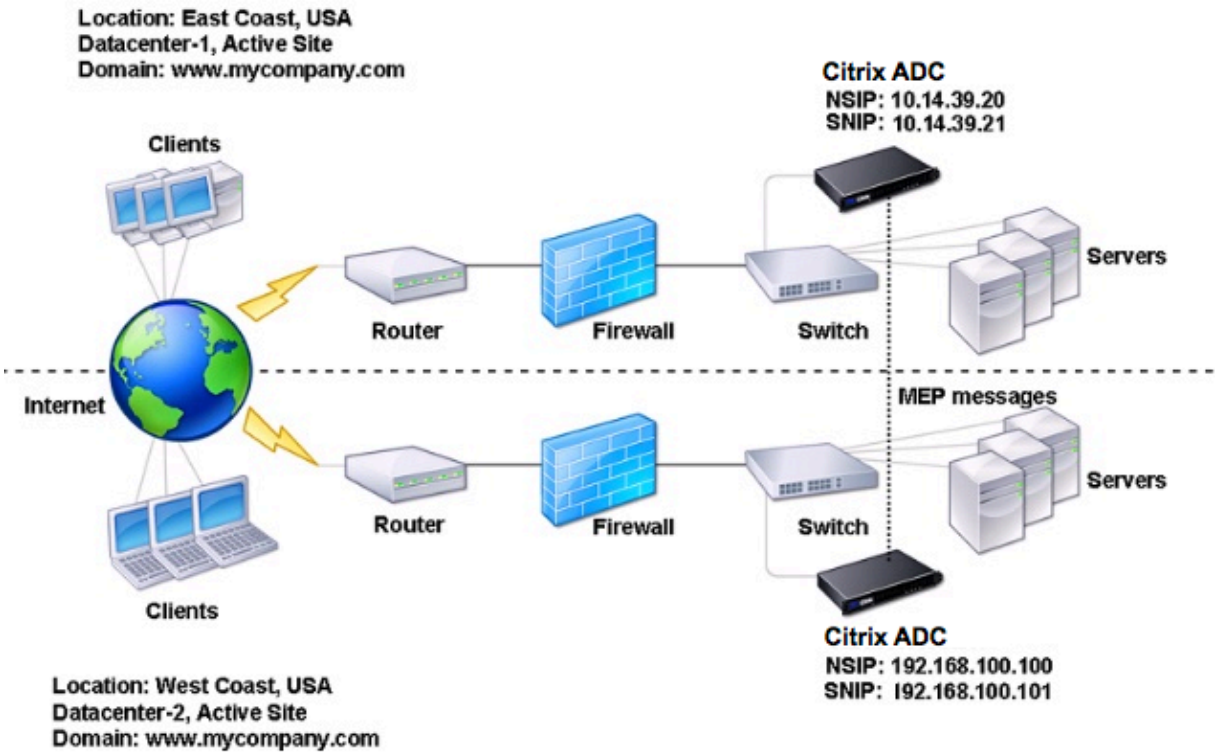
2. Click **Done** if the configuration is complete. The GSLB dashboard appears.
3. Click **Test GSLB Setup** to make sure that the ADNS services or the DNS servers are responding with the correct IP address for the domain name that is configured in the GSLB setup.

Configure GSLB entities individually

September 14, 2021

Global server load balancing is used to manage traffic flow to a web site hosted on two separate server farms that ideally are in different geographic locations. For example, consider a Web site, www.mycompany.com, which is hosted on two geographically separated server farms or data centers. Both server farms use Citrix ADC appliances. The Citrix ADC appliances in these server farms are set up in one-arm mode and function as authoritative DNS servers for the www.mycompany.com domain. The following figure illustrates this configuration.

Figure 1. Basic GSLB Topology



To configure such a GSLB setup, you must first configure a standard load balancing setup for each server farm or data center. This enables you to balance load across the different servers in each server farm. Then, configure both Citrix ADC appliances as authoritative DNS (ADNS) servers. Next, create a GSLB site for each server farm, configure GSLB virtual servers for each site, create GLSB services, and bind the GSLB services to the GSLB virtual servers. Finally, bind the domain to the GSLB virtual servers. The GSLB configurations on the two appliances at the two different sites are identical, although the load-balancing configurations for each site is specific to that site.

Note: To configure a GSLB site in a Citrix ADC cluster setup, see Setting Up GSLB in a Cluster.

Configuring a Standard Load Balancing Setup

A load balancing virtual server balances the load across different physical servers in the data center. These servers are represented as services on the Citrix ADC appliance, and the services are bound to the load balancing virtual server.

For details on configuring a basic load balancing setup, see Load Balancing.

Configure an authoritative DNS service

September 14, 2021

When you configure the Citrix ADC appliance as an authoritative DNS server, it accepts DNS requests from the client and responds with the IP address of the data center to which the client should send requests.

Note: For the Citrix ADC appliance to be authoritative, you must also create SOA and NS records. For more information about SOA and NS records, see Domain Name System.

To create an ADNS service by using the command line interface

At the command prompt, type the following commands to create an ADNS service and verify the configuration:

```
add service <name> <IP>@ ADNS <port>

show service <name>
<!--NeedCopy-->
```

Example:

```
add service Service-ADNS-1 10.14.39.21 ADNS 53

show service Service-ADNS-1
<!--NeedCopy-->
```

To modify an ADNS service by using the command line interface

At the command prompt, type the following command:

```
set service <name> <IPAddress> ADNS <port>
<!--NeedCopy-->
```

Example:

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

To remove an ADNS service by using the command line interface

At the command prompt, type the following command:


```
rm service <name>
<!--NeedCopy-->
```

Example:

```
rm service Service-ADNS-1
<!--NeedCopy-->
```

To configure an ADNS service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Add a new ADNS service, or select an existing service and edit its settings.

Configure a basic GSLB site

September 14, 2021

A GSLB site is a representation of a data center in your network and is a logical grouping of GSLB virtual servers, services, and other network entities. Typically, in a GSLB set up, there are many GSLB sites that are equipped to serve the same content to a client. These are usually geographically separated to ensure that the domain is active even if one site goes down completely. All of the sites in the GSLB configuration must be configured on every Citrix ADC appliance hosting a GSLB site. In other words, at each site, you configure the local GSLB site and each remote GSLB site.

Once GSLB sites are created for a domain, the Citrix ADC appliance sends client requests to the appropriate GSLB site as determined by the GSLB algorithms configured.

To create a GSLB site by using the command line interface

At the command prompt, type the following commands to create a GSLB site and verify the configuration:

```
add gslb site <siteName> <siteIPAddress>
show gslb site <siteName>
<!--NeedCopy-->
```

Example:

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
<!--NeedCopy-->
```


To modify or remove a GSLB Site by using the command line interface

- To modify a GSLB site, use the set gslb site command, which is just like using the add gslb site command, except that you enter the name of an existing GSLB Site.
- To unset a site parameter, use the unset gslb site command, followed by the siteName value and the name of the parameter to be reset to its default value.
- To remove a GSLB site, use the rm gslb site command, which accepts only the <name> argument.

To configure a basic GSLB site by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. Add a new GSLB site, or select an existing GSLB site and edit its settings.

To view the statistics of a GSLB site by using the command line interface

At the command prompt, type:

```
stat gslb site <siteName>
<!--NeedCopy-->
```

Example:

```
stat gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To view the statistics of a GSLB site by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. Select the GSLB site and click **Statistics**.

Configure a GSLB service

September 14, 2021

A GSLB service is a representation of a load balancing or content switching virtual server. A local GSLB service represents a local load balancing or content switching virtual server. A remote GSLB service represents a load balancing or content switching virtual server configured at one of the other sites in the GSLB setup. At each site in the GSLB setup, you can create one local GSLB service and any number of remote GSLB services.

> **Important**
>
> If the load balancing virtual server is either in a GSLB node itself or is in a child node (in parent-child deployment) and no monitors are bound to the GSLB service, then make sure the following:
>
> The GLSB service IP address, port number, and protocol match the virtual server that the service is representing. Else, the service state is marked as DOWN.

To create a GSLB service by using the command line interface

At the command prompt, type the following commands to create a GSLB service and verify the configuration:

```
add gslb service <serviceName> <serverName | IP> <serviceType> <port>-
   siteName <string>
show gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80  -siteName Site-
   GSLB-East-Coast
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

To modify or remove a GSLB service by using the command line interface

- To modify a GSLB service, use the set gslb service <serviceName> command. For this command, specify the name of the GSLB service whose configuration you want to modify. You can change the existing values of the parameters either specified by you or set by default. You can change the value of more than one parameter in the same command. Refer to the add gslb service command for details about the parameters. Example

  ```
  > set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -
     maxClient 8
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 25 kbits
  <!--NeedCopy-->
  ```

- To reset a parameter to its default value, you can use the unset gslb service <serviceName> command and the parameters to be unset. Example


```
> unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
Done
> sh gslb service SKP_GSLB_NOTCNAME_SVC2
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 0 kbits
<!--NeedCopy-->
```

- To remove a GSLB service, use the rm gslb service <serviceName> command.

To create a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Add a new GSLB service, or select an existing service and edit its settings.

To view the statistics of a GSLB service by using the command line interface

At the command prompt, type:

```
stat gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
stat gslb service Service-GSLB-1
<!--NeedCopy-->
```

To view the statistics of a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Select the GSLB Service and click **Statistics**.

Configure a GSLB service group

September 14, 2021

Service group enables you to manage a group of services as easily as a single service. For example, if you enable or disable an option, such as compression, health monitoring, or graceful shutdown, for a service group, the option gets enabled or disabled for all the members of the service group.

After creating a service group, you can bind it to a virtual server, and you can add services to the group. You can also bind monitors to the service groups.

> **Important**
>
> If the load balancing virtual server is either in a GSLB node itself or is in a child node (in parent-child deployment) and no monitors are bound to the GSLB service, then make sure the following:
>
> The GLSB service group IP address, port number, and protocol match the virtual server that the service is representing. Else, the service state is marked as DOWN.

The Citrix ADC supports the following types of GSLB service groups.

- IP address based service groups
- Domain name based service groups
- Domain name based autoscale service groups

GSLB domain name based autoscale service groups

The Citrix ADC hybrid and multi-cloud global server load balancing (GSLB) solution enables customers to distribute application traffic across multiple data centers in hybrid clouds, multiple clouds, and on premises. The Citrix ADC GSLB solution supports various load balancing solutions, such as the Citrix ADC load balancer, Elastic Load Balancing (ELB) for Amazon Web Services (AWS), and other third-party load balancers. Also, the GSLB solution performs global load balancing even if the GSLB and load balancing layers are independently managed.

In cloud deployments, users are given a domain name as a reference when accessing the load balancing solution for management purposes. It is recommended that external entities do not use the IP addresses that these domain names resolve to. Also, the load balancing layers scale up or down based on the load, and the IP addresses are not guaranteed to be static. Therefore, it is recommended to use the domain name to refer to the load balancing endpoints instead of IP addresses. This requires the GSLB services to be referred using the domain name instead of IP addresses and it must consume all the IP addresses returned for the load balancing layer domain name and have a representation for the same in GSLB.

To use domain names instead of IP addresses when referring to the load balancing endpoints, you can use the domain name based service groups for GSLB.

Monitor GSLB domain name based service groups

The Citrix ADC appliance has two built-in monitors that monitor TCP-based applications; tcp-default and ping-default. The tcp-default monitor is bound to all TCP services and the ping-default monitor is bound to all non-TCP services. The built-in monitors are bound by default to the GSLB service groups. However, it is recommended to bind an application specific monitor to the GSLB service groups.

Recommendation for setting the trigger monitors option to MEPDOWN

The Trigger Monitors option can be used to indicate if the GSLB site must use the monitors always, or use monitors when metrics exchange protocol (MEP) is DOWN.

The Trigger Monitors option is set to ALWAYS by default.

When the Trigger Monitors option is set to ALWAYS, each GSLB node triggers the monitors independently. If each GSLB node triggers the monitors independently, then each GSLB node might operate on different set of GSLB services. This might result in discrepancies in the DNS responses for the DNS requests landing on these GSLB nodes. Also, if each GSLB node is monitoring independently, then the number of monitor probes reaching the load balancer entity increases. The persistence entries also become incompatible across the GSLB nodes.

Therefore, it is recommended that the Trigger Monitors option on GSLB site entity is set to MEPDOWN. When the Trigger Monitors option is set to MEPDOWN, the load balancing domain resolution and monitoring ownership lies with the local GSLB node. When Trigger Monitors option is set to MEPDOWN, the load balancing domain resolution and subsequent monitoring is done by the local GSLB node of a GSLB service group. The results are then propagated to all other nodes participating in GSLB by using the metrics exchange protocol (MEP).

Also, whenever the set of IP addresses associated with a load balancing domain are updated, it is notified through MEP.

Limitations of GSLB service groups

- For a load balancing domain, the IP address that is returned in the DNS response is generally the public IP address. The private IP address cannot be applied dynamically when the load balancing domain is resolved. Therefore, public IP port and private IP port for the GSLB domain name based autoscale service groups IP port bindings are the same. These parameters cannot be set explicitly for the domain name based autoscale service groups.
- Site persistence, DNS views, and clustering are not supported for GSLB service groups.

Configure and manage GSLB service groups by using the CLI

Operation	CLI Command
To add a GSLB service group|`add gslb serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (DISABLED	DNS)] -siteName <string>`
	Example: **add gslb serviceGroup** Service-Group-1 http -siteName Site1 -autoScale DNS
To bind a GSLB service group to a virtual server|`bind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>)| <serviceName>@	((-monitorName <string>@`

||Example: **bind gslb serviceGroup** Service-Group-1 203.0.113.2; **bind gslb serviceGroup** Service-Group-1 S1 80; **bind gslb serviceGroup** Service-Group-1 -monitorName Mon1
|To unbind a GSLB service group to a virtual server|unbind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>)| <serverName>@ | -monitorName <string>@)
||Example:**unbind gslb serviceGroup** Service-Group-1 -monitorName Mon1
|To set parameters for a GSLB service group|set gslb serviceGroup <serviceGroupName>@ [(<serverName>@ <port> [-weight <positive_integer>] [-hashId <positive_integer>] [-publicIP <ip_addr|ipv6_addr|*>] [-publicPort <port>])| -maxClient <positive_integer> | -cip (ENABLED | DISABLED)| <cipHeader> | -cltTimeout <secs> | -svrTimeout <secs> | -maxBandwidth <positive_integer> | -monThreshold <positive_integer> | -downStateFlush (ENABLED | DISABLED)] [-monitorName <string> -weight <positive_integer>] [-healthMonitor (YES | NO)] [-comment <string>] [-appflowLog (ENABLED | DISABLED)]
|To unset parameters from a GSLB service group|unset gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port> [-weight] [-hashId] [-publicIP] [-publicPort]] [-maxClient] [-cip] [-cltTimeout] [-svrTimeout] [-maxBandwidth] [-monThreshold] [-appflowLog] [-monitorName] [-weight] [-healthMonitor] [-cipHeader] [-downStateFlush] [-comment]
|To enable a GSLB service group|enable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
||Example:**enable gslb serviceGroup** SG1 S1 80
|To disable a GSLB service group|“‘disable gslb serviceGroup @ [@] [-delay] [-graceFul (YES /| NO)]

```
||Example:**disable gslb serviceGroup** SRG2 S1 80
||**Note**: The service group that has to be disabled must be a DBS service group and not an autoscale service group.
|To remove a GSLB service group|``rm gslb serviceGroup <serviceGroupName>``
||Example:**rm gslb serviceGroup** Service-Group-1
|To view the statistics of a GSLB service group|``stat gslb serviceGroup [<serviceGroupName>]``
||Example:**stat gslb serviceGroup** Service-Group-1
|To view the properties of a GSLB service group|``show gslb serviceGroup [<serviceGroupName>  -includeMembers]``
||Example: **show gslb serviceGroup** SG1; **show gslb serviceGroup** -includeMembers

### Changes to the existing GSLB CLI commands

The following table lists some of the changes that are done to the existing GSLBs commands after the introduction of the GSLB service groups.
```



```
13
14 |CLI Command|Change
15 |--|--|
16 |bind gslb vserver|The service group name is added to the bind command.
17 ||Example:```bind gslb vserver <name> ((-serviceName <string> [-weight
      <positive_integer>] ) | <serviceGroupName>@ | | (-domainName <string
      > [-TTL <secs>] [-backupIP<ip_addr|ipv6_addr|*>] [-cookieDomain <
      string>] [-cookieTimeout <mins>][-sitedomainTTL <secs>]) | (-
      policyName <string>@ [-priority<positive_integer>] [-
      gotoPriorityExpression <expression>] [-type REQUEST | RESPONSE )]))
      <!--NeedCopy-->
```

|unbind gslb vserver|The service group is added to the unbind command.
|Example:|“‘unbind gslb vserver (-serviceName @ /(-domainName [-backupIP] [-cookieDomain]) | -policyName @)

```
1  |show gslb site|When this command is executed, the GSLB service groups
      are also displayed.
2  |show gslb vs|When this command is executed, the GSLB service groups
      are be displayed.
3  |stat gslb vs|When this command is executed, the GSLB service groups
      statistics are also displayed.
4  |show lb monitor bindings|When this command is executed the GSLB
      service group bindings are also displayed.
5
6  ### Configure GSLB service groups by using the GUI
7
8  1.  Navigate to **Traffic Management** > **GSLB** > **Service Groups**.
9  1.  Create a service group and set the AutoScale Mode to DNS.
10
11 ### Configure site persistence for the GSLB service groups
12
13 You can configure site persistence for the IP address based and domain
      name based service groups. Site persistence is not supported for
      domain name based autoscale service groups.
14
15 #### To set site persistence based on HTTP cookies by using the CLI
16
17 -  For connection proxy persistence, you do not have to set the site
      prefix.
18
19  At the command prompt, type:
```

set gslb service group [-sitePersistence]


```
- For HTTP redirect persistence, you must first set the site prefix
  for a member of the service group and then set the HTTPRedirect
  persistence parameter for the service group.

At the command prompt, type:
```

set gslb servicegroup <serviceGroup member name|Ip> [-sitePrefix]

set gslb servicegroup [-sitePersistence]
‘‘‘

Examples:

- Connection proxy persistence

  set gslbservicegroup sg1 -sitePersistence connectionProxy

- HTTPRedirect persistence

  set gslb servicegroup sg2 test1 80 -sitePrefix vserver-GSLB-1

  set gslb servicegroup sg2 -sitePersistence hTTPRedirect

To set site persistence based on cookies by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Services Groups** and select the service group that you want to configure for site persistence (for example, servicegroup-GSLB-1).
2. Click the **Site Persistence** section and set the persistence that meets your requirement.

> **Tip**
>
> For deployment scenario and example configuration of GSLB service groups, see the following topics:
>
> - Use Case: Deployment of Domain Name Based Autoscale Service Group
> - Use Case: Deployment of IP Address Based Autoscale Service Group

Configure a GSLB virtual server

September 14, 2021

A GSLB virtual server is an entity that represents one or more GSLB services and balances traffic between them. It evaluates the configured GSLB methods or algorithms to select a GSLB service to which to send the client request.

> **Note**
>
> A GSLB virtual server protocol requirement is mainly to create a relation between the virtual server and the services that are bound to the virtual server. This also keeps CLI/APIs consistent for other types of virtual servers. The Service Type parameter on a service or a virtual server is not used while serving the DNS requests. It is instead referenced during site persistence, monitoring, and for doing lookups via MEP.

To create a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to add a GSLB virtual server and verify the configuration:

```
1 -  add gslb vserver <name> <serviceType> -ipType (IPv4 | IPv6)
2 -  show gslb vserver <name>
3 <!--NeedCopy-->
```

Example:

```
1 add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
2 add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
3 show gslb vserver Vserver-GSLB-1
4 show gslb vserver Vserver-GSLB-2
5 <!--NeedCopy-->
```

To modify or remove a GSLB virtual server by using the command line interface

- To modify a GSLB virtual server, use the `set gslb vserver` command. This command works similar to the `add gslb vserver` command, except that you enter the name of an existing GSLB virtual server.
- To reset a parameter to is default value, you can use the `unset gslb vserver` command followed by the vserverName value and the name of the parameter to be unset.
- To remove a GSLB virtual server, use the `rm gslb vserver` command, which accepts only the name argument.

To configure a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. Add a new GSLB virtual server, or select an existing GSLB virtual server and edit its settings.

To view the statistics of a GSLB virtual server by using the command line interface

At the command prompt, type:

```
stat gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
stat gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To view the statistics of a GSLB virtual server by using the configuration utility

Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**, select the virtual server and click **Statistics**.

GSLB virtual server statistics

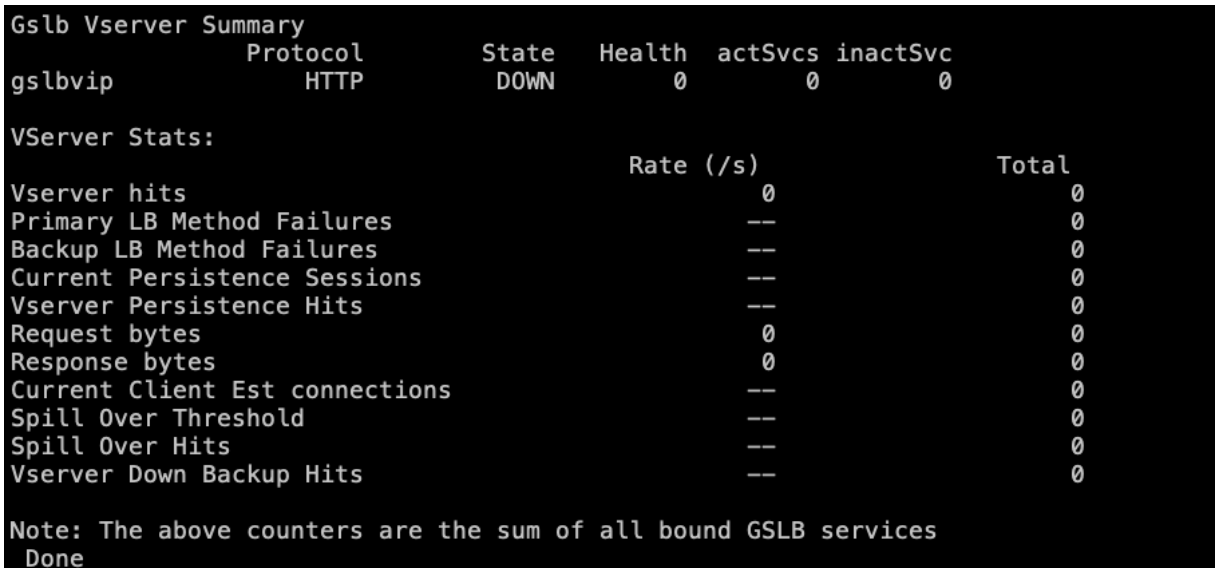
Starting from Citrix ADC version 12.1 build 51.xx and later, the GSLB virtual server statistics also display the following information in addition to details such as; vserver hits, current persistence session, request bytes, response bytes, spillover threshold, spillover hits, current client established connections, and vserver down backup hits.

- **Primary LB method failures**: Number of times the primary GSLB method has failed.
- **Backup LB method failures**: Number of times the backup GSLB method has failed.
- **Vserver persistence hits**: The number of times the request is served through the persistence sessions.

The GSLB virtual server statistics also display the statistics of the service group members bound to the virtual server.

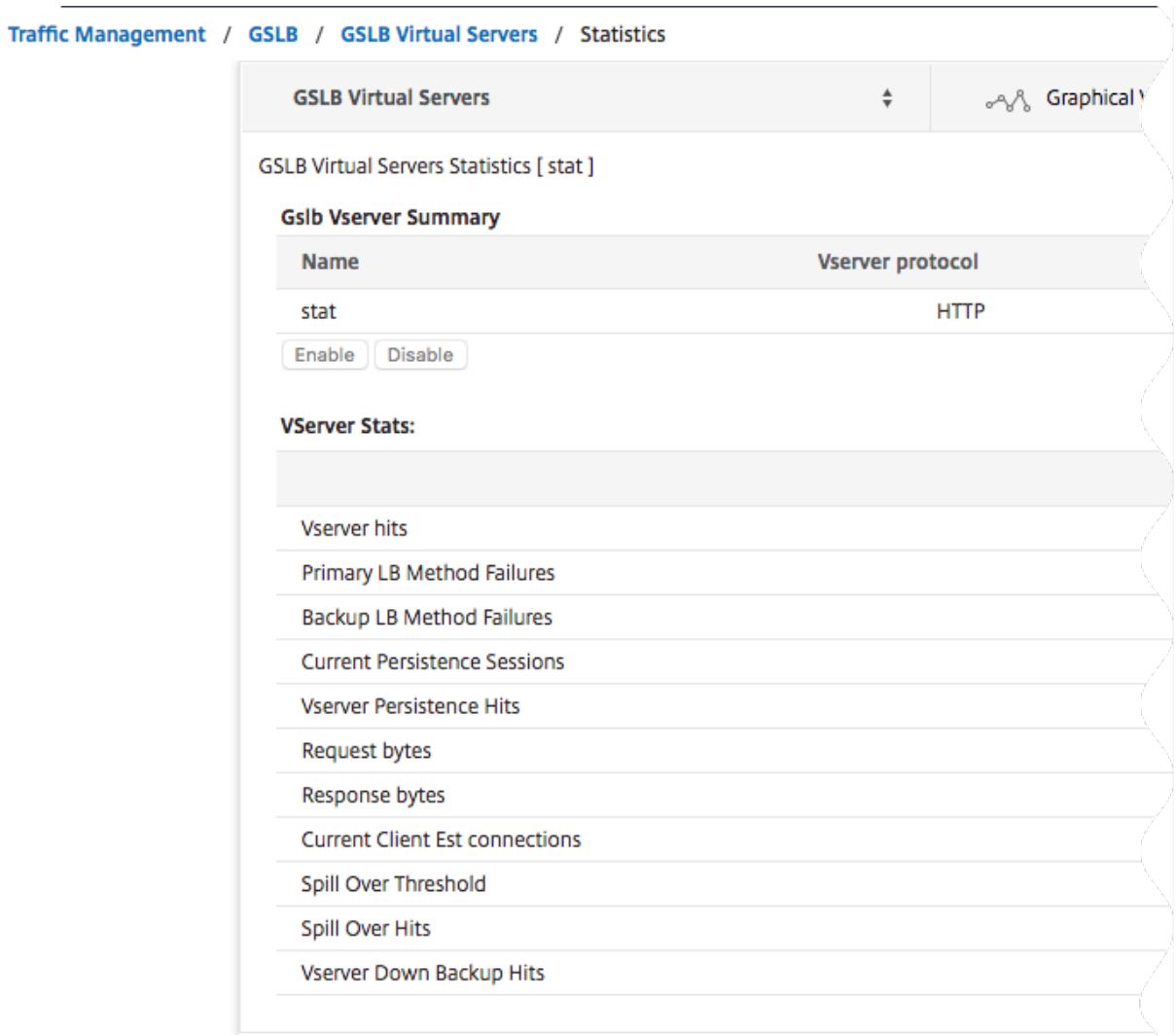
> **Note:**
> The primary or the backup method can fail when the primary method is static proximity and the backup method is RTT. In this scenario, if there is no location corresponding to LDNS IP, the static proximity fails and backup method is attempted. The statistics are updated based on the following:
>
> - If the backup method is successful, only primary method failure statistics are incremented.
> - If RTT calculation is not successful, then backup method also fails. In this case, both primary and backup method failure statistics are incremented.
>
> When the backup method fails, the last resort method of round robin is used.

The following image is a sample of GSLB virtual server statistics from CLI.



```
Gslb Vserver Summary
                Protocol        State   Health  actSvcs inactSvc
gslbvip             HTTP         DOWN        0        0        0

VServer Stats:
                                        Rate (/s)               Total
Vserver hits                                 0                      0
Primary LB Method Failures                  --                      0
Backup LB Method Failures                   --                      0
Current Persistence Sessions                --                      0
Vserver Persistence Hits                    --                      0
Request bytes                                0                      0
Response bytes                               0                      0
Current Client Est connections              --                      0
Spill Over Threshold                        --                      0
Spill Over Hits                             --                      0
Vserver Down Backup Hits                    --                      0

Note: The above counters are the sum of all bound GSLB services
 Done
```

The following image is a sample of GSLB virtual server statistics from GUI.



GSLB service statistics

When you run the stat gslb service command from the command line or click the **Statistics link** from the configuration utility, the following details of the service are displayed:

- **Request bytes.** Total number of request bytes received on this service or virtual server.
- **Response bytes.** Number of response bytes received by this service or virtual server.
- **Current client established connections.** Number of client connections in ESTABLISHED state.
- **Current load on the service.** Load on the service (Calculated from the load monitor bound to the service).

The data of number of requests and responses, and the number of current client and server connections may not be displayed or may not be synchronized with the data of the corresponding load balancing virtual server.

Clearing the GSLB virtual server or service statistics

Note: This feature is available in NetScaler release 10.5.e.

You can now clear the statistics of a GSLB virtual server and service. Citrix ADC provides the following two options to clear the statistics:

- **Basic**: Clears the statistics that are specific to the virtual server but retains the statistics that are contributed by the bound GLSB service.
- **Full**: Clears both the virtual server and the bound GSLB service statistics.

To clear the statistics of a GSLB virtual server by using the command line interface

At the command prompt, type:

```
stat gslb vserver <name> -clearstats <basic | full>
<!--NeedCopy-->
```

Example:

```
stat gslb vserver Vserver-GSLB-1 -clearstats basic
<!--NeedCopy-->
```

To clear the statistics of a GSLB service by using the command line interface

At the command prompt, type:

```
stat gslb service <name> -clearstats <basic | full>
<!--NeedCopy-->
```

Example:

```
stat gslb service service-GSLB-1 -clearstats basic
<!--NeedCopy-->
```

To clear the statistics of a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. Select the GSLB virtual server and, click **Statistics**, and then click **Clear**.
3. From the **Clear** drop-down list, select **Basic** or **Full**, and then click **OK**.

To clear the statistics of a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Select the GSLB service and, click **Statistics**, and then click **Clear**.
3. From the **Clear** drop-down list, select **Basic** or **Full**, and then click **OK**.

Enabling and Disabling GSLB Virtual Servers

When you create a GSLB virtual server, it is enabled by default. If you disable the GSLB virtual server, upon receiving a DNS request, the Citrix ADC appliance does not make any GSLB decision based on the GSLB method that is configured. Instead, the response to the DNS query contains the IP addresses of all the services bound to the virtual server.

To enable or disable a GSLB virtual server by using the command line interface

At the command prompt, type one of the following commands:

```
enable gslb vserver <name>@

disable gslb vserver <name>@
<!--NeedCopy-->
```

Example:

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To enable or disable a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. Select a virtual server and, from the **Action** list, select **enable** or **disable**.

Bind GSLB services to a GSLB virtual server

September 14, 2021

Once the GSLB services and virtual server are configured, relevant GSLB services must be bound to the GSLB virtual server to activate the configuration.

To bind a GSLB service to a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to bind a GSLB service to a GSLB virtual server and verify the configuration:

```
bind gslb vserver <name> -serviceName <string>

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To unbind a GSLB service from a GSLB virtual server by using the command line interface

At the command prompt, type:

```
unbind gslb vserver <name> -serviceName <string>
<!--NeedCopy-->
```

To bind GSLB services by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click a virtual server.
2. Click in the **Domains** section, and configure a domain and bind the domain.

Bind a domain to a GSLB virtual server

September 14, 2021

To make a Citrix ADC appliance the authoritative DNS server for a domain, you must bind the domain to the GSLB virtual server. When you bind a domain to a GSLB virtual server, the Citrix ADC appliance adds an address record for the domain, containing the name of the GSLB virtual server. The start of authority (SOA) and name server (NS) records for the GSLB domain must be added manually.

For details on configuring SOA and NS records, see Domain Name System.

To bind a domain to a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to bind a domain to a GSLB virtual server and verify the configuration:

```
bind gslb vserver <name> -domainName <string>
show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To unbind a GSLB domain from a GSLB virtual server by using the command line interface

At the command prompt, type:

```
unbind gslb vserver <name> -domainName <string>
<!--NeedCopy-->
```

To bind a domain to a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. In GSLB Virtual Servers pane, select the GSLB Virtual Server to which you want to bind the domain (for example, Vserver-GSLB-1) and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Domains tab, do one of the following:
   - To create a new Domain, click **Add**.
   - To modify an existing Domain, select the domain, and then click **Open**.
4. In the Create GSLB Domain or Configure GSLB Domain dialog box, specify values for the following parameters as shown:
   - Domain Name*—domainName (for example, www.mycompany.com)

   * A required parameter
5. Click Create.
6. Click OK.

To view the statistics of a domain by using the command line interface

At the command prompt, type:

```
stat gslb domain <name>
<!--NeedCopy-->
```

Example:

```
stat gslb domain www.mycompany.com
<!--NeedCopy-->
```

Note: To view statistics for a particular GSLB domain, enter the name of the domain exactly as it was added to the Citrix ADC appliance. If you do not specify the domain name, or if you specify an incorrect domain name, statistics for all configured GSLB domains are displayed.

To view the statistics of a domain by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. In GSLB Virtual Servers pane, select the GSLB Virtual Server (for example, Vserver-GSLB-1) and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Domains tab, select the domain, and then click **Statistic**s.

To view the configuration details of the entities bound to a GSLB domain using the command line

Note: This feature is available in NetScaler release 10.5.e.

At the command prompt, type:

```
show gslb domain <name>
<!--NeedCopy-->
```

Example:

```
show gslb domain gslb1.com
       gslb1.com
       gvs1 - HTTP    state: DOWN
       DNS Record Type: A
       Configured Method: LEASTCONNECTION
       Backup Method: ROUNDROBIN
       Persistence Type: NONE
       Empty Down Response: DISABLED
```



```
9         Multi IP Response: DISABLED
10        Dynamic Weights: DISABLED
11
12        gsvc1 (10.102.239.165: 80)- HTTP State: DOWN    Weight: 1
13               Dynamic Weight: 0    Cumulative Weight: 1
14               Effective State: DOWN
15               Threshold : BELOW
16
17               Monitor Name : http
18                      State: DOWN   Weight: 1
19                      Probes: 144   Failed [Total: 144 Current: 144]
20                      Last response: Failure - TCP syn sent, reset
                           received.
21                      Response Time: 2000 millisec
22
23        gsvc2 (10.102.239.179: 80)- HTTP State: DOWN    Weight: 1
24               Dynamic Weight: 0    Cumulative Weight: 1
25               Effective State: DOWN
26               Threshold : BELOW
27
28               Monitor Name : http-ecv
29                      State: DOWN   Weight: 1
30                      Probes: 141   Failed [Total: 141 Current: 141]
31                      Last response: Failure - TCP syn sent, reset
                           received.
32                      Response Time: 2000 millisec
33 Done
34 <!--NeedCopy-->
```

To view the configuration details of the entities bound to a GSLB domain by using the configuration utility

Note: This feature is available in NetScaler release 10.5.e.

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click a virtual server.
2. Click the field below the **Domains** pane.
3. In the **GSLB Virtual Server Domain Binding** dialog box, select a domain, and then click **Show Bindings**.

Example of a GSLB setup and configuration

September 14, 2021

An organization has a geographically dispersed network and has three data centers located in the United States, Mexico, and Colombia. In the configuration related to these locations, these are referred to as US, MX, and CO respectively. At each location, the company has a server farm, which provides the same content and the setup is working as expected. The Citrix ADC appliance at each location is configured through a virtual server with the HTTP protocol on port 80.

The organization has implemented the GSLB setup by adding a site identifier at each site. The site identifier includes a site name and an IP address that is owned by the Citrix ADC appliance and is used for the GSLB communications.
Each site has a site local to the appliance. Also, each site has two sites remote to the local appliance. On each site, a GSLB virtual server is created with the same name. This virtual server identifies the website of the organization globally and does not have any IP address associated with it.
The setup also has GSLB services configured that point to the load balancing virtual servers configured on each GSLB site by specifying the IP address, protocol, and port number of the respective virtual server. These services are bound to the GSLB virtual server.

Note: In the procedure below, the commands use private IP addresses for the GSLB sites. For public sites and GSLB services, ensure that you use public IP addresses for these sites.

The following table lists the IP addresses and locations used in the example:

IP Address	Location
10.3.1.101	Site IP of local Citrix ADC.
172.16.1.101	Site IP of remote location site-MX.
192.168.1.101	Site IP of remote location site-CO.
172.16.1.100	Service IP of remote location site-MX.
10.3.1.100	Service IP of local Citrix ADC.
192.168.1.100	Service IP of remote location site-CO.

When adding a GSLB site, if the site communicates over the internet only then use the “Public IP” field. For example, when there is no site to site VPN connectivity between the GSLB sites.

To configure the GSLB setup with Citrix ADC appliances by using the CLI commands

1. Enable the GSLB feature, if not already done.


```
enable ns feature gslb
<!--NeedCopy-->
```

2. Identify a SNIP that for adding local GSLB site.
3. Add the GSLB site for the local Citrix ADC appliance.

```
add gslb site site-US 10.3.1.101
<!--NeedCopy-->
```

4. Add the GSLB sites for the remote Citrix ADC appliances.

```
add gslb site site-MX 172.16.1.101
add gslb site site-CO 192.168.1.101
<!--NeedCopy-->
```

5. Add the GSLB virtual server that references a service being used in the GSLB setup:

```
add gslb vserver gslb-lb HTTP
<!--NeedCopy-->
```

6. Add the GSLB services for each site participating in the GSLB setup:

```
add gslb service gslb_SVC30 172.16.1.100 HTTP 80 -siteName site-MX
add gslb service gslb_SVC10 10.3.1.100 HTTP 80 -siteName site-US
add gslb service gslb_SVC20 192.168.1.100 HTTP 80 -siteName site-
   CO
<!--NeedCopy-->
```

7. Bind the GSLB services to the GSLB virtual server:

```
bind gslb vserver gslb-lb -serviceName gslb_SVC10
bind gslb vserver gslb-lb -serviceName gslb_SVC20
bind gslb vserver gslb-lb -serviceName gslb_SVC30
<!--NeedCopy-->
```

8. Bind the domain to the GSLB virtual server.

```
bind gslb vserver gslb-lb -domainName www.mycompany.com -TTL 30
<!--NeedCopy-->
```

9. Add an ADNS service that listens to the DNS queries.

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```


Synchronize the configuration in a GSLB setup

September 14, 2021

Typically, a GSLB setup has a few data centers with a GSLB site configured for each data center. In each Citrix ADC, participating in GSLB, configure one GSLB site as a local site and the others as remote sites. When you add another GSLB site at a later point, you must ensure that the configuration across all GSLB sites is identical. You can use the Citrix ADC's GSLB configuration synchronization option to synchronize the configuration across the GSLB sites.

The Citrix ADC appliance from which you use the synchronization option is referred to as the 'main site' and the GSLB sites on which the configuration is copied as the 'subordinate sites'. When you synchronize a GSLB configuration, the configurations on all the GSLB sites participating in the GSLB setup are made similar to the configuration on the main site.

Synchronization is done only on the parent sites. Synchronization does not affect GSLB child sites' configuration. This is because the parent site and the child site configurations are not identical. The child sites configuration consists only of its own and its parent site's details. Also, GSLB services are not always required to be configured in the child sites.

- The main node finds the differences between the configuration of the main node and subordinate node, and changes the configuration of the subordinate node to make it similar to the main node.

  If you force a synchronization (use the 'force sync' option), the appliance deletes the GSLB configuration from the subordinate node and then configures the subordinate to make it similar to the main node.

- During synchronization, if a command fails, synchronization is not aborted and the error message are logged into a **.err** file in the **/var/netscaler/gslb** directory.
- Synchronization is done only on the parent sites. Synchronization does not affect the GSLB child sites' configuration. This is because the parent site and the child site configurations are not identical. The child sites configuration consists only of its own and its parent site's details. Also, GSLB services are not always required to be configured in the child sites.
- If you disable the internal user login, the GSLB auto sync uses the SSH keys to synchronize the configuration. But, to use GSLB auto sync in the partition environment, you must enable the internal user login and make sure that the partition user name in the local and remote GSLB sites is the same.

> **Note**
>
> - On the remote GSLB site RPC node, configure the firewall to accept auto-sync connections by specifying the remote site IP (cluster IP address for cluster setup) and port (3010 for RPC

> and 3008 for secure RPC). If the default route to reach the remote sites is in the management subnet, as in most cases, then NSIP is used as the source IP address.
>
> To configure a different source IP address, you must have the GSLB site IP address and the SNIP in a different subnet. Also, you must have an explicit route defined to the remote site IP address through a GSLB site IP subnet.
>
> For enhanced security, Citrix recommends that you change the internal user account and RPC node passwords. Internal user account password is changed through RPC node password. For details, see Change an RPC node password.

If you use the saveconfig option, the sites that participate in the synchronization process automatically save their configuration, in the following way:

On the remote GSLB site RPC node, configure the firewall to accept auto-sync connections by specifying the remote site IP (cluster IP address for cluster setup) and port (3010 for RPC and 3008 for secure RPC). If the default route to reach the remote sites is in a management subnet, as in most cases, then NSIP is used as the source IP address.

To configure a different source IP address, you must have the GSLB site IP address and the SNIP in a different subnet. Also, you must have an explicit route defined to the remote site IP address through the GSLB site IP subnet. The source IP address cannot be synchronized across the sites participating in GSLB because the source IP address for an RPC node is specific to each Citrix ADC appliance. Therefore, after you force a synchronization (using the sync gslb config -forceSync command or by selecting the ForceSync option in the GUI), you have to manually change the source IP addresses on the other Citrix ADC appliances. Port 22 is also required for synchronizing the database files to the remote site.

To improve the time taken for configuration synchronization on all GSLB sites

Configure the TCP profile settings at the command prompt as follows:

```
1 set tcpprofile nstcp_internal_apps -bufferSize 4194304 -sendBuffsize
     4194304 -tcpmode ENDPOINT
2 <!--NeedCopy-->
```

Limitations of synchronization

- On the main site, the names of the remote GSLB sites must be identical to the names of sites configured on the Citrix ADC appliances hosting those sites.
- During the synchronization, traffic disruptions may occur.
- Citrix ADC is tested to synchronize up to 200,000 lines of the configuration.
- Synchronization may fail:
  - If the spill over method is changed from CONNECTION to DYNAMIC CONNECTION.

  - If you interchange the site prefix of the GSLB services bound to a GSLB virtual server on the main node and then try to synchronize.
  - If the RPC node passwords are different for NSIP and loopback IP address.
  - If you perform synchronization on GSLB sites that are configured in different partitions of the same Citrix ADC appliance.
- If you have configured the GSLB sites as High Availability (HA) pairs, the RPC node passwords of primary and secondary nodes must be the same.
- If you rename any GLSB entity that is part of your GSLB configuration (use the "show gslb runningConfig" command to display the GSLB configuration). You must use the force sync option to synchronize the configuration to other GSLB sites.

> **Note:**
>
> - In incremental synchronization, you don't have to use the force sync option to synchronize the configuration to other GSLB sites. This is applicable starting Citrix ADC release 13.0 build 79.x onwards.

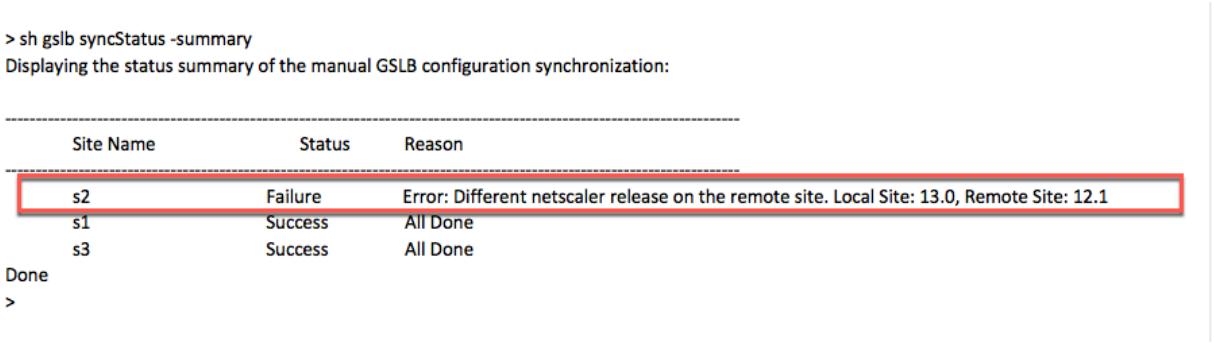
Note: To overcome the limitations due to some settings in the GSLB configuration, you can use the force sync option. But, if you use the force sync option the GSLB entities are removed and readded to the configuration and the GSLB statistics are reset to zero. Hence the traffic is disrupted during the configuration change.

Points to note before starting the synchronization of a GSLB setup

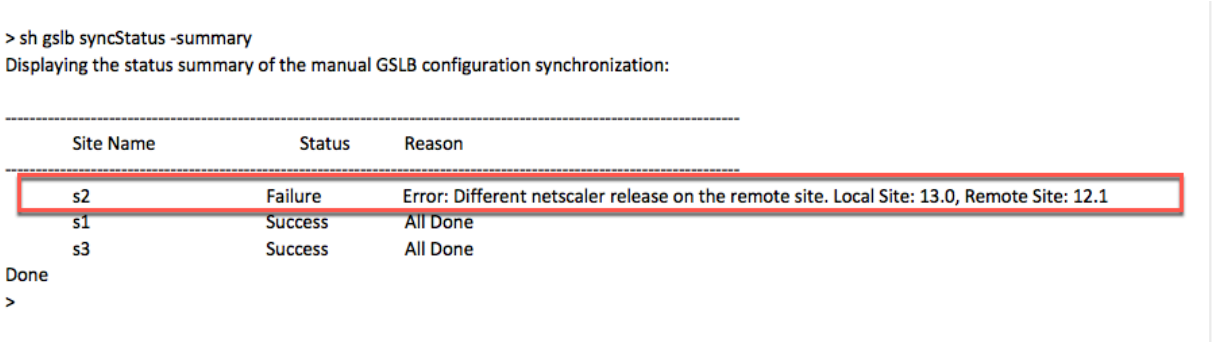
Before you start the synchronization of a GSLB setup, make sure that:

- On all the GSLB sites including the main site, management access and SSH must be enabled for the IP address of the corresponding GSLB site. The IP address of a GSLB site must be an IP address owned by the Citrix ADC appliance. For more information about adding the GSLB site IP addresses and enabling Management Access, see "Configuring a Basic GSLB Site".
- The GSLB configuration on the Citrix ADC appliance that is considered as the main site is complete and appropriate to be copied on all the sites.
- If you are synchronizing the GSLB configuration for the first time, all the sites participating in GSLB must have the GSLB site entity of their respective local sites.
- You are not synchronizing sites that, by design, do not have the same configuration.
- The main site and the subordinate sites run the same Citrix ADC versions. Starting from release 12.1, build 50.x, the appliance checks for the firmware version on main and subordinate sites before initiating synchronization. If the main and the subordinate sites run different versions, the synchronization is aborted for that remote site to avoid pushing any incompatible changes across the versions. Also, an error message displaying the site details on which the synchronization aborted appears.

The following figures display sample error messages from the CLI and the GUI.



```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status          Reason
-------------------------------------------------------------------------------------------------------------
        s2                       Failure         Error: Different netscaler release on the remote site. Local Site: 13.0, Remote Site: 12.1
        s1                       Success         All Done
        s3                       Success         All Done
Done
>
```



```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status          Reason
-------------------------------------------------------------------------------------------------------------
        s2                       Failure         Error: Different netscaler release on the remote site. Local Site: 13.0, Remote Site: 12.1
        s1                       Success         All Done
        s3                       Success         All Done
Done
>
```

> **Important**
>
> The following directories are synchronized as part of the GSLB configuration synchronization.
>
> - /var/netscaler/locdb/
> - /var/netscaler/ssl/
> - /var/netscaler/inbuilt_db/

Manual synchronization between sites participating in GSLB

September 14, 2021

The manual synchronization of GSLB configuration across the master site and the slave sites is performed in the following manner:

- The master site detects the differences between the configuration of its own site and the slave site.
- The master site applies the difference in configuration to the slave site.
- The master site performs the configuration synchronization with all the slave sites in the GSLB setup, and completes the synchronization process.

Important: After a GSLB configuration is synchronized, the configuration cannot be rolled back on any of the GSLB sites. Perform the synchronization only if you are sure that the synchronization process

does not overwrite the configuration on the remote site. Site synchronization is undesirable when the local and remote sites have different configurations by design, which leads to site outage. If some commands fail and some commands succeed, the successful commands are not rolled back.

Points to note

- If you force a synchronization (use the 'force sync' option), the Citrix ADC appliance deletes the GSLB configuration from the slave site. Then, master site configures the slave site to make it similar to its own site.
- During synchronization, if a command fails, synchronization is not aborted. The error messages are logged into an .err file in the /var/netscaler/gslb directory.
- If you use the `saveconfig` option, the sites participating in the synchronization process automatically save their configuration, in the following way:
  - The master site saves its configuration immediately before it initiates the synchronization process.
  - The slave sites save their configuration after the process of synchronization is complete. A slave site saves its configuration only if the configuration difference was applied successfully on it. If synchronization fails on a slave site, you must manually investigate the cause of the failure and take corrective actions.

To synchronize a GSLB configuration by using the CLI:

At the command prompt, type the following commands to synchronize GSLB sites and verify the configuration:

```
sync gslb config [-preview | -forceSync <string> | -nowarn | -
   saveconfig] [-debug]
show gslb syncStatus
<!--NeedCopy-->
```

Example:

```
sync gslb config

[WARNING]: Syncing config may cause configuration loss on other site.

Please confirm whether you want to sync-config (Y/N)? [N]:y

Sync Time: Dec 9 2011 10:56:9

Retrieving local site info: ok

Retrieving all participating gslb sites info: ok

```



```
Gslb_site1[Master]:

Getting Config: ok

Gslb_site2[Slave]:

Getting Config: ok

Comparing config: ok

Applying changes: ok

Done
<!--NeedCopy-->
```

To synchronize a GSLB configuration by using the GUI:

1. Navigate to **Traffic Management > GSLB > Dashboard**.
2. Click **Auto Synchronization GSLB** and select **ForceSyn**.
3. In **GSLB Site Name**, select the GSLB sites that are to be synchronized with the master node configuration.

Previewing GSLB synchronization

By previewing the GSLB synchronization operation, you can see the differences between the master node and each slave node. If there are any discrepancies, you can troubleshoot before synchronizing the GSLB configuration.

To preview the GSLB synchronization output by using the CLI:

At the command prompt, type the following command:

```
sync gslb config -preview
<!--NeedCopy-->
```

To preview the GSLB synchronization output by using the GUI:

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Dashboard**.
2. Click **Auto Synchronization GSLB** and select **Preview**.
3. Click **Run**.
   A progress window displays any discrepancies in the configuration.

Debugging the commands triggered during synchronization process

You can view the status (success or failure) of each command triggered during the synchronization process and troubleshoot accordingly.

To debug the GSLB synchronization commands by using the CLI:

At the command prompt, type the following command:

```
sync gslb config -debug
<!--NeedCopy-->
```

To debug the GSLB synchronization commands by using the GUI:

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Dashboard.**
2. Click **Auto Synchronization GSLB** and select **Debug**.
3. Click **Run**. A progress window displays the status of each command triggered during synchronization.

Real-time synchronization between sites participating in GSLB

September 14, 2021

You can use the “AutomaticConfigSync” option to automatically synchronize the real-time GSLB configuration of main site to all the subordinate sites. You do not have to manually trigger the AutoSync option to synchronize the configuration.

You can automatically synchronize the GSLB configuration of main site to all the subordinate sites by using incremental synchronization or full synchronization. The “GSLBSyncMode” parameter allows you to choose the synchronization mode.

> Note:
>
> Starting from Citrix ADC release 13.0 build 79.x, incremental synchronization of GSLB synchronization is supported. By default, the synchronization is performed using incremental synchronization. Incremental synchronization can be performed by enabling the “IncrementalSync” parameter. For details, see Incremental synchronization of GSLB configuration.

Best practices for using the real-time synchronization feature

- It is recommended that all the Citrix ADC appliances participating as sites have the sameCitrix ADC software version.
- To change the RPC node password, first change the password on the subordinate site and then on the main site.

- Configure local GSLB sites on each site participating in GSLB.
- Enable automaticConfigSync on one of the sites where the configuration is performed. This site eventually gets synchronized to other GSLB sites.
- If there is a new configuration or changes are made to the existing configuration, make sure to check the status using the “show gslb syncStatus” command to confirm if the changes are synchronized across all sites or if there was any error.
- RSYNC port monitoring must be enabled.

To enable real-time synchronization by using the CLI

At the command prompt, type:

```
set gslb parameter -automaticConfigSync (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set gslb parameter -automaticConfigSync ENABLED
<!--NeedCopy-->
```

To enable real-time synchronization by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Change GSLB Settings**.
2. Select **Automatic ConfigSync**.

   Note: This option must be enabled only in the site where the configuration is performed.

For information on the following topics, see Manual synchronization between sites participating in GSLB.

- Previewing GSLB synchronization
- Debugging the commands triggered during synchronization process

Points to note

- The consolidated log file related to the real-time synchronization is stored in the /var/netscaler/gslb/periodic_sync.log directory.
- The default configuration file is stored in the /var/netscaler/gslb_sync/ directory.
- The main site uses the following directory structure:
  - The main site stores all its files in the /var/netscaler/gslb_sync/master directory.
  - The main site stores its configuration file that must be synced to the subordinate sites, in the /var/netscaler/gslb_sync/master/gslbconf/ directory.

  - The status files pulled from all the subordinate sites are stored in the /var/netscaler/gslb_sync/master/sl directory.
- The subordinate site uses the following directory structure:
  - The subordinate site picks up the latest configuration file to be applied from the /var/netscaler/gslb_sync/slave/gslbconf directory.
  - The subordinate site stores its status file in the /var/netscaler/gslb_sync/slave/gslbstatus directory.
- In an admin partition setup, the same directory structure is maintained in the location: /var/-partitions/`partition name`/netscaler/gslb_sync.
- The clocks on all the sites must be set accurately to a real-time standard like Coordinated Universal Time (UTC).

Incremental synchronization of GSLB configuration

The automatic GSLB configuration synchronization feature checks for the configuration changes on the main site at the interval of every 10 seconds, and performs a synchronization. This sync interval value is configurable.

In incremental synchronization, only the configurations that have changed on the main site between the last synchronization and the subsequent sync interval (10 secs) is synchronized across all the subordinate sites. Incremental synchronization is the default behavior. Pushing only the incremental configurations considerably reduces the configuration file size, and thus the synchronization time. If an incremental synchronization fails, the system automatically performs a full configuration synchronization.

Incremental synchronization is performed in the following manner:

- The main site pushes the configuration file comprising of only its latest changes to all the subordinate sites. The latest change refers to the configurations that has changed between the last synchronization and the subsequent sync interval (10 secs).
- Each subordinate site applies the latest change to its own site.
- Incremental synchronization in not attempted on the subordinate sites, which are in DOWN state. When the site comes back UP, again synchronization is performed.
- The subordinate site generates status logs at each step and copies them to a file at a specific location.
- The main site pulls the status log files from the specified location.
- The main site prepares a log file with logs combined from all the subordinate sites.
- This combined log file is stored in /var/netscaler/gslb/periodic_sync.log file.

For more information on the directories where the configuration files are stored, see “Points to note” section.

To enable GSLB configuration incremental synchronization by using the CLI

```
set gslb parameter -AutomaticConfigSync (ENABLED | DISABLED) -
    GSLBSyncMode (IncrementalSync | FullSync) -GslbConfigSyncMonitor (
    ENABLED | DISABLED) -GSLBSyncInterval <secs> -GSLBSyncLocFiles (
    ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
    IncrementalSync
<!--NeedCopy-->
```

The incremental synchronization provides the following configurable parameters, which reduce the overall time taken to synchronize the GSLB configuration.

- **GslbConfigSyncMonitor**—Enable the GSLB Config Sync Monitor parameter to monitor the state of the subordinate sites' RSYNC port, which is the SSH port 22 on remote GSLB site IP address. If the monitor shows the subordinate site state as DOWN, the RSYNC operation to that site is skipped. This reduces the synchronization delays caused by attempting to connect to the remote sites that are DOWN.

  Example to enable RSYNC port monitoring in CLI:

  ```
  set gslb parameter -GSLBSyncMode IncrementalSync -
      GslbConfigSyncMonitor ENABLED
  <!--NeedCopy-->
  ```

- **GSLBSyncInterval**—Set the time interval (in seconds) at which the GSLB configuration synchronization occurs. By default, the automatic GSLB configuration sync feature synchronizes the GSLB configuration automatically at every 10 seconds. You can change the time interval to any desired value. Refrain from setting this to a lower value, for example, not lesser than 5 seconds. Because, synchronizing frequently might increase the management CPU consumption.

  > Note:
  >
  > In an admin partition setup, the time interval can be set only in the default partition because it is a global parameter.

  Example to set the sync interval:

  ```
  set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
      IncrementalSync -GSLBSyncInterval 7
  <!--NeedCopy-->
  ```


- **GSLBSyncLocFiles**—During GSLB config synchronization, by default, the changes in the location DB files are detected and automatically synchronized. Because the location DB directories do not change often, admins can disable automatically synchronizing the location DB files. Instead, admins must manually copy the location DB files to the GSLB subordinate sites. Synchronizing location DB files takes much time. Thus, avoiding it reduces the overall synchronization time.

  Example to disable automatically synchronizing the location DB files:

```
set gslb parameter -GSLBSyncMode IncrementalSync -
   GSLBSyncLocFiles DISABLED
<!--NeedCopy-->
```

To enable GSLB incremental synchronization by using the GUI

1. Navigate to **Traffic Management > GSLB > Dashboard > Change GSLB settings**.
2. In the **Set GSLB Parameters** page, you can perform the following:
   - To enable incremental synchronization, choose **IncrementalSync** from the **GSLB Sync Mode** drop-down menu.
   - To set the Automatic GSLB config sync interval, enter the time in seconds in the **GSLB Sync Interval** field.
   - To enable RSYNC port monitoring, select the **GSLB Config Sync Monitor** check box.
   - To disable automatically synchronizing the location DB files, clear the **GSLB Sync Loc Files** check box.

Full synchronization of GSLB configuration

Whenever there is a configuration change in the main site, the complete GSLB running configuration on the main site is pushed to all the subordinate sites.
Even if incremental synchronization is configured, a full synchronization is performed when the main site does not know the configuration status of the subordinate site. Some of such scenarios are as follows:

- Enable the automatic GSLB configuration synchronization feature for the first time.
- Reboot the Citrix ADC appliance.
- GSLB deployment has multiple main sites, and another main site becomes the active main site.
- Add a new subordinate site to the GSLB deployment.

GSLB configuration full synchronization is performed in the following manner:

- The main site pushes its latest configuration file to all the subordinate sites.

- Each subordinate site compares its own configuration with the latest configuration file sent by the main site. The subordinate site identifies the difference in configuration, and applies the delta configuration for its own site.
- The subordinate site generates status logs at each step and copies them to a file at a specific location.
- The main site pulls the status log files from the specified location.
- The main site prepares a log file with logs combined from all the subordinate sites.
- This combined log file is stored in /var/netscaler/gslb/periodic_sync.log file.

If you attempt to manually synchronize (with the `sync gslb config` command) a site while it is being autosynchronized, a "Sync in progress" error message appears. Autosynchronization cannot be triggered for a site that is in the process of being synchronized manually.

> **Attention:**
>
> Starting with Citrix ADC 12.1 build 49.37, SNMP traps are generated when you synchronize the GSLB configuration. In real-time synchronization, the synchronization status in the first SNMP trap is captured as failure. You can ignore this status because a second SNMP trap is automatically generated immediately after the first trap with the actual synchronization status. However, if the synchronization failed in the second attempt also, SNMP trap is not generated because the synchronization status has not changed from the previous synchronization status.

For details on configuring Citrix ADC appliance to generate traps, see Configuring the Citrix ADC to generate SNMP traps.

To enable GSLB full synchronization by using the CLI

```
set gslb parameter -GSLBSyncMode (IncrementalSync | FullSync)
<!--NeedCopy-->
```

Example:

```
set gslb parameter -GSLBSyncMode FullSync
<!--NeedCopy-->
```

To enable GSLB incremental synchronization by using the GUI:

1. Navigate to **Traffic Management > GSLB > Dashboard > Change GSLB settings**.
2. In the **Set GSLB Parameters** page, choose **FullSync** from the **GSLB Sync Mode** drop-down menu.

Multiple main sites in a GSLB deployment

The Citrix ADC appliance supports multiple main sites in an active-passive deployment. It is recommended to have two main sites in a GSLB deployment to cope against GSLB main site failure. Having two main sites can avoid single point of failure of GSLB configuration synchronization. At any time, only one main site can actively process the GSLB configuration from the user. If the configuration changes are performed simultaneously in more than one main site, it might lead to configuration inconsistency or configuration losses. Hence, it is recommended to perform configuration changes from only one main site at a time, and use the other main site as a backup when the active main site fails.

> Note:
>
> When multiple main sites are used in a GSLB deployment, RSYNC monitoring must be enabled.

To make a GSLB node as one of the main sites for GSLB configuration synchronization, run the following command:

```
set gslb parameter -automaticConfigSync Enabled
<!--NeedCopy-->
```

View GSLB synchronization status and summary

September 14, 2021

After the GSLB configuration is synchronized across the GSLB sites, you can view the detailed status and the summary of the last GSLB sync operation. This is applicable to both manual and real-time GSLB synchronization.

To view the GSLB synchronization status or summary by using the CLI

At the command prompt, type:

```
show gslb sync status
<!--NeedCopy-->
```

or

```
show gslb syncStatus -summary
<!--NeedCopy-->
```


Sample configuration output for GSLB manual synchronization

The following output displays the status of the manual GSLB configuration synchronization.

```
> sh gslb syncStatus
Displaying the status of the manual GSLB configuration synchronization:

gslb_site1[Master]:
        Getting Config:  ok
gslb_site2[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
gslb_natsite1[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
 Done
>
```

The following output displays the status summary of the manual GSLB configuration synchronization.

```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------------
        Site Name                Status          Reason
-------------------------------------------------------------------------------------------------------------------
        gslb_site1               Success         All Done
        gslb_site2               Failure         Error executing command on gslb site...ERROR: Connection failed
        gslb_natsite1            Success         All Done
Done
>
```

Sample configuration output for GSLB real-time synchronization

The following output displays the status of the real-time GSLB configuration synchronization for the master site:


```
> sh gslb syncStatus
Displaying the status of the real time GSLB configuration
    synchronization as master node:

site2[Master]:
                New GSLB configuration detected at Fri Jan 23 20:54:24
                    2020
                Fetching current configuration: Done
                Updating default.conf file: Done
site1[Slave]:
                Syncing gslb static proximity database to node site1:
                    Done
                Syncing inbuilt GSLB static proximity database to node
                    site1: Done
                Syncing ssl certificates, keys and CRLS to node site1:
                    Done
                Syncing current configuration to site1: Done
                Pulling status files from site1: Status file not
                    available yet(Sync in progress)
                Pulling status files from site1: Done
                site1 received new configuration from 10.102.217.205 in
                     file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                    conf
                Firing set gslb parameter -startConfigSync ENABLED
                    command: Done
                Fetching running GSLB Config: Done
                Comparing config: Done
                Applying changes: Done
                Firing set gslb parameter -startConfigSync DISABLED
                    command: Done
                Updating default.conf file: Done
Done
<!--NeedCopy-->
```

The following output displays the status of the real-time GSLB configuration synchronization for the slave site:

```
> sh gslb syncStatus
Displaying the status of the real time GSLB configuration
    synchronization as slave node:

                site1 received new configuration from 10.102.217.205 in
                     file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                    conf
```



```
                Firing set gslb parameter -startConfigSync ENABLED
                    command: Done
                Fetching running GSLB Config: Done
                Comparing config: Done
                Applying changes: Done
                Firing set gslb parameter -startConfigSync DISABLED
                    command: Done
                Updating default.conf file: Done
Done
<!--NeedCopy-->
```

The following output displays the status summary of the real-time GSLB configuration synchronization for the master site:

```
> sh gslb syncStatus -summary
Displaying the status summary of the real time GSLB configuration
    synchronization as master node:

---------------------------------------------------------------------------

                Site Name                                   Status
                                Reason
---------------------------------------------------------------------------

                site2                                       Success
                             All Done
                site1                                       Success
                             All Done

Done
<!--NeedCopy-->
```

The following output displays the status summary of the real-time GSLB configuration synchronization for slave site:

```
> sh gslb syncStatus  - summary
Displaying the status summary of the real time GSLB configuration
    synchronization as slave node:

---------------------------------------------------------------------------

                Site Name                                   Status
                                Reason
---------------------------------------------------------------------------
```



```
7                  site1                                        Success
                                  All Done
8
9  Done
10 <!--NeedCopy-->
```

To view the GSLB synchronization status or summary by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Dashboard**.
2. Click **View Synchronization Summary** or **View Synchronization Status**, as required.

SNMP traps for GSLB configuration synchronization

September 14, 2021

Starting with Citrix ADC 12.1 build 49.xx, the Citrix ADC appliance generates SNMP traps for both local and remote sites when you synchronize the GSLB configuration. SNMP traps are generated for both manual synchronization and real-time synchronization.

When you synchronize the GSLB configuration for the first time, SNMP traps are generated. In the subsequent synchronization attempts, the SNMP traps are generated only if there is a change in the synchronization status from the previous synchronization status. Also, the SNMP traps are generated only for sites for which the synchronization status changed from the previous state.

For example, consider that the first GSLB configuration synchronization is successful. When you synchronize the configuration for the second time and if the synchronization is successful again, then SNMP traps are not generated because the status is not changed. However, in the third attempt, if the synchronization fails for one of the sites, then SNMP trap is generated for that site alone.

In a high availability and a cluster setup, the appliance generates the SNMP traps when you synchronize the GSLB configuration from the new node irrespective of the previous synchronization status. Also, if SNMP trap option was previously disabled and then enabled, SNMP traps are generated from that point onwards irrespective of previous synchronization status.

The SNMP traps of GSLB configuration synchronization provide the following details:

- Name of the GSLB site for which the SNMP trap is sent.
- GSLB configuration synchronization status: Success or Failure.
- GSLB configuration synchronization mode: Incremental sync or Full sync.
- (Optional) Detailed information about the SNMP traps.

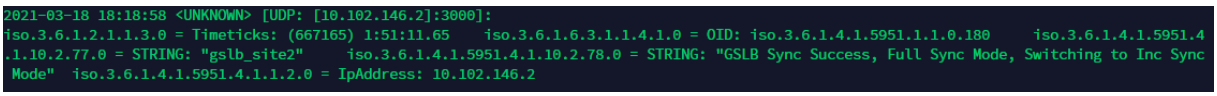
The SNMP traps are generated in the following scenarios:

- GSLB synchronization status for a GSLB site flips from Success to Failure, and conversely.
- GSLB synchronization mode changes from incremental synchronization to full synchronization, and conversely.

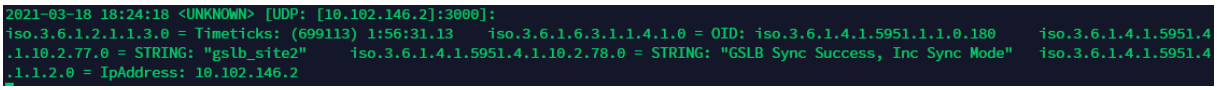
> Note:
>
> Even when incremental synchronization is enabled, if full synchronization is performed on a GSLB site for some reason, the reason for Full sync is mentioned in the "Detailed information" section of the trap message. For example, when a new GSLB site is added to the GSLB configuration.

Sample SNMP trap messages

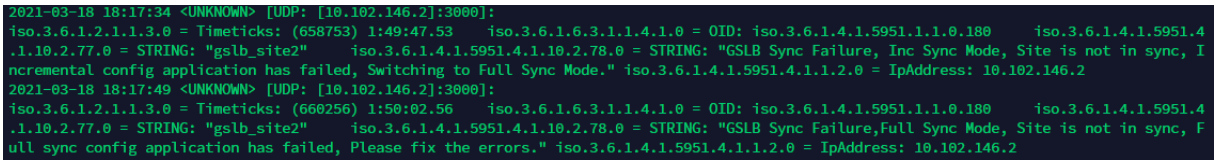
The following figure displays a sample SNMP trap for gslb_site2, where the GSLB configuration synchronization is successful using the Full sync mode.



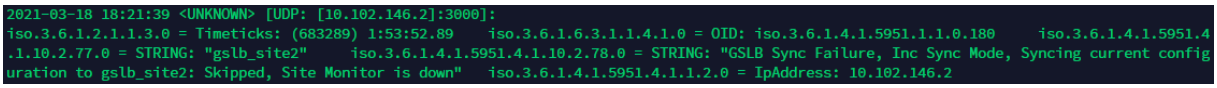
The following figure displays a sample SNMP trap for gslb_site2, where the GSLB configuration synchronization is successful using the incremental sync mode.



The following figure displays a sample SNMP trap for gslb_site2, where the GSLB configuration synchronization using the incremental sync mode is failed. The error message indicates that you must manually fix the errors to complete the synchronization.



The following figure displays a sample SNMP trap for gslb_site2, where the GSLB configuration synchronization using the incremental sync mode is failed. It also indicates the reason for sync failure, that is the site monitor is DOWN.



GSLB dashboard

September 14, 2021

You can view the overall status of the GSLB sites participating in GSLB on the GSLB dashboard.

You can access the GSLB settings from the dashboard. You can also start the GSLB configuration wizard from the dashboard. Additionally, you can perform the synchronization and test the GSLB setup from the dashboard.

To access the GSLB dashboard, navigate to **Configuration** > **Traffic Management** > **GSLB** > **Dashboard**.

Monitor GSLB services

September 14, 2021

When you bind a remote service to a GSLB virtual server, the GSLB sites exchange metric information, including network metric Information, which is the round-trip-time and persistence Information.

If a metric exchange connection is momentarily lost between any of the participating sites, the remote site is marked as DOWN and load balancing is performed on the remaining sites that are UP. When metric exchange for a site is DOWN, the remote services belonging to the site are marked DOWN as well.

The Citrix ADC appliance periodically evaluates the state of the remote GSLB services by using either MEP or monitors that are explicitly bound to the remote services. Binding explicit monitors to local services is not required, because the state of the local GSLB service is updated by default using the MEP. However, you can bind explicit monitors to a remote service. When monitors are explicitly bound, the state of the remote service is not controlled by the metric exchange.

By default, when you bind a monitor to a remote GSLB service, the Citrix ADC appliance uses the state of the service reported by the monitor. However, you can configure the Citrix ADC appliance to use monitors to evaluate services in the following situations:

- Always use monitors (default setting).
- Use monitors when MEP is DOWN.
- Use monitors when remote services and MEP are DOWN.

The second and third of the above settings enable the appliance to stop monitoring when MEP is UP. For example, in a hierarchical GSLB setup, a GSLB site provides the MEP information about its child sites to its parent site. Such an intermediate site may evaluate the state of the child site as DOWN because of network issues, though the actual state of the site is UP. In this case, you can bind monitors to the services of the parent site and disable MEP to determine the actual state of the remote service. This option enables you to control the manner in which the states of the remote services are determined.

To use monitors, first create them, and then bind them to GSLB services.

Configure monitor trigger

You can configure a GSLB site to always use monitors (the default), use monitors when MEP is down, or use monitors when both the remote service and MEP are down. In the latter two cases, the Citrix ADC appliance stops monitoring when MEP returns to the UP state.

To configure monitor triggering by using the command line interface

At the command prompt, type:

```
set gslb site <siteName> -triggerMonitor (ALWAYS | MEPDOWN |
   MEPDOWN_SVCDOWN)
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-North-America -triggerMonitor Always
<!--NeedCopy-->
```

To configure monitor triggering by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and double-click the site.
2. In the **Trigger Monitors** drop-down list, select an option for when to trigger monitoring.

Add or remove monitors

To add a monitor, you specify the type and the port. You cannot remove a monitor that is bound to a service. You must first unbind the monitor from the service.

To add a monitor by using the command line interface

At the command prompt, type the following commands to create a monitor and verify the configuration:

```
add lb monitor <monitorName> -type <monitorType> -destPort <portNumber>

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
```



```
<!--NeedCopy-->
```

To remove a monitor by using the command line interface

At the command prompt, type:

```
rm lb monitor <monitorName>
<!--NeedCopy-->
```

To add a monitor by using the configuration utility

Navigate to
Traffic Management > Load Balancing > Monitors, and add or delete a monitor.

Bind monitors to a GSLB service

Once you create monitors, you must bind them to GSLB services. When binding monitors to the services, you can specify a weight for the monitor. After binding one or more weighted monitors, you can configure a monitor threshold for the service. This threshold takes the service down if the sum of the bound monitor weights falls below the threshold value.

Note: In the configuration utility, you can set both the weight and the monitoring threshold at the same time that you bind the monitor. When using the command line, you must issue a separate command to set the service's monitoring threshold.

To bind the monitor to the GSLB service by using the command line interface

At the command prompt, type:

```
bind monitor <name> <serviceName> [ -state (Enabled | Disabled) ] -
   weight <positiveInteger>
<!--NeedCopy-->
```

Example:

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
<!--NeedCopy-->
```

To set the monitoring threshold for a GSLB service by using the command line interface

At the command prompt, type:


```
set gslb service <ServiceName> -monThreshold <PositiveInteger>
<!--NeedCopy-->
```

Example:

```
set gslb service service-GSLB-1 -monThreshold 9
<!--NeedCopy-->
```

To bind the monitor to the GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Click the **Monitor** section and bind the monitor to the GSLB service.

To set the monitoring threshold for a GSLB service by using the configuration utility

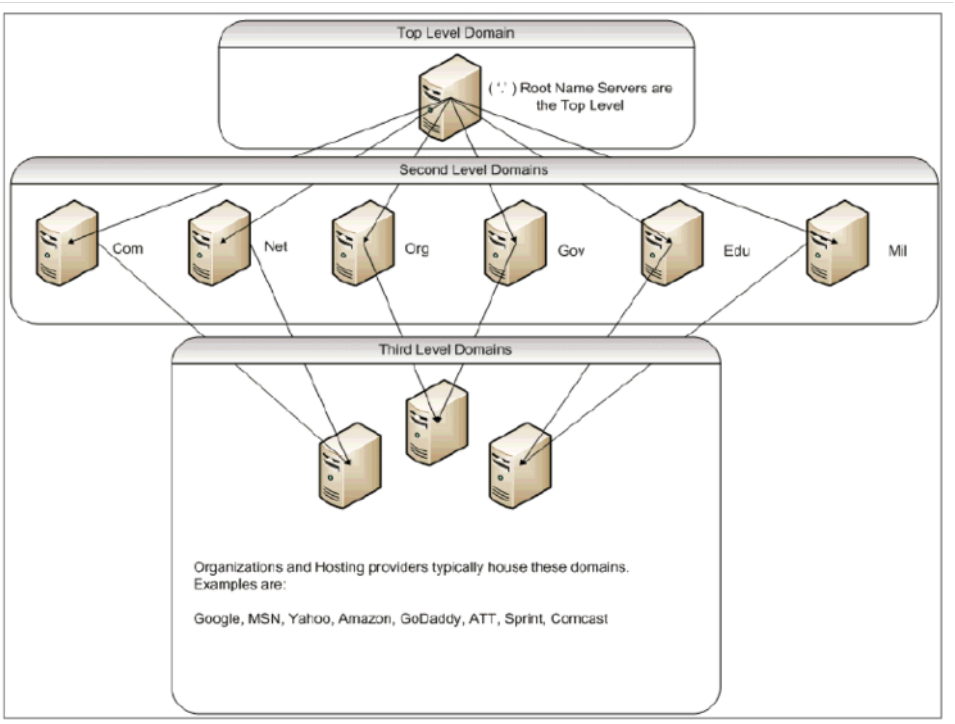
1. Navigate to Traffic Management > GSLB > Services.
2. Click the **Monitor Threshold** section and enter a threshold value.

How domain name system supports GSLB

September 14, 2021

The domain name system (DNS) is considered as a distributed database, which uses the Client/Server architecture. Name Servers are the servers in the architecture, and the resolvers are the clients that are library routines installed on an operating system that create and send queries across the network.

The logical hierarchy of the DNS is shown in the following diagram:



> Note:
>
> The second-level root servers are responsible for maintaining Name server to Address mappings for Name server delegations within the .com, .net, .org, .gov domains, and so on. Each domain within the second-level domains is responsible for maintaining Name Server to Address mappings for the lower-level organizational domains. At the organization level, the individual host addresses are resolved for www, FTP, and other service providing hosts.

Delegation

The main purpose of the current DNS topology is to ease the burden of maintaining all address records on one authority. This allows for delegation of an organization name space to that particular organization. The organization can then further delegate its space to subdomains within the organization. For example, under citrix.com you can create subdomains called `sales.citrix.com`, `education.citrix.com`, and `support.citrix.com`. The corresponding departments can maintain their own set of Name servers that are authoritative for their subdomain, and then maintain their own set of host name to address mappings. No single department is responsible for maintaining all of Citrix address records. Each department can change addresses and modify topologies, and not impose more work at the higher-level domain or organization.

Benefits of the hierarchical topology

Some of the benefits of the hierarchical topology include:

- Scalability
- Adding caching functionality into Name servers at each level, where a DNS request is served by a host that is not authoritative for a particular domain but can contribute the answer to the query, and cut down on congestion and response time.
- Caching also creates redundancy and resiliency to server failure. If one name server fails, it is still possible that records can be served from other servers that have recent cached copies of the same records.

Resolvers

Resolvers are the client component in the DNS system. Programs that are running on a host that need information from the domain name space use the resolver. The resolver handles:

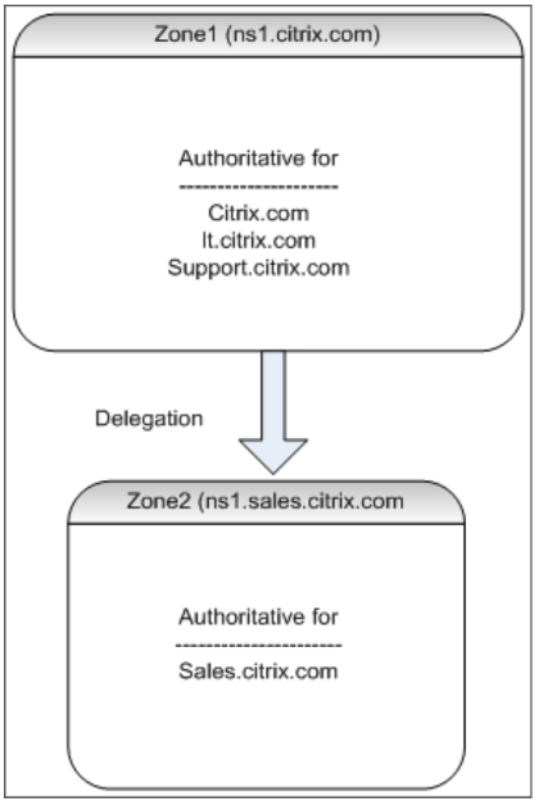
- Querying a name server.
- Interpreting responses (which might be resource records or an error).
- Returning the information to the programs that requested it.

The resolver is a set of library routines that are compiled into programs like telnet, FTP, and ping. They are not separate processes. The resolvers can put together a query, send it, and wait for an answer. And, send it again (possibly to a secondary Name Server) if it is not answered within a certain time. These types of resolvers are known as stub resolvers. Some resolvers have the added functionality to cache records, and honor time to live (TTL). In Windows, this functionality is available through the DNS Client service; viewable through the “services.msc” console.

Name Servers

Name Servers generally store complete information about a particular part of a domain name space (called a zone). The Name Server is then said to have authority for that zone. They can also be authoritative for multiple zones.

The difference between a domain and a zone is subtle. A domain is the full set of entities including its subdomains while a zone is only the information within a domain that is not delegated to another Name Server. An example of a zone is `citrix.com`, while `sales.citrix.com` is a separate zone if that zone is delegated to another Name Server within the subdomain. In this case, the primary Citrix zone can include `citrix.com`, `it.citrix.com`, and `support.citrix.com`. Because the `sales.citrix.com` is delegated, it is not part of the zone that the `citrix.com` Name Server is authoritative over. The following diagram shows the two zones.

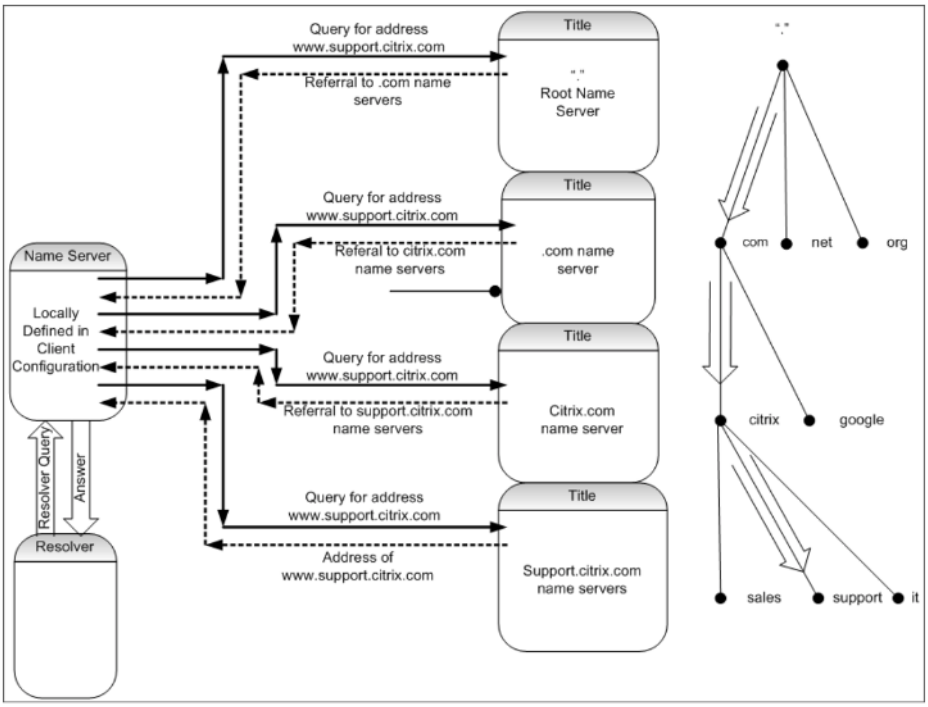


To properly delegate a subdomain, you must assign authority for the subdomain to different Name Servers. In the preceding example, the `ns1.citrix.com` does not contain information about the `sales.citrix.com` subdomain. Instead, it contains pointers to the Name servers that are authoritative for the `ns1.sales.citrix.com` subdomain.

Root name servers and query resolution

Root Name servers know the IP addresses of all the Name servers authoritative for the second-level domains. If a Name Server does not have information about a given domain in its own data files, then it only needs to contact a root server to begin traversing the proper branch of the **DNS** tree structure to eventually get to the given domain. This involves a series of requests to multiple Name servers to help with the tree traversal to find the next authoritative Name server, which needs to be contacted for further resolution.

The following diagram shows a typical DNS request, assuming that there is no cached record for the requested name during the traversal. The following example uses a mock up of the Citrix domain.



Recursive and non-recursive queries

The preceding example demonstrates the two types of queries that can occur.

- Recursive query: The query between the resolver and the locally configured Name server is recursive. This means that the Name server receives the query and does not respond to the resolver until the query is fully answered, or an error is returned. If the Name Server receives a referral to the query, then the Name server follows the referral until the Name server finally receives the answer (IP address) returned.
- Non-recursive query: The query that the locally configured Name server makes to the subsequent authoritative domain-level Name server is non-recursive (or iterative). Each request is immediately responded with either a referral to a lower-level authoritative server or the answer to the query, if the queried Name server contains the answer in its data files or its cache.

Caching

Although the resolution process is involved, and might potentially require small requests to several hosts, it is fast. One of the factors that increases the speed of DNS resolution is caching. Each time a Name server receives a recursive query, it might have to communicate to other servers to eventually get to the proper authoritative server for the specific request. It stores all of the information that it receives for future reference. When the next client makes a similar request, such as a different host

but in the same domain, it already knows the Name server that is authoritative for that domain, and can send a request directly there instead of starting at the root Name server.

Caching can occur for negative responses also, such as the queries for hosts that do not exist. In this case, the server must not query the authoritative Name server for the requested domain to figure out that the host does not exist. To save time, the Name server simply checks the cache and responds back with the negative record.

Name servers do not cache records indefinitely, or else you can never update the IP addresses. To avoid synchronization problems, DNS responses contain a time to live (TTL). This field describes the time interval for which the cache can store a record before it must discard it and check with the authoritative Name server for any updated records. If the records have not changed, the use of TTL also allows rapid dynamic responses from devices performing GSLB.

Resource record types

Various RFCs provide a comprehensive list of DNS resource record types and its description. The following table lists the common resource record types.

Resource record type	Description	RFC
A	A host address	RFC 1035
NS	An authoritative name server	RFC 1035
MD	A mail destination (Obsolete - use MX)	RFC 1035
MF	A mail forwarder (Obsolete - use MX)	RFC 1035
CNAME	The canonical name for an alias	RFC 1035
SOA	Marks the start of a zone of authority	RFC 1035
WKS	A well known service description	RFC 1035
PTR	A domain name pointer	RFC 1035
HINFO	Host information	RFC 1035
MINFO	Mailbox or mail list information	RFC 1035
MX	Mail exchange	RFC 1035
TXT	Text strings	RFC 1035

Resource record type	Description	RFC
AAAA	IP6 Address	RFC 3596
SRV	Server selection	RFC 2782]

How GSLB supports DNS

GSLB uses algorithms and protocols that decide which IP address must be sent for a DNS query. GSLB sites are geographically distributed and there is a DNS authoritative Name server at each site running as a service on the Citrix ADC appliance. All Name servers at the various sites involved are authoritative for the same domain. Each of the GSLB domains is a subdomain for which a delegation is configured. Therefore, the GSLB Name servers are authoritative and can use one of the various load balancing algorithms to decide which IP address to return.

A delegation is created by adding a Name server record for the GSLB domain in the parent domain database files and a subsequent address record for the Name servers that are used for the delegation. For example, if you want to use GSLB for `www.citrix.com`, then the following Bind SOA file can be used to delegate requests to `www.citrix.com` to Name Servers: Netscaler1 and Netscaler2.

```
################################################################################

@ IN SOA citrix.com. hostmaster.citrix.com. (
1 ; serial
3h ; refresh
1h ; retry
1w ; expire
1h ) ; negative caching TTL
IN NS ns1
IN NS ns2
IN MX 10 mail

ns1   IN A 10.10.10.10
ns2   IN A 10.10.10.20
mail  IN A 10.20.20.50

### Old Configuration if www was not delegated to a GSLB name server
www IN A 10.20.20.50

### Updated Configuration
Netscaler1 IN A xxx.xxx.xxx.xxx
Netscaler2 IN A yyy.yyy.yyy.yyy
www IN NS Netscaler1.citrix.com.
```



```
www IN NS Netscaler2.citrix.com.
###
IN MX 20 mail2
mail2 IN A 10.50.50.20
##############################################################################


<!--NeedCopy-->
```

Understanding BIND is not a requirement for configuring DNS. All compliant DNS server implementations have a method of creating the equivalent delegation. Microsoft DNS servers can be configured for delegation using the instructions at Create a zone delegation.

What makes GSLB on Citrix ADC appliance different from using the standard DNS service for distributing traffic is that the Citrix ADC GSLB sites exchange data using a proprietary protocol called Metric Exchange Protocol (MEP). With MEP, the GSLB sites are able to maintain information about all other sites. When a DNS request is received, the MEP considers the GSLB metrics to determine information such as the following:

- Site with the least number of current connections
- Site that is closest to the LDNS server, which sent the request based on round trip times (RTT).

There are several load balancing algorithms that can be used, but GSLB is a DNS with the brain underneath telling the Name Server (hosted on the Citrix ADC appliance) which address must be sent based on metrics of the participating sites.

Other benefits that GSLB provides are the ability to maintain persistence (or site affinity). Responses to the incoming DNS queries can be compared with the source IP address to determine if that address was directed to a particular site in the recent past. If so, then the same address is sent in the DNS response to ensure that the client session is maintained.

Another form of persistence is obtained at the site level by using HTTP redirects, or HTTP proxying. These forms of persistence occur after the DNS response occurs. Therefore, if you get an HTTP request at a site that contains a cookie to direct the request to a different participating site, then you can either respond with a redirect or proxy the request to the appropriate site.

Metric exchange protocol

Metric Exchange Protocol (MEP) is used to share the data used in GSLB calculations across sites. Using MEP connections, you exchange three types of data. These connections need not be secure over TCP port 3011 or can be secure using SSL over TCP port 3009.

The following three types of data are exchanged, and have their own intervals and exchange methods.

- **Site metric exchange:** This is a polling exchange model. For example, if site1 has a configuration for site2 services, then every second site1 asks site2 for the status of the GSLB services. Site2 responds with the state and other load details.
- **Network metric exchange:** This is the LDNS RTT information exchange, which is used in the dynamic proximity load balancing algorithm. This is a push exchange model. Every five seconds, each site pushes its data to other participating sites.
- **Persistency exchange:** This is for SOURCEIP persistency exchange. This is also a push exchange model. Every five seconds, each site pushes its data to other participating sites.

By default, site services are monitored over MEP based on polling information only. If you bind monitors based on monitor interval, the state is updated and you can control the frequency of the updates by setting the monitoring interval accordingly.

Upgrade recommendations for GSLB deployment

September 14, 2021

This section provides recommendations on the sequence in which GSLB nodes need to be upgraded in various GSLB setups. It also addresses a few FAQs.

Note: The Citrix ADC appliance from which the GSLB synchronization is started is referred to as the 'main site' and the GSLB sites on which the configuration is copied as the 'subordinate sites'.

Before you start the upgrade process, read the prerequisites mentioned in the following topics:

- Before you begin
- Upgrade a high availability pair.
- Upgrade a cluster.

Points to note when upgrading GSLB setups

- In an HA setup, first upgrade the subordinate sites and then the main site.
- In an HA setup, service states might not propagate from an older build primary node to a newer build secondary node. However, if the builds are of different versions, but have the same HA version, the service state might still propagate.
- If GSLB is configured within a cluster, first upgrade the non-owner nodes, and then upgrade the owner node. If there is one site or multiple sites in a cluster, follow the same upgrade sequence in each of the site.
- Enable new GSLB features only after you upgrade all nodes to a newer build.

- Upgrade all GSLB nodes to the latest build. There is no functional impact on the available features when some of the GSLB nodes are using an older version and some of the GSLB nodes are upgraded to a newer version.

FAQs

- **Are GSLB service states propagated when instances run different software versions?**

  GSLB MEP is functional when instances run on different versions and GSLB service states are propagated across GSLB sites. There is no impact on the MEP communication when instances run different versions after an upgrade.

- **Is it recommended to do configuration changes during an upgrade?**

  In a GSLB setup, when a main site is being upgraded, it is not recommended to do configuration changes on any other GSLB nodes.

Related resources

The following resources provide information about upgrading a Citrix ADC instance using Citrix ADM:

- 10 ways Citrix ADM service supports easier Citrix ADC upgrades
- Use Citrix ADM service to upgrade Citrix ADC instances
- Use Citrix ADM software to upgrade Citrix ADC instances

Use case: Deployment of domain name based autoscale service group

September 14, 2021

> **Tip**
>
> For information about the GSLB service groups, see Configuring a GSLB Service Group

Deployment scenario

Two datacenters are deployed in two AWS regions, one in Sydney and one in North Virginia. Another datacenter is deployed in Azure. An AWS ELB in each AWS region is used to load balance the application servers. ALB is used for Azure to load balancing the application server. The Citrix ADC appliances are configured for GSLB for the ELBs and ALB using GSLB domain name based autoscale service group.

> **Important**
>
> You must configure the required security groups in AWS and attach it to the GSLB instance. Port 53 must be allowed in the security group inbound and outbound rules. Also, ports (3009 or 3011 depending on secure MEP configuration) for MEP communication must be open. For application monitoring, the corresponding ports must be allowed in the security group outbound rules.

The configuration steps for the above deployment scenario and the corresponding CLI commands are as follows:

1. Create datacenters (represented by GSLB sites).

   ```
   add gslb site aws-sydney 192.0.2.2

   add gslb site aws-nvirginia 198.51.100.111

   add gslb site alb-southindia 203.0.113.6
   ```

2. Add a name server with the DNS gateway IP address where the GSLB node is added. This must be done in all datacenters.

   ```
   add dns nameServer 8.8.8.8
   ```

3. Add servers for ELB and ALB.

   ```
   add server aws-sydney_server lb-sydney-1052691850.ap-southeast-2.elb.
   amazonaws.com

   add server aws-nvirginia_server LB-nvirginia-860559595.us-east-1.elb.
   amazonaws.com

   add server alb-southindia_server alb.southindia.cloudapp.azure.com
   ```

4. Add GSLB autoscale service groups for each ELB and ALB and bind each server to the respective service group.

   ```
   add gslb serviceGroup aws-nvirginia_sg HTTP -autoScale DNS -siteName
   aws-nvirginia

   add gslb serviceGroup aws-sydney_sg HTTP -autoScale DNS -siteName aws-
   sydney

   add gslb serviceGroup alb-southindia_sg HTTP -autoScale DNS -siteName
   alb-southindia

   bind gslb serviceGroup aws-nvirginia_sg aws-nvirginia_server 80

   bind gslb serviceGroup aws-sydney_sg aws-sydney_server 80

   bind gslb serviceGroup alb-southindia_sg alb-southindia_server 80
   ```


5. Add a GSLB virtual server and bind the application domain and the service groups to this virtual server.

```
add gslb vserver gv1 HTTP

bind gslb vserver gv1 -serviceGroupName aws-nvirginia_sg

bind gslb vserver gv1 -serviceGroupName aws-sydney_sg

bind gslb vserver gv1 -serviceGroupName alb-southindia_sg
```

Use case: Deployment of IP address based GSLB service group

September 14, 2021

> **Tip**
>
> For information about the GSLB service groups, see Configuring a GSLB Service Group.

Deployment scenario

If there are multiple applications hosted on the same application server, the GSLB should probe these applications to see if the applications are responding or not. If an application is not responding, the user must be directed to the server on which the application is UP. Also, if one of the applications is DOWN, then the server should not be marked DOWN, because the other applications are UP.

In the following example, multiple applications (HTTPS) are hosted on one server in each GSLB site and hence all these applications resolve to one IP address of the respective site.

Using the GSLB service groups, you can have the same server with an IP address and port bound to multiple service groups where each service group represents a different application.

An application specific monitor is bound to the service groups that marks the service group as DOWN if the application is DOWN. Thus, whenever an application is DOWN, only that application is taken out from the setup and not the server.

```
```
add gslb serviceGroup app1_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app2_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app1_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2
```

 
```

```
7
8  add gslb serviceGroup app2_site2 HTTP -maxClient 0 -cip DISABLED -
       cltTimeout 180 -svrTimeout 360 -siteName s2
9
10 add lb monitor http_app2 HTTP -respCode 200 -httpRequest "GET /testsite
       /app2.html"
11
12 add lb monitor http_app1 HTTP -respCode 200 -httpRequest "GET /testsite
       /app1.html"
13
14 bind gslb serviceGroup app1_site1 192.0.2.140 80
15
16 bind gslb serviceGroup app1_site1 -monitorName http_app1
17
18 bind gslb serviceGroup app2_site1 192.0.2.140 80
19
20 bind gslb serviceGroup app2_site1 -monitorName http_app2
21
22 bind gslb serviceGroup app1_site2 192.0.2.142 80
23
24 bind gslb serviceGroup app1_site2 -monitorName http_app1
25
26 bind gslb serviceGroup app2_site2 192.0.2.142 80
27
28 bind gslb serviceGroup app2_site2 -monitorName http_app2
29 <!--NeedCopy--> ```
```

How-to articles

September 14, 2021

The GSLB how-to articles contain information about some of the important GSLB configurations such as customizing GSLB configuration, configuring persistent connections, disaster recovery, and so on.

Customizing Your GSLB Configuration

Configuring Persistent Connections

Managing Client Connections

Configuring GSLB for Proximity

Protecting the GSLB Setup Against Failure

Configuring GSLB for Disaster Recovery

Customize your GSLB configuration

September 14, 2021

Once your basic GSLB configuration is operational, you can customize it by modifying the bandwidth of a GSLB service, configuring CNAME based GSLB services, static proximity, dynamic RTT, persistent connections, or dynamic weights for services, or changing the GSLB Method.

You can also configure monitoring for GSLB services to determine their states.

These settings depend on your network deployment and the types of clients you expect to connect to your servers.

Modify maximum connections or maximum bandwidth for a GSLB service

You can restrict the number of new clients that can simultaneously connect to a load balancing or content switching virtual server by configuring the maximum number of clients and/or the maximum bandwidth for the GSLB service that represents the virtual server.

To modify the maximum clients or bandwidth of a GSLB service by using the command line interface

At the command prompt, type the following command to modify the maximum number of client connections or the maximum bandwidth of a GSLB service and verify the configuration:

```
set gslb service <serviceName> [-maxClients <positive_integer>] [-
    maxBandwidth <positive_integer>]
show gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
set glsb service Service-GSLB-1 -maxBandwidth 100 -maxClients 100
show gslb service Service-GSLB-1
<!--NeedCopy-->
```


To modify the maximum clients or bandwidth of a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**, and double-click a service.
2. Click in the **Other Settings** section and set the following parameters:
    - Max Clients—maxClients
    - Max Bandwidth—maxBandwidth

Create CNAME-based GSLB services

To configure a GSLB service, you can use the IP address of the server or a canonical name of the server. If you want to run multiple services (like an FTP and a Web server, each running on different ports) from a single IP address or run multiple HTTP services on the same port, with different names, on the same physical host, you can use canonical names (CNAMES) for the services.

For example, you can have two entries in DNS as ftp.example.com and www.example.com for FTP services and HTTP services on the same domain, example.com. CNAME-based GSLB services are useful in a multilevel domain resolver configuration or in multilevel domain load balancing. Configuring a CNAME-based GSLB service can also help if the IP address of the physical server is likely to change.

If you configure CNAME-based GSLB services for a GSLB domain, when a query is sent for the GSLB domain, the Citrix ADC appliance provides a CNAME instead of an IP address. If the A record for this CNAME record is not configured, the client must query the CNAME domain for the IP address. If the A record for this CNAME record is configured, the Citrix ADC appliance provides the CNAME with the corresponding A record (IP address). The Citrix ADC appliance handles the final resolution of the DNS query, as determined by the GSLB method. The CNAME records can be maintained on a different Citrix ADC appliance or on a third-party system.

In an IP-address-based GSLB service, the state of a service is determined by the state of the server that it represents. However, a CNAME-based GSLB service has its state set to UP by default; the virtual server IP (VIP) address or metric exchange protocol (MEP) are not used for determining its state. If a desktop-based monitor is bound to a CNAME-based GSLB service, the state of the service is determined according to the result of the monitor probes.

You can bind a CNAME-based GSLB service only to a GSLB virtual server that has the DNS Record Type as CNAME. Also, a Citrix ADC appliance can contain at most one GSLB service with a given CNAME entry.

The following are some of the features supported for a CNAME-based GSLB service:

- GSLB-policy based site affinity is supported, with the CNAME as the preferred location.
- Source IP persistence is supported. The persistency entry contains the CNAME information instead of the IP address and port of the selected service.

The following are the limitations of CNAME-based GSLB services:

- Site persistence is not supported, because the service referenced by a CNAME can be present at any third-party location.
- Multiple-IP-address response is not supported because one domain cannot have multiple CNAME entries.
- Source IP Hash and Round Robin are the only load balancing methods supported. The Static Proximity method is not supported because a CNAME is not associated with an IP address and static proximity can be maintained only according to the IP addresses.

Note: The Empty-Down-Response feature should be enabled on the GSLB virtual server to which you bind the CNAME-based GSLB service. If you enable the Empty-Down-Response feature, when a GSLB virtual server is DOWN or disabled, the response to a DNS query, for the domains bound to this virtual server, contains an empty record without any IP addresses, instead of an error code.

To create a CNAME-based GSLB service by using the command line interface

At the command prompt, type:

```
add gslb service <serviceName> -cnameEntry <string> -siteName <string>
<!--NeedCopy-->
```

Example:

```
add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -
    siteName Site-GSLB-East-Coast
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -
    siteName Site-GSLB-West-Coast
<!--NeedCopy-->
```

To create a CNAME-based GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Create a service, and set the **Type to Canonical Name Based**.

Configure transition Out-Of-Service State (TROFS) in GSLB

When you configure persistence on a GSLB virtual server to which a service is bound, the service continues to serve requests from the client even after it is disabled, accepting new requests or connections only to honor persistence. After a configured period of time, known as the graceful shutdown period, no new requests or connections are directed to the service, and all of the existing connections are closed.

When disabling a service, you can specify a graceful shutdown period, in seconds, by using the delay argument. During the graceful shutdown period, if the service is bound to a virtual server, its state appears as Out of Service.

Configure dynamic weights for services

In a typical network, there are servers that have a higher capacity for traffic than others. However, with a regular load balancing configuration, the load is evenly distributed across all services even though different services represent servers with different capacities.

To optimize your GSLB resources, you can configure dynamic weights on a GSLB virtual server. The dynamic weights can be based on either the total number of services bound to the virtual server or the sum of the weights of the individual services bound to the virtual server. Traffic distribution is then based on the weights configured for the services.

When dynamic weights are configured on the GSLB virtual server, requests are distributed according to the load balancing method, the weight of the GSLB service, and the dynamic weight. The product of the weight of the GSLB service and the dynamic weight is known as the cumulative weight. Therefore, when dynamic weight is configured on the GSLB virtual server, requests are distributed on the basis of the load balancing method and the cumulative weight.

When dynamic weight for a virtual server is disabled, the numerical value is set to 1. This ensures that the cumulative weight is a non-zero integer at all times.

Dynamic weight can be based on the total number of active services bound to load balancing virtual servers or on the weights assigned to the services.

Consider a configuration with two GSLB sites configured for a domain and each site has two services that can serve the client. If a service at either site goes down, the other server in that site has to handle twice as much traffic as a service at the other site. If dynamic weight is based on the number of active services, the site with both services active has twice the weight of the site with one service down and therefore receives twice as much traffic.

Alternatively, consider a configuration in which the services at the first site represent servers that are twice as powerful as servers at the second site. If dynamic weight is based on the weights assigned to the services, twice as much traffic can be sent to the first site as to the second.

Note: For details on assigning weights to load balancing services, see Assigning Weights to Services.

As an illustration of how dynamic weight is calculated, consider a GSLB virtual server that has a GSLB service bound to it. The GSLB service represents a load balancing virtual server that in turn has two services bound to it. The weight assigned to the GSLB service is 3. The weights assigned to the two services are 1 and 2 respectively. In this example, when dynamic weight is set to:

- **Disabled**:The cumulative weight of the GSLB virtual server is the product of the dynamic weight (disabled = 1) and the weight of the GSLB service (3), so the cumulative weight is 3.

- **SERVICECOUNT**: The count is the sum of the number of services bound to the load balancing virtual servers corresponding to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.
- **SERVICEWEIGHT**: The dynamic weight is the sum of the weights of services bound to the load balancing virtual servers corresponding to the GSLB service (3), and the cumulative weight is the product of the dynamic weight (3) and the weight of the GSLB service (3), which is 9.

Note: Dynamic weights are not applicable when content switching virtual servers are configured.

To configure a GSLB virtual server to use dynamic weights by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight SERVICECOUNT | SERVICEWEIGHT
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
<!--NeedCopy-->
```

To set GSLB virtual server to use dynamic weights by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers, double-click the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).
2. Click the **Method** section and, from the **Dynamic Weight** drop-down list, select **SERVICECOUNT** or **SERVICEWEIGHT**.

How to configure persistence in GSLB

September 14, 2021

Persistence ensures that a series of client requests for a particular domain name is sent to the same data center instead of being load balanced. If persistence is configured for a particular domain, it takes precedence over the configured GSLB method. You can use persistence for deployments where an information related to a client transaction is stored locally on an instance, which has served the initial requests. For example, the deployments for e-commerce that uses a shopping cart, where the server needs to maintain the state of the connection to track the transaction. The Citrix ADC appliance selects a data center to process a client request. With persistence enabled, it forwards the same IP address of the selected data center for all subsequent Domain Name System (DNS) requests. If a

persistence session points to a data center that is DOWN, the Citrix ADC appliance uses the configured GSLB method to select a new data center. It then becomes persistent for subsequent requests from the client.
For persistence in GSLB, the same set of persistence identifiers (persistID) must be configured on the GSLB virtual servers in all data centers. The GSLB module uses the persistence identifier to uniquely identify a GSLB virtual server. When Source IP persistence is enabled on the GSLB virtual server, the persistence sessions are also exchanged as part of the metrics exchange. For the Citrix ADC appliance to support persistence across sites, persistence related configuration must be done on all the participating GSLB sites. Citrix recommends persistence in GSLB for stateful applications, which requires clients to reconnect to the same application instance for the subsequent requests.

You can achieve persistence in GSLB by the following ways:

- Persistence on GSLB virtual server
- Site persistence on GSLB services

Persistence on GSLB virtual server

Persistence on GSLB virtual server is used during the DNS requests. The Source IP address of the DNS request is used to create persistence session between the client and the data center. DNS clients are generally the Local DNS (LDNS) or DNS gateways proxying a set of clients sitting behind them (in ISPs). Persistence on a GSLB virtual server is application protocol agnostic.
In general, multiple DNS gateways or Local Domain Name Servers (LDNS) are configured in the client network. Citrix recommends you to configure an appropriate persistence mask because for the subsequent DNS requests, irrespective of the upstream LDNS devices used to connect to the ADC appliance, the client is able to persist to the same data center, which had served the earlier requests. After the persistence session is created for an LDNS IP address, all the end clients connecting using that LDNS are given the same data center IP address.

Site persistence on GSLB services

Site persistence becomes effective while processing the application requests. Site persistence works only for HTTP and HTTPS traffic because the persistency is achieved using HTTP cookie. As cookies are maintained on HTTP clients (browsers), it gives visibility into the clients sitting behind the DNS gateways. When you use cookies to achieve persistency for clients, no resources are consumed on the ADC appliance for each incoming client. When you bring a GSLB service DOWN with a delay time, the service goes into the transition to out of service (TROFS) state. Persistence is supported as long as the service is in the UP or TROFS state. That is, if the same client sends a request for the same service within the specified delay time after a service is marked TROFS, the same GSLB site (data center) services the request.

If you access an application through an alias, ensure that the CNAME record is also configured on the Citrix ADC appliance. In a parent-child topology, site persistence does not work when you access an application through an alias.

> Note
>
> If the connection proxy is specified as the site persistence method and you also want to configure persistence on LB virtual servers, source IP persistence is not recommended. When the connection is proxied, an IP address owned by the ADC appliance is used, and not the actual IP address of the client.
> Configure an appropriate persistence, which does not use source IP of the HTTP(S) request to identify the client, for example, cookie persistence or rule-based persistence.

Configure persistence based on source IP address

If source IP persistence is configured on GSLB virtual server, persistence sessions are created for the source IP address of the DNS request. Depending on the Extended Client Subnet (ECS) feature, the source IP address of the DNS request is taken from any of the following:

- The source IP in the IP header of the incoming DNS request packet
- The ECS option in the DNS request For more information on ECS, see Use the EDNS0 client subnet option for Global Server Load Balancing.

Persistence sessions for a client last until the persistence timeout. After the timeout period expires, existing persistence sessions are cleared. For subsequent requests, a new GSLB decision is made and a different GSLB service IP address might be selected.
The source IP persistence on GSLB virtual server and site persistence on GSLB service complements each other. If source IP persistence is disabled on GSLB virtual server, the GSLB virtual server chooses a different GSLB service each time the DNS tries to do the resolution. The client also connects to a different GSLB service and the data center which receives the application request proxy the connection to the data center which served the client first. This might add some latency. So by enabling source IP persistence on GSLB virtual server can avoid frequent such multiple hops for application requests. If the source IP persistence session has expired and the client reconnects after that, the site persistence connects the client back to the data center, which had served the client initially. Also, if the client connects back through a DNS gateway, which does not fall within the persistence mask range configured, then as well site persistence helps clients stick to the data center that served the first request.

To configure persistence based on source IP address by using the CLI

At the command prompt, type:

```
set gslb vserver <name> -persistenceType (SOURCEIP|NONE) -persistenceId
    <positive_integer> [-persistMask <netmask>]  -[timeout <mins>]
```



```
2 <!--NeedCopy-->
```

Example:

```
1 set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -
     persistenceId 23 -persistMask 255.255.255.255  - timeout 2
2 <!--NeedCopy-->
```

To configure persistence based on source IP address by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).
2. Click the **Persistence** section and, from the **Persistence** drop-down list, select **SOURCEIP** and set the following parameters:
   - Persistence Id—persistenceID
   - Time-out—timeout
   - IPv4 Netmask or IPv6 Mask length—persistMask

Configure site persistence based on HTTP cookies

Site persistence is achieved using HTTP cookies (known as a “site cookie”) to reconnect the client to the same server. When the GSLB appliance responds to a client DNS request by sending the IP address of the selected GSLB site, the client sends an HTTP request to that GSLB site. Application endpoint in that GSLB site adds a site cookie to the HTTP header, and site persistence is in effect.
If the client sends a DNS query after the client cache is expired, the DNS request might be directed to a different GSLB site. The new GSLB site uses the site cookie present in the client request header to implement persistence. Site persistence feature becomes active under the following conditions:

- When the domain name in the host header matches one of the GSLB domains
- When site persistence is enabled on the GSLB service that represents the virtual server receiving the application traffic.

The site cookie contains information about the selected GSLB service on which the client has a persistent connection. If the GSLB service pointed by the cookie is DOWN or removed from the GLSB configuration, then the virtual server that receives the traffic continues to process the traffic. The cookie expiration is based on the cookie timeout configured on the Citrix ADC appliance. If the virtual server names are not identical on all the sites, you must use the persistence identifier. Cookies inserted are compliant with RFC 2109.

Citrix ADC supports two types of site persistence:

- Connection Proxy
- HTTP redirect

Connection Proxy

In the Connection Proxy mode of site persistence, the data center that receives the subsequent application request performs the following tasks to establish a connection:

1. Creates a connection to the GSLB site that inserted the site cookie.
2. Proxies the client request to the original site.

   > **Note:**
   >
   > The proxy server establishes connection with the original site using the following details:
   >
   > - The SNIP of the new site is the source IP address.
   > - The GSLB service public IP address of the original site is the destination IP address.
   > - An ephemeral port is the source port and GSLB service port is the destination port.
   > - Uses either HTTP or HTTPS protocols depending on the GSLB service type.

3. Receives a response from the original GSLB site.
4. Relays that response back to the client.
5. Closes the connection.

HTTP redirect

If the GSLB configuration uses HTTP redirect persistence, the new site redirects the request to the site that originally inserted the cookie. Domain name in the redirect URL is the site domain. Ensure that both cookies and SSL certificates are applicable to both GSLB domain and site domain. To apply cookies for both GSLB and site domain, the cookie domain must be the site to GSLB domain. To apply SSL certificates to both GSLB and site domain, certificate bound to the SSL virtual server must be a wildcard certificate.

Connection proxy occurs when the following conditions are satisfied:

- Requests are sent for a domain participating in GSLB. The domain is obtained from the URL/Host header.
- The local GSLB service has connection proxy enabled.
- The request includes a valid cookie that contains the IP address of an active remote GSLB service.

> **Note**
>
> In a GSLB parent-child configuration, the connection proxy works as intended even when a GSLB service is not configured on a child site. However, if you have extra configuration such as client authentication, client IP address insertion, or other SSL-specific requirement, you must add an explicit GSLB service on the site and configure it accordingly.

> For more information about parent-child topology, see Parent-Child Topology Deployment Using the MEP Protocol.

To set persistence based on HTTP cookies by using the CLI

At the command prompt, type:

```
set gslb service <serviceName> -sitePersistence (ConnectionProxy [-
   sitePrefix <prefix>] | HTTPredirect -sitePrefix <prefix>)
<!--NeedCopy-->
```

Example:

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -
   sitePrefix vserver-GSLB-1
<!--NeedCopy-->
```

To set persistence based on cookies by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Services** and select the service that you want to configure for site persistence (for example, service-GSLB-1).
2. Click the **Site Persistence** section and set persistence based on cookies.

Manage client connections

September 14, 2021

To facilitate management of client connections, you can enable delayed cleanup of connections to the virtual server. You can then manage local DNS traffic by configuring DNS policies.

Enable delayed cleanup of virtual server connections

The state of a virtual server depends on the states of the services bound to it, and the state of each service depends on the monitors bound to it. If a server is slow or down, the monitoring probes time out and the service that represents the server is marked as DOWN. A virtual server is marked as DOWN only when all services bound to it are marked as DOWN. You can configure services and virtual servers to either terminate all connections when they go down, or allow the connections to go through. The latter setting is for situations in which a service is marked as DOWN because of a slow server.

When you configure the down state flush option, the Citrix ADC appliance performs a delayed cleanup of connections to a GSLB service that is down.

To enable delayed cleanup of virtual server connections by using the command line interface

At the command prompt, type the following commands to configure delayed connection cleanup and verify the configuration:

```
set gslb service <name> -downStateFlush (ENABLED | DISABLED)
show gslb service <name>
<!--NeedCopy-->
```

Example:

```
set gslb service Service-GSLB-1 -downStateFlush ENABLED
Done

show gslb service Service-GSLB-1
Done
<!--NeedCopy-->
```

To enable delayed cleanup of virtual server connections by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services** and double-click the service.
2. Click the **Other Settings** section and select the **Down State Flush** option.

Manage local DNS traffic by using DNS policies

You can use DNS policies to implement site affinity by directing traffic from the IP address of a local DNS resolver or network to a predefined target GSLB site. This is configured by creating DNS policies with DNS expressions and binding the policies globally on the Citrix ADC appliance.

DNS expressions

The Citrix ADC appliance provides certain predefined DNS expressions that can be used for configuring actions specific to a domain. Such actions can, for example, drop certain requests, select a specific view for a specific domain, or redirect certain requests to a specific location.

These DNS expressions (also called *rules*) are combined to create DNS policies that are then bound globally on the Citrix ADC appliance.

Following is the list of predefined DNS qualifiers available on the Citrix ADC appliance:

- CLIENT.UDP.DNS.DOMAIN.EQ(“domainname”)
- CLIENT.UDP.DNS.IS_AREC
- CLIENT.UDP.DNS.IS_AAAAREC
- CLIENT.UDP.DNS.IS_SRVREC
- CLIENT.UDP.DNS.IS_MXREC
- CLIENT.UDP.DNS.IS_SOAREC
- CLIENT.UDP.DNS.IS_PTRREC
- CLIENT.UDP.DNS.IS_CNAME
- CLIENT.UDP.DNS.IS_NSREC
- CLIENT.UDP.DNS.IS_ANYREC

The CLIENT.UDP.DNS.DOMAIN DNS expression can be used with string expressions. If you are using domain names as part of the expression, they must end with a period (.). For example, CLIENT.UDP.DNS.DOMAIN.ENDSWITH(“abc.com.”)

To create an expression by using the configuration utility

1. Click the icon next to the Expression text box. Click Add. (Leave the Flow Type and Protocol drop-down list boxes empty.) Follow these steps to create a rule.
2. In the Qualifier box, select a qualifier (for example, LOCATION).
3. In the Operator box, select an operator (for example, ==).
4. In the Value box, type a value (for example, Asia, Japan….).
5. Click OK. Click Create and click Close. The rule is created.
6. Click OK.

Configure DNS actions

A DNS policy includes the name of a DNS action to be performed when the policy rule evaluates to TRUE. A DNS action can do one of the following:

- Send the client an IP address for which you have configured a DNS view. For more information about DNS views, see Adding DNS Views.
- Send the client the IP address of a GSLB service after referring to a list of preferred locations that overrides static proximity behavior. For more information about preferred locations, see Overriding Static Proximity Behavior by Configuring Preferred Locations.
- Send the client a specific IP address as determined by the evaluation of the DNS query or response (DNS response rewrite).
- Forward a request to the name server without performing a lookup in the appliance’s DNS cache.
- Drop a request.

You cannot create a DNS action for dropping a DNS request or for bypassing the DNS cache on the appliance. If you want to drop a DNS request, use the built-in action, dns_default_act_Drop. If you want to bypass the DNS cache, use the built-in action, dns_default_act_Cachebypass. Both actions are available along with custom actions in the Create DNS Policy and the Configure DNS Policy dialog boxes. These built-in actions cannot be modified or removed.

To configure a DNS action by using the command line interface

At the command prompt, type the following commands to configure a DNS action and verify the configuration:

```
add dns action <actionName> <actionType> (-IPAddress <ip_addr |
    ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
    ...) [-TTL <secs>]

show dns action [<actionName>]
<!--NeedCopy-->
```

Examples

Example 1: Configuring DNS Response Rewrite. The following DNS action sends the client a preconfigured IP address when the policy to which the action is bound evaluates to true:

```
add dns action dns_act_response_rewrite Rewrite_Response -IPAddress
    192.0.2.20 192.0.2.56 198.51.100.10
 Done

show dns action dns_act_response_rewrite
1)  ActionName:  dns_act_response_rewrite ActionType:  Rewrite_Response
     TTL:  3600               IPAddress:  192.0.2.20 192.0.2.56
    198.51.100.10
 Done
<!--NeedCopy-->
```

Example 2: Configuring a DNS-View Based Response. The following DNS action sends the client an IP address for which you have configured a DNS view:

```
add dns action send_ip_from_view_internal_ip ViewName -viewName
    view_internal_ip
Done

show dns action send_ip_from_view_internal_ip
1)  ActionName:  send_ip_from_view_internal_ip ActionType:  ViewName
      ViewName:  view_internal_ip
```



```
6   Done
7  <!--NeedCopy-->
```

Example 3: Configuring a Response Based on a Preferred Location List. The following DNS action sends the client the IP address that corresponds to the preferred location that it selects from the specified list of locations:

```
1  add dns action send_preferred_location GslbPrefLoc -preferredLocList NA
      .tx.ns1.*.*.* NA.tx.ns2.*.*.* NA.tx.ns3.*.*.*
2  Done
3
4  show dns action  send_preferred_location
5  1)  ActionName:  send_preferred_location ActionType:  GslbPrefLoc
      PreferredLocList:  "NA.tx.ns1.*.*.*" "NA.tx.ns2.*.*.*" "NA.tx.ns3
      .*.*.*"
6   Done
7  <!--NeedCopy-->
```

To configure a DNS action by using the Citrix ADC configuration utility

1. Navigate to Traffic Management > DNS > Actions, create or edit a DNS action.
2. In the Create DNS Action or Configure DNS Action dialog box, set the following parameters:
   - Action Name (cannot be changed for an existing DNS action)
   - Type (cannot be changed for an existing DNS action)
     To set the
     Type parameter, do one of the following:
     - To create a DNS action that is associated with a DNS view, select View Name. Then, from the View Name list, select the DNS view that you want to use in the action.
     - To create a DNS action with a preferred location list, select Preferred Location List. In Preferred Location, enter a location, and then click Add. Add as many DNS locations as you want.
     - To configure a DNS action for rewriting a DNS response on the basis of policy evaluation, select Rewrite Response. In IP Address, enter an IP address, and then click Add. Add as many IP addresses as you want.
   - TTL (applicable only to the Rewrite Response action type)

Configure DNS policies

DNS policies operate on a location database that uses static and custom IP addresses. The attributes of the incoming local DNS request are defined as part of an expression, and the target site is defined as part of a DNS policy. While defining actions and expressions, you can use a pair of single quotation

marks (‘’) as a wildcard qualifier to specify more than one location. When a DNS policy is configured and a GSLB request is received, the custom IP address database is first queried for an entry that defines the location attributes for the source:

- When a DNS query comes from an LDNS, the characteristics of the LDNS are evaluated against the configured policies. If they match, an appropriate action (site affinity) is executed. If the LDNS characteristics match more than one site, the request is load balanced between the sites that match the LDNS characteristics.
- If the entry is not found in the custom database, the static IP address database is queried for an entry, and if there is a match, the above policy evaluation is repeated.
- If the entry is not found in either the custom or static databases, the best site is selected and sent in the DNS response on the basis of the configured load balancing method.

The following restrictions apply to DNS policies created on the Citrix ADC appliance.

- A maximum of 64 policies are supported.
- DNS policies are global to the Citrix ADC appliance and cannot be applied to a specific virtual server or domain.
- Domain or virtual server specific binding of policy is not supported.

You can use DNS policies to direct clients that match a certain IP address range to a specific site. For example, if you have a GSLB setup with multiple GSLB sites that are separated geographically, you can direct all clients whose IP address is within a specific range to a particular data center.

Both TCP-based and UDP-based DNS traffic can be evaluated. Policy expressions are available for UDP-based DNS traffic on the server and for both UDP-based DNS traffic and TCP-based DNS traffic on the client side. Additionally, you can configure expressions to evaluate queries and responses that involve only the following DNS question types (or QTYPE values):

- A
- AAAA
- NS
- SRV
- PTR
- CNAME
- SOA
- MX
- ANY

The following response codes (RCODE values) are also supported:

- NOERROR - No error
- FORMERR - Format error
- SERVFAIL - Server failure

- NXDOMAIN - Non-existent domain
- NOTIMP - Query type not implemented
- REFUSED - Query refused

You can configure expressions to evaluate DNS traffic. A DNS expression begins with the DNS.REQ or DNS.RES prefixes. Functions are available for evaluating the queried domain, the query type, and the carrier protocol. For more information about DNS expressions, see "Expressions for Evaluating a DNS Message and Identifying Its Carrier Protocol" in "Policy Configuration and Reference".

To add a DNS policy by using the command line interface

At the command prompt, type the following commands to create a DNS policy and verify the configuration:

```
add dns policy <name> <rule> <actionName>
show dns policy <name>
<!--NeedCopy-->
```

Example:

```
> add dns policy policy-GSLB-1 'CLIENT.UDP.DNS.DOMAIN.EQ("domainname")'
    my_dns_action
 Done
> show dns policy policy-GSLB-1
    Name: policy-GSLB-1
    Rule: CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
    Action Name: my_dns_action
    Hits: 0
    Undef Hits: 0

 Done
<!--NeedCopy-->
```

To remove a configured DNS policy by using the command line interface

At the command prompt, type:

```
rm dns policy <name>
<!--NeedCopy-->
```

To configure a DNS policy by using the Citrix ADC configuration utility

1. Navigate to Traffic Management > DNS > Policies and create a DNS policy.

2. In the Create DNS Policy or Configure DNS Policy dialog box, set the following parameters:
   - Policy Name (cannot be changed for an existing policy)
   - Action
   - Expression

     To specify an expression, do the following:
     a) Click Add, and then, in the drop-down box that appears, select the expression element with which you want to begin the expression. A second list appears. The list contains a set of expression elements that you can use immediately after the firs expression element.
     b) In the second list, select the expression element that you want, and then enter a period.
     c) After each selection, if you enter a period, the next set of valid expression elements appear in a list. Select expression elements and fill in arguments to functions until you have the expression you want.
3. Click Create or OK, and then click Close.

Bind DNS policies

DNS policies are bound globally on the Citrix ADC appliance and are available for all configured GSLB virtual servers. Even though DNS policies are globally bound, policy execution can be limited to a specific GSLB virtual server by specifying the domain in the expression.

Note: Even though the bind dns global command accepts REQ_OVERRIDE and RES_OVERRIDE as valid bind points, those bind points are redundant, because DNS policies can be bound only globally. Bind your DNS policies only to the REQ_DEFAULT and RES_DEFAULT bind points.

To bind a DNS policy globally by using the command line interface

At the command prompt, type the following commands to bind a DNS policy globally and verify the configuration:

```
bind dns global <policyName> <priority> [-gotoPriorityExpression <
    string>] [-type <type>]
show dns global -type <type>
<!--NeedCopy-->
```

Example:

```
bind dns global policy-GSLB-1 10 -gotoPriorityExpression END
 Done
show dns global -type REQ_DEFAULT
1)  Policy Name: policy-GSLB-1
```



```
    Priority: 10
    GotoPriorityExpression: END
 Done
<!--NeedCopy-->
```

To bind a DNS policy globally by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the Bind/Unbind DNS Policy(s) to Global dialog box, click Insert Policy.
4. In the Policy Name column, select, from the list, the policy that you want to bind. Alternatively, in the list, click New Policy, and then create a DNS policy by setting parameters in the Create DNS Policy dialog box.
5. To modify a policy that is already bound globally, click the name of the policy, and then click Modify Policy. Then, in the Configure DNS Policy dialog box, modify the policy, and then click OK.
6. To unbind a policy, click the name of the policy, and then click Unbind Policy.
7. To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
8. To regenerate assigned priorities, click Regenerate Priorities. The priority values are modified to begin at 100, with increments of 10, without affecting the order of evaluation.
9. Click OK.

To view the global bindings of a DNS policy by using the command line interface

At the command prompt, type:

```
show dns global
```

To view the global bindings of a DNS policy by using the configuration utility

1. Navigate to **Traffic Management** > **DNS** > **Policies**.
2. In the details pane, click **Global Bindings**. The global bindings of all DNS policies appear in this dialog box.

Adding DNS Views

You can configure DNS views to identify various types of clients and provide an appropriate IP address to a group of clients who query for the same GSLB domain. DNS views are configured by using DNS policies that select the IP addresses sent back to the client.

For example, if you have configured GSLB for your company's domain and have the server hosted in your company's network, clients querying for the domain from within your company's internal network can be provided with the server's internal IP address instead of the public IP address. Clients that query DNS for the domain from the Internet, on the other hand, can be provided the domain's public IP address.

To add a DNS view, you assign it a name of up to 31 characters. The leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ). After adding the view, you configure a policy to associate it with clients and a part of the network, and you bind the policy globally. To configure and bind a DNS policy, see **Managing Local DNS Traffic by Using DNS Policies**.

To add a DNS view by using the command line interface

At the command prompt, type the following commands to create a DNS view and verify the configuration:

```
1 add dns view <viewName>
2 show dns view <viewName>
3 <!--NeedCopy-->
```

Example:

```
1 add dns view PrivateSubnet
2 show dns view PrivateSubnet
3 <!--NeedCopy-->
```

To remove a DNS view by using the command line interface

At the command prompt, type:

```
1 rm dns view <viewName>
2 <!--NeedCopy-->
```

To add a DNS view by using the configuration utility

Navigate to Traffic Management > DNS > Views and add a DNS view.

For details on how to create a DNS policy and how to bind DNS policies globally, see **Managing Local DNS Traffic by Using DNS Policies**.

Configure GSLB for proximity

September 14, 2021

When you configure GSLB for proximity, client requests are forwarded to the closest data center. The main benefit of the proximity-based GSLB method is faster response times resulting from the selection of the closest available data center. Such a deployment is critical for applications that require fast access to large volumes of data.

You can configure GSLB for proximity based on the round trip time (RTT), static proximity, or a combination of the two.

Configure dynamic round trip time (RTT) method

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the Citrix ADC appliance probes the client's local DNS server and gathers RTT metric information. The appliance then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value

To configure GSLB for proximity with dynamic method, you must first configure the basic GSLB set up and then configure dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

Once you have configured a basic GSLB setup, configure the dynamic RTT method.

For details on how to configure the GSLB virtual server to use the dynamic RTT method for load balancing, see Configuring Dynamic RTT.

Configure static proximity

The static proximity method for GSLB uses an IP address-based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The Citrix ADC appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the Citrix ADC appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

To configure GSLB for proximity with static proximity, you must first configure the basic GSLB set up and then configure static proximity.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

Once you have configured a basic GSLB setup, configure static proximity.

For details on how to configure the GSLB virtual server to use static proximity for load balancing, see Configuring Static Proximity.

Configure static proximity and dynamic RTT

You can configure the GSLB virtual server to use a combination of static proximity and dynamic RTT when you have some clients coming from an internal network like a branch office. You can configure GSLB such that the clients coming from the branch office or any other internal network are directed to a particular GSLB site that is geographically close to the client network. For all other requests, you can use dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

Once you have configured a basic GSLB setup, configure the GSLB virtual server to use static proximity for all traffic originating from an internal network and then use dynamic RTT for all other traffic.

For details on how to configure static proximity, see Configuring Static Proximity and for details on how to configure dynamic RTT, see Configuring Dynamic RTT.

Protect the GSLB setup against failure

September 14, 2021

You can protect your GSLB setup against failure of a GSLB site or a GSLB virtual server by configuring the following:

- A backup GSLB virtual server
- A Citrix ADC appliance to respond with multiple IP addresses

- A backup IP address for a GSLB domain

You can also divert excess traffic to a backup virtual server by using spillover.

Configure a backup GSLB virtual server

Configuring a backup entity for a GSLB virtual server ensures that DNS traffic to a site is not interrupted if the GSLB virtual server goes down. The backup entity can be another GSLB virtual server, or it can be a backup IP address. With a backup entity configured, if the primary GSLB virtual server goes down, the backup entity handles DNS requests. To specify what must happen when the primary GSLB virtual server comes back up again, you can configure the backup entity to continue handling traffic until you manually enable the primary virtual server to take over (using the disablePrimaryOnDown option).

Note: You can configure a single backup entity as backup for multiple GSLB virtual servers.

To configure a backup GSLB virtual server by using the command line interface

At the command prompt, type the following commands to configure a GSLB virtual server as backup virtual server and verify the configuration:

```
set gslb vserver <name> -backupVServer <name> [-disablePrimaryOnDown (
    ENABLED | DISABLED)]

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -
    disablePrimaryOnDown ENABLED
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

To set GSLB virtual server as a backup virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**, and double-click the GSLB virtual server.
2. Select the **Backup Virtual Server** section and choose the backup virtual server.

Configure a GSLB setup to respond with multiple IP addresses

A typical DNS response contains the IP address of the best performing GSLB service. However, if you enable multiple IP responses (MIR), the Citrix ADC appliance sends the best GSLB service as the first

record in the response and adds the remaining active services as extra records. If MIR is disabled (the default), the Citrix ADC appliance sends the best service as the only record in response.

To configure a GSLB virtual server for multiple IP responses by using the command line interface

At the command prompt, type the following commands to configure a GSLB virtual server for multiple IP responses and verify the configuration:

```
set gslb vserver<name> -MIR (ENABLED | DISABLED)
-  show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
 show gslb vserver <vserverName>
<!--NeedCopy-->
```

To set a GSLB virtual server for multiple IP responses by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
2. On the **Advanced** tab, under When this virtual server is "UP," select the Send all "active" service IP in response (MIR) check box, and select **OK**.

Configuring a GSLB Virtual Server to Respond with an Empty Address Record When DOWN

A DNS response can contain either the IP address of the requested domain or an answer stating that the IP address for the domain is not known by the DNS server, in which case the query is forwarded to another name server. These are the only possible responses to a DNS query.

When a GSLB virtual server is disabled or in a DOWN state, the response to a DNS query for the GSLB domain bound to that virtual server contains the IP addresses of all the services bound to the virtual server. However, you can configure the GSLB virtual server to in this case send an empty down response (EDR). When this option is set, a DNS response from a GSLB virtual server that is in a DOWN state does not contain IP address records, but the response code is successful. This prevents clients from attempting to connect to GSLB sites that are down.

Note: You must configure this setting for each virtual server to which you want it to apply.

To configure a GSLB virtual server for empty down responses by using the command line interface

At the command prompt, type:

```
set gslb vserver<name> -EDR (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
 Done
<!--NeedCopy-->
```

To set a GSLB virtual server for empty down responses by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
2. On the Advanced tab, under When this virtual server is “Down,” select the Do not send any service’s IP address in response (EDR) check box.
3. Click **OK**.

Configure a backup IP address for a GSLB domain

You can configure a backup site for your GSLB configuration. With this configuration in place, if all the primary sites go DOWN, the IP address of the backup site is provided in the DNS response.

Typically, if a GSLB virtual server is active, that virtual server sends a DNS response with one of the active site IP addresses as selected by the configured GSLB method. If all the configured primary sites in the GSLB virtual server are inactive (in the DOWN state), the authoritative domain name system (ADNS) server or DNS server sends a DNS response with the backup site’s IP address.

Note: When a backup IP address is sent, persistence is not honored.

To set a backup IP address for a domain by using the command line interface

At the command prompt, type the following commands to set a backup IP address and verify the configuration:

```
set gslb vserver <name> -domainName <string> -backupIP <IPAddress>
show gslb vserver <name>
<!--NeedCopy-->
```


Example:

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP
   10.102.29.66
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

To set a backup IP address for a domain by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the GSLB virtual server to which you want to bind the backup domain (for example, vserver-GSLB-1).
2. Click the **Domains** section, configure the GSLB domain and specify the IP address of the backup domain in the **Backup IP** field.

Divert excess traffic to a backup virtual server

Once the number of connections to a primary GSLB virtual server exceeds the configured threshold value, you can use the spillover option to divert new connections to a backup GSLB virtual server. This threshold value can be calculated dynamically or set manually. Once the number of connections to the primary virtual server drops below the threshold, the primary GSLB virtual server resumes serving client requests.

You can configure persistence with spillover. When persistence is configured, new clients are diverted to the backup virtual server if that client is not already connected to a primary virtual server. When persistence is configured, connections that were diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections to the primary virtual server drops below the threshold. Instead, the backup virtual server continues to process those connections until they are terminated by the user. Meanwhile, the primary virtual server accepts new clients.

The threshold can be measured by the number of connections, bandwidth, and health of the services.

If the backup virtual server reaches the configured threshold and is unable to take any additional load, the primary virtual server diverts all requests to the designated redirect URL. If a redirect URL is not configured on the primary virtual server, subsequent requests are dropped.

The spillover feature prevents the remote backup GSLB service (backup GSLB site) from getting flooded with client requests when the primary GSLB virtual server fails. This occurs when a monitor is bound to a remote GSLB service, and the service experiences a failure that causes its state to go DOWN. The monitor continues to keep the state of the remote GSLB service UP, however, because of the spillover feature.

As part of the resolution to this problem, two states are maintained for a GSLB service, the primary state and effective state. The primary state is the state of the primary virtual server and the effective

state is the cumulative state of the virtual servers (primary and backup chain). The effective state is set to UP if any of the virtual servers in the chain of virtual servers is UP. A flag that indicates that the primary VIP has reached the threshold is also provided. The threshold can be measured by either the number of connections or the bandwidth.

A service is considered for GSLB only if its primary state is UP. Traffic is directed to the backup GSLB service only when all the primary virtual servers are DOWN. Typically, such deployments have only one backup GSLB service.

Adding primary and effective states to a GSLB service has the following effects:

- When source IP persistence is configured, the local DNS is directed to the previously selected site only if the primary virtual server on the selected site is UP and below threshold. Persistence can be ignored in the round robin mode.
- If cookie-based persistence is configured, client requests are redirected only when the primary virtual server on the selected site is UP.
- If the primary virtual server has reached its saturation and the backup VIPs is absent or down, the effective state is set to DOWN.
- If external monitors are bound to an HTTP-HTTPS virtual server, the monitor decides the primary state.
- If there is no backup virtual server to the primary virtual server and the primary virtual server has reached its threshold, the effective state is set to DOWN.

To configure backup GSLB virtual server by using the command line interface

At the command prompt, type the following commands to configure backup GSLB virtual server and verify the configuration:

```
set gslb vserver <name> -soMethod <method> -soThreshold <threshold> -
    soPersistence ( **ENABLED** | **DISABLED** ) -soPersistenceTimeout <
    timeout>
show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000
    -soPersistence ENABLED -soPersistenceTimeout 2
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```


To configure backup GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the virtual server that you want to configure as backup (for example, Vserver-LB-1).
2. Click the **SpillOver** section and set the following parameters:
   - Method— soMethod
   - Threshold— soThreshold
   - Persistence Time-out (min) — soPersistenceTimeout
3. Select the Persistence option and click **OK**.

Configure GSLB for disaster recovery

September 14, 2021

Disaster recovery capability is critical, because downtime is costly. A Citrix ADC appliance configured for GSLB forwards traffic to the least-loaded or the best-performing data center. This configuration, referred to as an active-active setup, not only improves performance, but also provides immediate disaster recovery by routing traffic to other data centers if a data center that is part of the setup goes down. Alternatively, you can configure an active-standby GSLB setup for disaster recovery only.

Configure GSLB for disaster recovery in an active-standby data center setup

A conventional disaster recovery setup includes an active data center and a standby data center. The standby data center is a remote site. When a failover occurs as a result of a disaster event that causes the primary active data center to be inactive, the standby data center becomes operational.

Configuring disaster recovery in an active-standby data-center setup consists of the following tasks.

- Create the active data center.
  - Add a local GSLB site.
  - Add a GSLB vserver, which represents the active data center.
  - Bind the domain to the GSLB virtual server.
  - Add gslb services and bind the services to active GSLB virtual server.
- Create the standby data center.
  - Add a remote gslb site.
  - Add a gslb vserver, which represents standby data center.
  - Add gslb services which represents standby data center and bind the services to the standby gslb vserver.
  - Designate the standby data center by configuring the standby GSLB virtual server as the backup virtual server for the active GSLB virtual server.

Once you have configured the primary data center, replicate the configuration for the backup data center and designate it as the standby GSLB site by designating a GSLB virtual server at that site as the backup virtual server.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

To designate the standby GSLB site by using the command line interface

At both the active site and the remote site, at the command prompt, type:

```
set gslb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
<!--NeedCopy-->
```

To configure the standby site by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server for the primary site.
2. Click the **Backup Virtual Server** section and select a backup virtual server.

By default, once the primary virtual server becomes active, it starts receiving traffic. However, if you want the traffic to be directed to the backup virtual server even after the primary virtual server becomes active, use the
'disable primary on down' option.

Configure for disaster recovery in an active-active data center setup

An active-active GSLB deployment, in which both GSLB sites are active, removes any risk that may arise in having a standby data center. With such a setup, web or application content can be mirrored in geographically separate locations. This ensures that data is consistently available at each distributed data center.

To configure GSLB for disaster recovery in an active-active data center set up, you must first configure the basic GSLB setup on the first data center and then configure all other data centers.

First create at least two GSLB sites. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server in the local site. Finally, at the local site, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Once you have configured the first data center, replicate the configuration for other data centers part of the setup.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

Configuring for Disaster Recovery with Weighted Round Robin

When you configure GSLB to use the weighted round robin method, weights are added to the GSLB services and the configured percentage of incoming traffic is sent to each GSLB site. For example, you can configure your GSLB setup to forward 80 percent of the traffic to one site and 20 percent of the traffic to another. After you do this, the Citrix ADC appliance will send four requests to the first site for each request that it sends to the second.

To set up the weighted round robin method, first create two GSLB sites, local and remote. Next, for the local site create a GSLB virtual server and GSLB services, and bind the services to the virtual server. Configure the GSLB method as round robin. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Each service that represents a physical server in the network has weights associated with it. Therefore the GSLB service is assigned a dynamic weight that is the sum of weights of all services bound to it. Traffic is then split between the GSLB services based on the ratio of the dynamic weight of the particular service to the total weight. You can also configure individual weights for each GSLB service instead of the dynamic weight.

If the services do not have weights associated with them, you can configure the GSLB virtual server to use the number of services bound to it to calculate the weight dynamically.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

Once you configure a basic GSLB setup, you must configure the weighted round robin method such that the traffic is split between the configured GSLB sites according to the weights configured for the individual services.

To configure a virtual server to assign weights to services by using the command line interface

At the command prompt, type one of the following commands, depending upon whether you want to create a new load balancing virtual server or configure an existing one:

```
add lb vserver <name>@ -weight <WeightValue> <ServiceName>
set lb vserver <name>@ -weight <WeightValue> <ServiceName>
<!--NeedCopy-->
```

Example:


```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
<!--NeedCopy-->
```

To set dynamic weight by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
<!--NeedCopy-->
```

To add weights to the GSLB services by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -serviceName GSLBServiceName -weight
   WeightValue
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
<!--NeedCopy-->
```

To configure a virtual server to assign weights to services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and double-click the virtual server (for example, Vserver-LB-1).
2. Click the Services section and set the weight of a service.

To add weights to the GSLB services by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server (for example, vserver-GSLB-1)
2. Click the Services section and set the weight of the service in the Weight field.

To set dynamic weight by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server (for example, vserver-GSLB-1).
2. Click the **Method** section and, from the **Dynamic Weight** drop-down list select **SERVICEWEIGHT**.

Configuring for Disaster Recovery with Data Center Persistence

Data center persistence is required for web applications that require maintaining a connection with the same server instead of having the requests load balanced. For example, in an e-commerce portal, maintaining a connection between the client and the same server is critical. For such applications, HTTP redirect persistence can be configured in an active-active setup.

To configure GSLB for disaster recovery with data center persistence, you must first configure the basic GSLB set up and then configure HTTP redirect persistence.

First create two GSLB sites, local and remote. Next, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Next, create a load balancing virtual server with the same virtual server IP address as the GSLB service. Finally, duplicate the previous steps for the remote configuration, or configure the Citrix ADC appliance to autosynchronize your GSLB configuration.

For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

Once you have configured a basic GSLB setup, configure HTTP redirect precedence to enable data center persistence.

To configure HTTP redirect by using the command line interface

At the command prompt, type the following commands to configure HTTP redirect and verify the configuration:

```
set gslb service <serviceName> -sitePersistence <sitePersistence> -
   sitePrefix <string>
show gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -
   sitePrefix vserver-GSLB-1
show gslb service Service-GSLB-1
<!--NeedCopy-->
```


To configure HTTP redirect by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services and double-click the GSLB service to be configured.
2. Click the **Site Persistence** section, select the **HTTPRedirect** option, and in the **Site Prefix** text box, enter the site prefix (for example, vserver-GSLB-1).

> **Note**
>
> When site persistence is not configured and if a load balancing virtual server that is configured as a local GSLB service is DOWN, the HTTP requests are redirected to other healthy GSLB sites using a 302 redirect.

Override static proximity behavior by configuring preferred locations

September 14, 2021

You might want to direct traffic from a local DNS (LDNS) server or network to a GSLB service other than the GSLB service that the static proximity method selects for that traffic. That is, you have a *preferred location* for that traffic. To override the static proximity method with preferred locations, you can do the following:

1. Configure a DNS action that consists of a list of preferred locations. For more information about configuring a DNS action, see Configuring a DNS Action.
2. Configure a DNS policy to identify the traffic arriving from the LDNS server or network for which you want to override static proximity, and apply the action in the policy.
3. Bind the policy to the global request bind point.

In the DNS action, you can configure a list of up to 8 preferred locations. The locations must be provided in the dotted qualifier notation, which is the notation in which you add custom locations to the static proximity database. The locations can include wildcards for qualifiers that you want to omit. For information about the dotted qualifier notation for locations, see Adding Custom Entries to a Static Proximity Database. When entering the preferred locations, you must enter them in the descending order of priority.

When a policy evaluates to
TRUE, the Citrix ADC appliance matches the preferred locations, in priority order, with the locations of GSLB services. Matches are of the following two types:

- If all the non-wildcard qualifiers in a preferred location match the corresponding qualifiers in the location of a GSLB service, the match is considered a perfect match. For example, a GSLB service location of *.UK.*.* or Europe.UK.*.* is a perfect match for the preferred location *.UK.*.*.

- If only a subset of the non-wildcard qualifiers match, the match is considered a partial match. For example, a GSLB service location of Europe.EG is a partial match for the preferred location Europe.UK.

When a DNS policy evaluates to
TRUE, the following algorithm is used to select a GSLB service:

1. The appliance evaluates the preferred location that has the highest priority and moves down the priority order until a perfect match is found between a preferred location and the location of a GSLB service.

   If a perfect match is found, the appliance checks whether the corresponding GSLB service is up. If it is up, it returns the IP address of the GSLB service in the DNS response. If multiple perfect matches are found (which can happen when one or more wildcards are used in a preferred location), the appliance checks the state of each of the corresponding GSLB services and load balances the GSLB services that are up.

2. If a perfect match is not found for any of the preferred locations, the appliance returns to the preferred location that has the highest priority and moves down the priority order until a partial match is found between a preferred location and the location of a GSLB service.

   If a partial match is found, the appliance checks whether the corresponding GSLB service is up. If it is up, it returns the IP address of the GSLB service in the DNS response. If multiple partial matches are found, the appliance checks the state of each of the corresponding GSLB services and load balances the GSLB services that are up.

3. If none of the perfect and partial matches are up, the appliance load balances all other available GSLB services.

   In this way, the appliance implements a type of site affinity for traffic that matches the DNS policy.

Example

Consider a GSLB configuration that consists of the following eight GSLB services:

- Asia.IN
- Asia.JPN
- Asia.HK
- Europe.UK
- Europe.RU
- Europe.EG
- Africa.SD
- Africa.ZMB

Further consider the following DNS action and policy configuration:


```
> add dns action prefLoc11 GslbPrefLoc -preferredLocList "Asia.HK" "
   Europe.UK"
 Done
> add dns policy dnsPolPrefLoc "CLIENT.IP.SRC.MATCHES_LOCATION("*.ZMB
   .*.*")" prefLoc11
 Done
<!--NeedCopy-->
```

When the appliance receives a request from the location *.ZMB..**, the preferred locations are evaluated as follows:

1. The appliance attempts to find a GSLB service whose location is a perfect match for Asia.HK, which is the preferred location that has the highest priority. It finds that the GSLB service at Asia.HK is a perfect match. If the GSLB service is up, it sends the client the IP address of the GSLB service.
2. If the GSLB service at Asia.HK is down, the appliance attempts to find a perfect match for the second preferred location, Europe.UK. It finds that the GSLB service at Europe.UK is a perfect match. If the GSLB service is up, it sends the client the IP address of the service.
3. If the GSLB service at Europe.UK is down, it returns to the preferred location that has the highest priority, Asia.HK, and looks for partial matches. For Asia.HK, it finds that Asia.IN and Asia.JPN are partial matches. If only one of the corresponding GSLB services is up, it sends the client the IP address of the service. If both locations are up, it load balances the two services.
4. If all partial matches for Asia.HK are down, the appliance looks for partial matches for Europe.UK. It finds that Europe.RU and Europe.EG are partial matches for the preferred location. If only one of the corresponding GSLB services is up, it sends the client the IP address of the service. If both locations are up, it load balances the two services.
5. If all partial matches for Europe.UK are down, the appliance load balances all other available GSLB services. In the current example, the appliance load balances Africa.SD and Africa.ZMB because the remaining six GSLB services have been found to be down.

Configure GSLB service selection using content switching

September 14, 2021

In a typical GSLB deployment, you can prioritize the selection of a set of GSLB services bound to a GSLB virtual server, but you cannot do the following:

- Restrict the selection of a GSLB service from a subset of GSLB services bound to a GSLB virtual server for the given domain.

- Apply different load balancing methods on the different subsets of GSLB services in the deployment.
- Apply spillover policies on a subset of GSLB services, and you cannot have a backup for a subset of GSLB services.
- Configure a subset of GSLB services to serve different content. That is, you cannot content switch between servers in different GSLB sites. The GSLB configuration assumes that the servers contain the same content.
- Define a subset GSLB service with different priorities and specify an order in which the services in the subset are applied to a request.

You can now configure a content switching (CS) policy to customize the GSLB deployment. First configure a set of GSLB services and bind it to a GSLB virtual server. Then, configure a CS virtual server of target type GSLB, define a CS policy and action with the GSLB virtual server as target virtual server, and bind the CS policy to CS virtual server.

> **Important**
>
> - Only CS policies with DNS based expressions can be bound to a CS virtual server of target type GSLB.
> - If a GLSB service is bound to a CS virtual server through a GSLB virtual server, you cannot bind another GSLB virtual server bound with the same GSLB service to the CS virtual server.

Example

Consider a GLSB deployment that includes two GSLB sites. At each site, four GSLB services (S-1, S-2, S-3, and S-4) are bound to GSLB virtual server VS-1. You can configure a content switching (CS) virtual server of target type GSLB and define a CS policy and action with VS-1 as the target virtual server, so that requests for content in English are served only by S-1 and S-2, and requests for content in the local language are served only by S-3 and S-4.

You can give S-1 priority by configuring a backup virtual server to VS-1 and binding S-2 to the backup virtual server. S-1 serves the client requests. If the server S-1 represents goes down, S-2 serves the requests. If both S-1 and S-2 are down, clients receive an empty response.

To configure GSLB Service Selection using Content Switching:

1. Configure GSLB. For instructions, see Configuring Global Server Load Balancing.
2. Configure a Content Switching (CS) virtual server of target type GSLB. For more information, see Creating Content Switching Virtual Servers.
3. Configure Content Switching (CS) policies. For more information, see Configuring Content Switching Policies.
4. Configure CS actions that designate a GSLB virtual server as the target virtual server. For more information, see Configuring a Content Switching Action.
5. Bind the CS policies to the CS virtual server. For more information, see Binding Policies to a Content Switching Virtual Server.

6. Bind the domain to the CS virtual server instead of the GSLB virtual server.

Sample Configuration

The following sample configuration sends requests from the client with IP address 5.5.5.5 to SERVICE_GSLB1 and SERVICE_GSLB2. SERVICE_GSLB1 has a higher priority than SERVICE_GSLB2, and SERVICE_GSLB2 serves the client requests only when SERVICE_GSLB1 is down. If both SERVICE_GSLB1 and SERVICE_GSLB2 are down, SERVICE_GSLB3 and service-GSLB4 are not considered, and a blank response is sent to the client.

```
add cs vs CSVSERVER_GSLB http  -targettype GSLB
Done
add gslb vs VSERVER_GSLB1 http
Done
add gslb vs VSERVER_GSLB2 http
Done
add gslb vs VSERVER_GSLB_BACKUP1 http
Done
set gslb vs VSERVER_GSLB1 -backupvserver VSERVER_GSLB_BACKUP1
Done
add gslb service SERVICE_GSLB1 1.1.1.1 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB2 1.1.1.2 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB3 1.1.1.3 HTTP 80 -sitename site2
Done
add gslb service SERVICE_GSLB4 1.1.1.4 HTTP 80 -sitename site2
Done
bind gslb vs VSERVER_GSLB1 -servicename SERVICE_GSLB1
Done
bind gslb vs VSERVER_GSLB_BACKUP1 -servicename SERVICE_GSLB2
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB3
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB4
Done
add cs action a1 -targetvserver VSERVER_GSLB1
Done
add cs policy p1 -rule "CLIENT.IP.SRC.EQ(5.5.5.5)" -action a1
Done
bind cs vs CSVSERVER_GSLB -domainName www.abc.com
Done
bind cs vs CSVSERVER_GSLB -policyname p1 -priority 1
Done
```



```
add cs action a2 -targetvserver VSERVER_GSLB2
Done
add cs policy p2 -rule "CLIENT.IP.SRC.EQ(6.6.6.6)" -action a2
Done
bind cs vs CSVSERVER_GSLB -policyname p2 -priority 2
Done
<!--NeedCopy-->
```

Associate a target virtual server expression to a GSLB content switching action

You can now associate a target virtual server expression to a GSLB content switching action. This allows GSLB content switching virtual server to use policy expressions to compose the target GSLB virtual server name while processing the DNS requests.

To configure a content switching action that specifies an expression by using the CLI

At the command prompt, type the following command to configure the content switching action to retrieve the HTTP callout response.

```
add cs action <name> -targetVserverExpr <expression>
<!--NeedCopy-->
```

Example:

```
add cs action csact_GSLB_VServer -targetVserverExpr "SYS.HTTP_CALLOUT(
   GSLB_Method_API)"
<!--NeedCopy-->
```

To configure a content switching action that specifies an expression by using the GUI

1. Navigate to **Traffic Management > Content Switching > Actions**.
2. Configure a content switching action, and specify an **Expression** that dynamically computes the name of the target load balancing virtual server.

Configure GSLB for DNS queries with NAPTR records

September 14, 2021

In a typical Global Server Load Balancing (GSLB) deployment, the Citrix ADC appliance receives DNS queries for A/AAAA records, selects the most appropriate GSLB service according to the configured

load balancing method, and returns the service's IP address as a reply to the DNS query. You can now configure the appliance to receive DNS queries for NAPTR records and respond with the list of services configured for a domain. The appliance also monitors the health of the services, and in the response it provides a list of only the services that are up.

Example:

In Telco deployments, you can configure a Citrix ADC appliance to receive DNS queries with NAPTR records from clients such as mobile management entities (MMEs), which play the role of a DNS resolver to discover all the services that are offered by the domain name. The appliance responds to the query with NAPTR records for all the services that are up. The MME can use this NAPTR response to run the S-NAPTR procedure to select the nodes on the basis of the service offered, colocation, topological closeness, and so on.

If multiple nodes qualify for selection, the MME can use the preference field in the NAPTR record from the Citrix ADC appliance to determine the node.

NAPTR Record Format

While responding to a DNS query with NAPTR record, a Citrix ADC appliance constructs a response NAPTR record for each GSLB service.

The following table lists the files in the NAPTR record:

Field	
Domain	The GSLB domain
TTL	The amount of time for which the NAPTR record can be cached.
Class	The class of the record. By default, this value is set to IN.
Type	The DNS record type.
Order	Specifies the order in which the NAPTR record MUST be processed. You can specify the order in the GSLB service. Otherwise, it is set to 1.
Preference	Specifies the order in which NAPTR records with equal "order" values SHOULD be processed, low numbers being processed before high numbers. If the order is not specified in the GSLB service, it is set to 1.

Field	
Flags	Controls the aspects of the rewriting and interpretation of the fields in the record. The Citrix ADC appliance sets this value to A.
Service	Specifies the available service(s).
Regular Expression	Regular expressions are not supported, so this value is set to NULL.
Replacement	The domain name of the node that hosts the services.

Configuration procedure

For detailed GSLB configuration instructions, see Configuring Global Server Load Balancing (GSLB). Make sure that you do the following:

- Set the following parameters while adding the GSLB virtual server:
  - serviceType: ANY
  - dnsRecordType: NAPTR
  - lbMethod: CUSTOMLOAD

Example:

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD
<!--NeedCopy-->
```

- While adding a GSLB site, set the *naptrReplacementSuffix* parameter to the domain name that you want to embed in the NAPTR records.

Example:

```
add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com
<!--NeedCopy-->
```

- Set the following parameters while adding the GSLB service:
  - naptrreplacement
  - naptrOrder
  - naptrServices
  - naptrDomainTTL
  - naptrPreference

Sample configuration

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

Done

add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com

Done

add gslb service sgw1 3.3.3.13 ANY *  -siteName site1 -naptrreplacement
     sgw1.site1. -naptrOrder 2 -naptrServices x-3gpp-sgw:x-s5-gtp -
    naptrDomainTTL 20 -naptrPreference   200

Done

add gslb service sgw2 3.3.3.11 ANY *  -siteName site1 -naptrreplacement
     sgw2.site1. -naptrOrder 5 -naptrServices x-3gpp-sgw:x-s5-gtp  -
    naptrDomainTTL 20  naptrPreference  100

Done

add gslb service sgw3 3.3.3.12 ANY *  -siteName site2 -naptrreplacement
     sgw3.site1. -naptrOrder 10 -naptrServices x-3gpp-sgw:x-s5-gtp  -
    naptrDomainTTL 20 naptrPreference   300

bind gslb vserver gslb_vs -serviceName sgw1

Done

bind gslb vserver gslb_vs -serviceName sgw2

Done

bind gslb vserver gslb_vs -serviceName sgw3

Done

bind gslb service sgw1 -monitorName ping

Done

bind gslb service sgw2 -monitorName ping

Done
```



```
38
39 bind gslb service sgw3 -monitorName ping
40
41 Done
42
43 bind gslb vserver gslb_vs -domainName gslb.com -TTL 5
44
45 Done
46 <!--NeedCopy-->
```

> **Note**
>
> DNS queries with NAPTR records are not supported in parent-child configuration.

Configure GSLB for wildcard domain

September 14, 2021

You can bind a wildcard DNS domain to a GSLB virtual server. Users accessing the applications behind a wildcard domain are routed to the best optimal data center, which hosts those applications. The wildcard domain handles requests for non-existent domains and subdomains. For more information about wildcard domains, see Supporting wildcard DNS domains.

To configure GSLB for a wildcard domain, you must first configure the basic GSLB setup. For details on how to configure a basic GSLB setup, see Configuring GSLB Entities Individually.

To configure a GSLB setup for wildcard domain by using the CLI

Perform the following steps to configure a GSLB setup for wildcard domain:

1. Create the GSLB sites.

```
1 add gslb site site1 10.0.1.10
2 add gslb site site2 20.0.1.10
3 <!--NeedCopy-->
```

2. Add the GSLB services for each site participating in the GSLB setup.

```
1 add gslb service svc1 -sitename site1 10.0.1.10  http 80
2 add gslb service svc2 -sitename site1 10.0.1.10  http 80
3 add gslb service svc3 -sitename site2 20.0.1.10  http 80
4 add gslb service svc4 -sitename site2 20.0.1.10  http 80
5 <!--NeedCopy-->
```


3. Add the GSLB virtual server that references a service being used in the GSLB setup.

```
add gslb vserver gslb_vs http
<!--NeedCopy-->
```

4. Add an ADNS service that listens to the DNS queries.

```
add service adns_udp 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

5. Bind the GSLB services to the GSLB virtual server.

```
bind gslb vserver gslb_vs -service svc1
bind gslb vserver gslb_vs -service svc2
bind gslb vserver gslb_vs -service svc3
bind gslb vserver gslb_vs -service svc4
<!--NeedCopy-->
```

6. Create a zone.

```
add dns soaRec test.com -originServer n1.test.com -contact n1.test
    .com
add dns nsrec test.com n1.test.com
add dns nsrec test.com n2.test.com
add dns zone test.com -proxymode no
<!--NeedCopy-->
```

7. Bind the domain name to the GSLB virtual server.

```
bind gslb vserver gslb_vs -domainName *.test.com
<!--NeedCopy-->
```

Use the EDNS0 client subnet option for Global Server Load Balancing

September 14, 2021

EDNS Client Subnet (ECS) is a Domain Name Server (DNS) header extension that provides the client subnet details. You can use these details to improve the accuracy of Citrix ADC Global Server Load Balancing (GSLB) by using the client network location rather than the DNS resolver location to determine the topological closeness of the client.

> **Note**
>
> Citrix ADC supports only EDNS0.

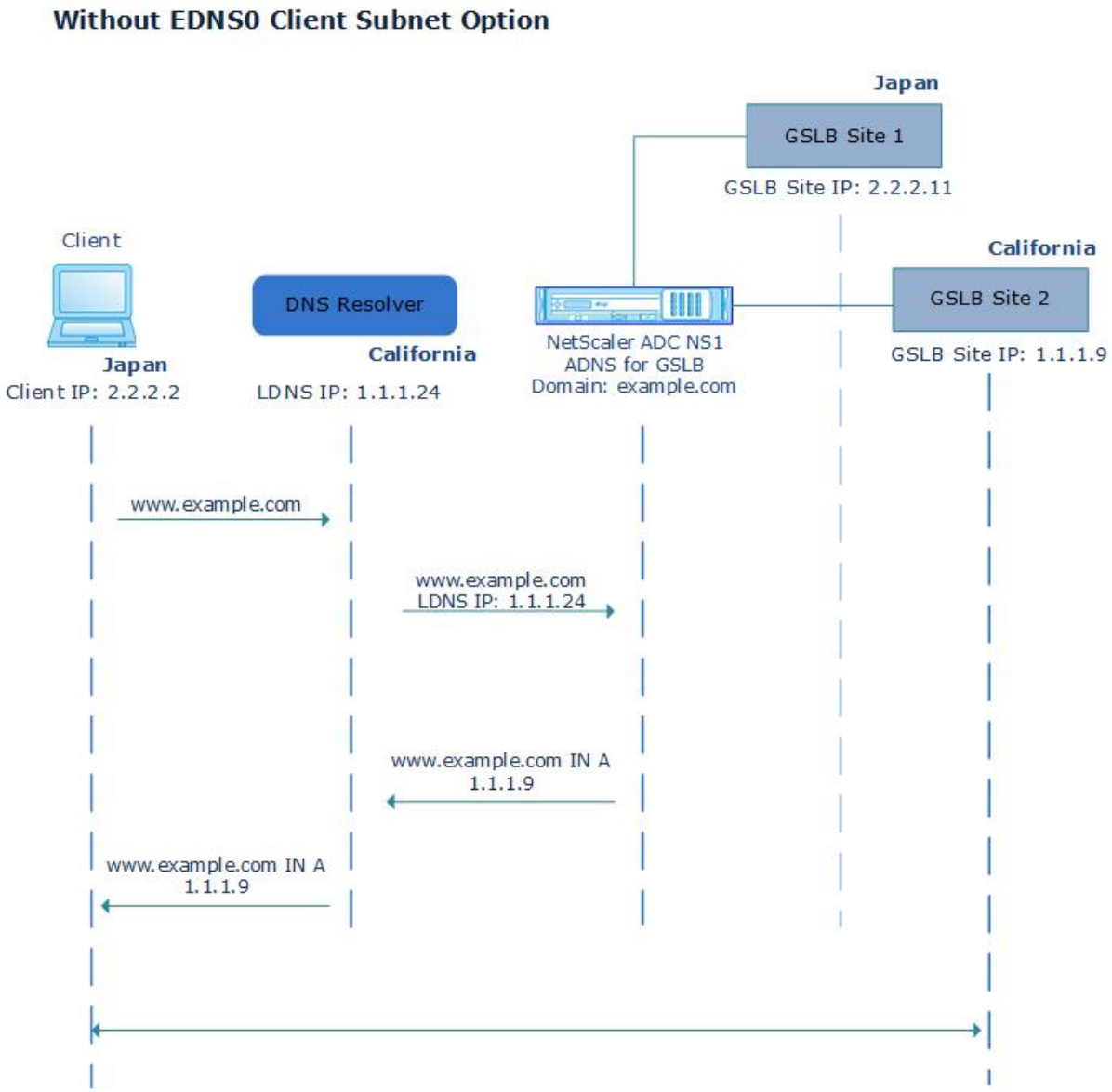
Important:

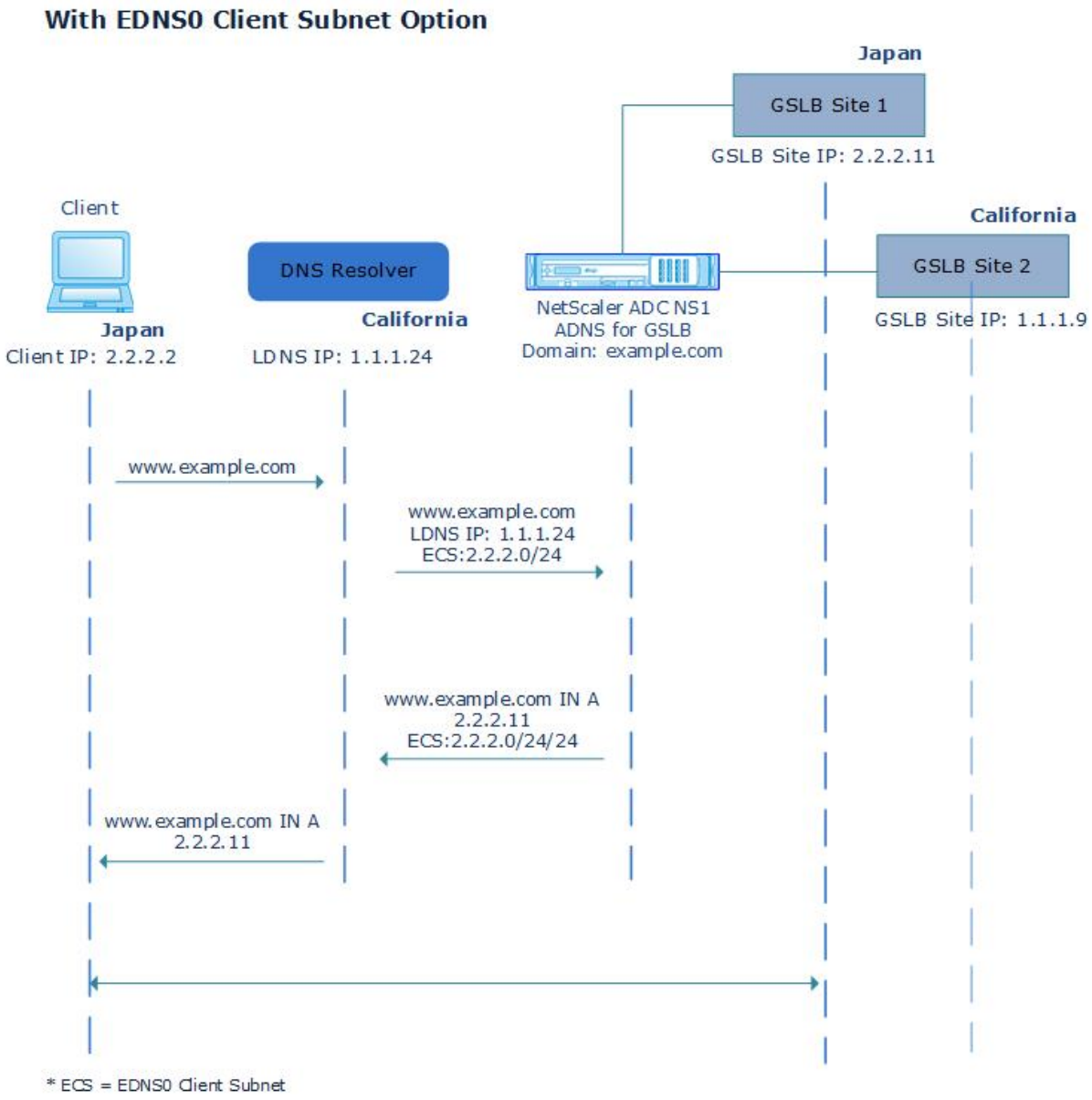
> Make sure that the Local Domain Name Server (LDNS) in your deployment supports the EDNS0 Client Subnet so that the incoming DNS queries contain the EDNS0 Client Subnet option and the Citrix ADC appliance uses the ECS address while processing the DNS query.

The Citrix ADC appliance uses the LDNS IP address for determining the topological closeness of the client and performs GSLB so, when you use proximity-based load balancing methods like static proximity or dynamic round-trip time (RTT). It happens in a typical GSLB deployment. But when a centralized DNS resolver, such as Google DNS or OpenDNS, is involved in the deployment, the Citrix ADC appliance sends the DNS request to a data center close to the centralized DNS resolver, which might not be close to the client. For example, in a typical Citrix ADC GSLB deployment using the static proximity load balancing method, an end-user request from Japan is sent to a data center in Japan and an end user request from California is sent to a data center in California. But if a centralized DNS resolver is involved, the Citrix ADC appliance might send a request from Japan to a data center in California.

You can use the ECS option in deployments that include the Citrix ADC appliance configured as an Authoritative DNS (ADNS) server for a GSLB domain. If you use static proximity as the load balancing method, you can use the IP subnet in the EDNS header instead of the LDNS IP address. This helps to determine the geographical proximity of the client. In proxy mode deployment, the Citrix ADC appliance forwards an ECS-enabled DNS query as-is to the back-end servers. The appliance does not cache ECS-enabled DNS responses.

> **Note**
>
> The ECS option is not applicable for all other deployment modes, such as ADNS mode for non-GSLB domains, resolver mode, and forwarder mode. The ECS option is ignored by the Citrix ADC appliance in the preceding mentioned modes. Also, by default, ECS is disabled for GSLB deployment.





To enable the EDNS0 Client Subnet option by using the command line interface:

At the command prompt, type:

```
set gslb vserver <vserver_name> **-ECS ENABLED

set gslb vserver vserver-GSLB-1 -ECS ENABLED
<!--NeedCopy-->
```


Address validation

You can configure a GSLB virtual server to verify that the address returned by the EDNS0 Client Subnet (ECS) option of the DNS query is not a private or an unroutable IP address. With address validation enabled, the Citrix ADC appliance ignores the ECS address in the DNS query if it is listed in the following table, and instead uses the LDNS IP address for global server load balancing.

> **Note**
>
> By default, address validation is disabled.

Address Type	Address	Description
IPV4	10.0.0.0/8	For private use
	172.16.0.0/12	For private use
	192.168.0.0/16	For private use
	0.0.0.0/8	Refers to the host on the network
	100.64.0.0/10	Shared address space
	127.0.0.0/8	Loopback address
	169.254.0.0/16	Link Local IPv4 address as defined in RFC 3927
	192.0.0.0/24	Used for IETF protocol assignments, includes the private space 192.168.0.0/16
	192.0.2.0/24	Used for documentation purposes
	192.88.99.0/24	Used for 6to4 Relay Anycast
	198.18.0.0/15	Used in Device benchmark testing
	198.51.100.0/24	Used for documentation purposes
	203.0.113.0/24	Used for documentation purposes
	240.0.0.0/4	Used as reserved
	255.255.255.255/32	Used for broadcast

Address Type	Address	Description
IPv6	::1/128	loopback address
	::/128	unspecified address
	::ffff:0:0/96	IPv4-mapped address
	100::/64	discard-only address block
	2001::/23	Used for IETF protocol assignments
	2001::/32	TEREDO
	2001:2::/48	Used for benchmarking
	2001:db8::/32	Used for documentation purposes
	2001:10::/28	ORCHID
	2002::/16	Used for 6to4 Relay Anycast
	fc00::/7	Unique-local
	fe80::/10	Link-local Unicast addresses

To enable address validation by using the command line interface

At the command prompt, type:

```
set gslb vserver <vserver_name> -ecsAddrValidation ENABLED

set gslb vserver vserver-GSLB-1 -ecsAddrValidation ENABLED
<!--NeedCopy-->
```

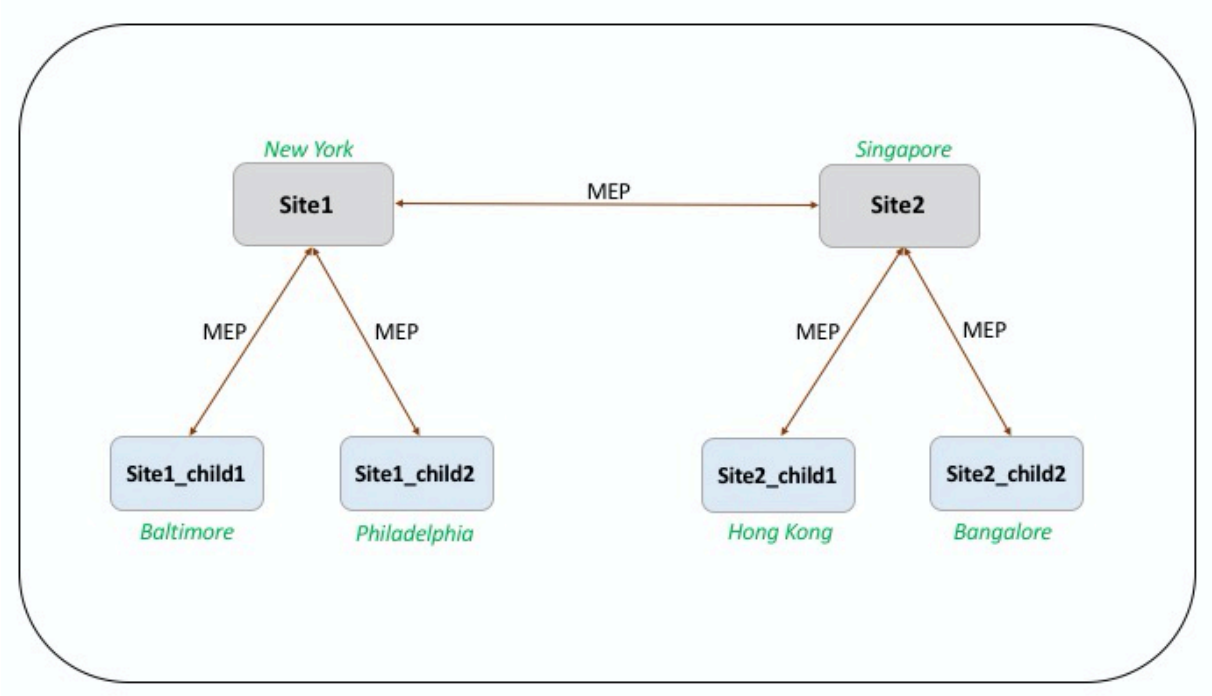
Example of a complete parent-child configuration using the metrics exchange protocol

September 14, 2021

Consider the following parent-child topology in which the GSLB sites are distributed globally.

- Site1 and Site2 are the parent sites.
- Site1_child1 and Site1_child2 are the child sites of Site1.

- Site2_child1 and Site2_child2 are the child sites of Site2.



The following commands illustrate the complete configuration of the parent-child topology.

site1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

```



```
add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
    publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
    site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
     10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
    publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
    site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
     10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
    appflowLog DISABLED

bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1

bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1

bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
<!--NeedCopy-->
```

site1_child1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO
```



```
2
3 add lb vserver lb1 HTTP 10.102.82.132 80 -persistenceType NONE -
     cltTimeout 180
4
5 bind lb vserver lb1 svc1
6 <!--NeedCopy-->
```

site1_child2

```
1 add gslb site site1 10.102.82.164 -publicIP 10.102.82.164
2
3 add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
     nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
     site1
4
5 You can add the following commands for load balancing configuration:
6
7 add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
      -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
     svrTimeout 360 -CKA NO -TCPB NO -CMP NO
8
9 add lb vserver lb1 HTTP 10.102.82.68 80 -persistenceType NONE -
     cltTimeout 180
10
11 bind lb vserver lb1 svc1
12 <!--NeedCopy-->
```

site2

```
1 add gslb site site1 10.102.82.164 -publicIP 10.102.82.164
2
3 add gslb site site2 10.106.24.164 -publicIP 10.106.24.164
4
5 add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
     nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
     site1
6
7 add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
     nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
     site1
8
```



```
add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
    publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
    site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
     10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
    publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
    site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
     10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
    appflowLog DISABLED

bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1

bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1

bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
<!--NeedCopy-->
```

site2_child1

```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
```



```
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.134 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2_child2

```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

Link load balancing

September 14, 2021

Link load balancing (LLB) balances outbound traffic across multiple Internet connections provided by different service providers. LLB enables the Citrix ADC appliance to monitor and control traffic so

that packets are transmitted seamlessly over the best possible link. Unlike with server load balancing, where a service represents a server, with LLB, a service represents a router or the next hop. A link is a connection between the Citrix ADC appliance and the router.

To configure link load balancing, many users begin by configuring a basic setup with default settings. A basic setup involves services, virtual servers, monitors, routes, an LLB method, and persistence (optional). Once a basic setup is operational, you can customize it for your environment.

Load balancing methods that are applicable to LLB are round robin, destination IP hash, least bandwidth, and least packets. You can optionally configure persistence for connections to be sustained on a specific link. The available persistence types are source IP address-based, destination IP address-based, and source IP and destination IP address-based. PING is the default monitor but configuring a transparent monitor is recommended.

You can customize your setup by configuring reverse NAT (RNAT) and backup links.

Configuring a Basic LLB Setup

September 14, 2021

To configure LLB, you first create services representing each router to the Internet Service Providers (ISPs). A PING monitor is bound by default to each service. Binding a transparent monitor is optional but recommended. Then, you create a virtual server, bind the services to the virtual server, and configure a route for the virtual server. The route identifies the virtual server as the gateway to the physical routers represented by the services. The virtual server selects a router by using the load balancing method that you specify. Optionally, you can configure persistence to make sure that all traffic for a particular session is sent over a specific link.

To configure a basic LLB setup, do the following:

- Configure services
- Configure an LLB virtual server and binding a service
- Configure the LLB method and persistence
- Configure an LLB route
- Create and bind a transparent monitor

Configure services

A default monitor (PING) is automatically bound to a service type of ANY when the service is created, but you can replace the default monitor with a transparent monitor, as described in Creating and Binding a Transparent Monitor.

To create a service by using the command line interface

At the command prompt, type:

```
add service <name> <IP> <serviceType> <port>

show service <name>
<!--NeedCopy-->
```

Example:

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: DOWN
        Last state change was at Tue Aug 31 04:31:13 2010
        Time since last state change: 2 days, 05:34:18.600
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 244705  Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

To create services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters:
   - Service Name*—name
   - Server—IP
   - Protocol*—serviceType (Select ANY from the drop-down list.)
   - Port*—port

A required parameter

1. Click Create.
2. Repeat Steps 2-4 to create another service.
3. Click Close.
4. In the Services pane, select the services that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure an LLB virtual server and bind a service

After you create a service, create a virtual server and bind services to the virtual server. The default LB method of least connections is not supported in LLB. For information about changing the LB method, see Configuring the LLB Method and Persistence.

To create a link load balancing virtual server and bind a service by using the command line interface

At the command prompt, type:

```
add lb vserver <name> <serviceType>

bind lb vserver < name> <serviceName>

show lb vserver < name>
<!--NeedCopy-->
```

Example:

```
add lb vserver LLB-vip any
bind lb vserver LLB-vip ISP1R_svc_any
sh lb vserver LLB-vip
```



```
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Thu Sep  2 10:51:32 2010
        Time since last state change: 0 days, 17:51:46.770
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED

1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
 Done
<!--NeedCopy-->
```

To create a link load balancing virtual server and bind a service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and create a virtual server for link load balancing. Specify **ANY** in the **Protocol** field.
2. In the **IP Address Type** drop-down list, select the desired option. Select **Non Addressable** to create a virtual server that is not directly accessible.
3. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.

Configure the LLB method and persistence

By default, the Citrix ADC appliance uses the least connections method to select the service for redirecting each client request, but you should set the LLB method to one of the supported methods. You can also configure persistence, so that different transmissions from the same client are directed to the same server.

To configure the LLB method and/or persistence by using the command line interface

At the command prompt, type the following command:

```
set lb vserver <name> -lbMethod <lbMethod> -persistenceType <
    persistenceType>

```



```
show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver LLB-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 00:52:21.200
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: SOURCEIP
     Persistence Mask: 255.255.255.255       Persistence v6MaskLength:
         128   Persistence Timeout: 2 min
        Connection Failover: DISABLED
<!--NeedCopy-->
```

To configure the link load balancing method and/or persistence by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to configure the load balancing method and/or persistence settings.
2. In the **Advanced Settings** section, select Method and configure the load balancing method.
3. In the **Advanced Settings** section, select **Persistence** and configure the persistence parameters.

Configure an LLB route

After configuring the IPv4 or IPv6 services, virtual servers, LLB methods, and persistence, you configure an IPv4 or IPv6 LLB route for the network specifying the LLB virtual server as the gateway. A route is a collection of links that are load balanced. Requests are sent to the LLB virtual server IP address that acts as the gateway for all outbound traffic and selects the router based on the LLB method configured.

To configure an IPv4 LLB route by using the command line interface

At the command prompt, type:

```
add lb route <network> <netmask> <gatewayName>

show lb route [<network> <netmask>]
<!--NeedCopy-->
```

Example:

```
add lb route 0.0.0.0 0.0.0.0 LLB-vip
show lb route 0.0.0.0 0.0.0.0
       Network          Netmask        Gateway/VIP             Flags
     -----------      -------------   --------------          --------
1)     0.0.0.0          0.0.0.0        LLB-vip               UP
<!--NeedCopy-->
```

To configure an IPv6 LLB route by using the command line interface

At the command prompt, type:

```
add lb route6 <network> <gatewayName>

show lb route6
<!--NeedCopy-->
```

Example:

```
add lb route6 ::/0 llb6_vs show lb route6 Network VIP Flags -----------
    --------- -------- 1) ::/0 llb6_vs UP
<!--NeedCopy-->
```

To configure an LLB route by using the configuration utility

Navigate to System > Network > Routes, and select **LLB**, and configure the LLB route.

Note: Select LLBV6 to configure an IPV6 route.

To configure an LLB route by using the configuration utility

1. Navigate to System > Network > Routes.
2. In the details pane, select one of the following:

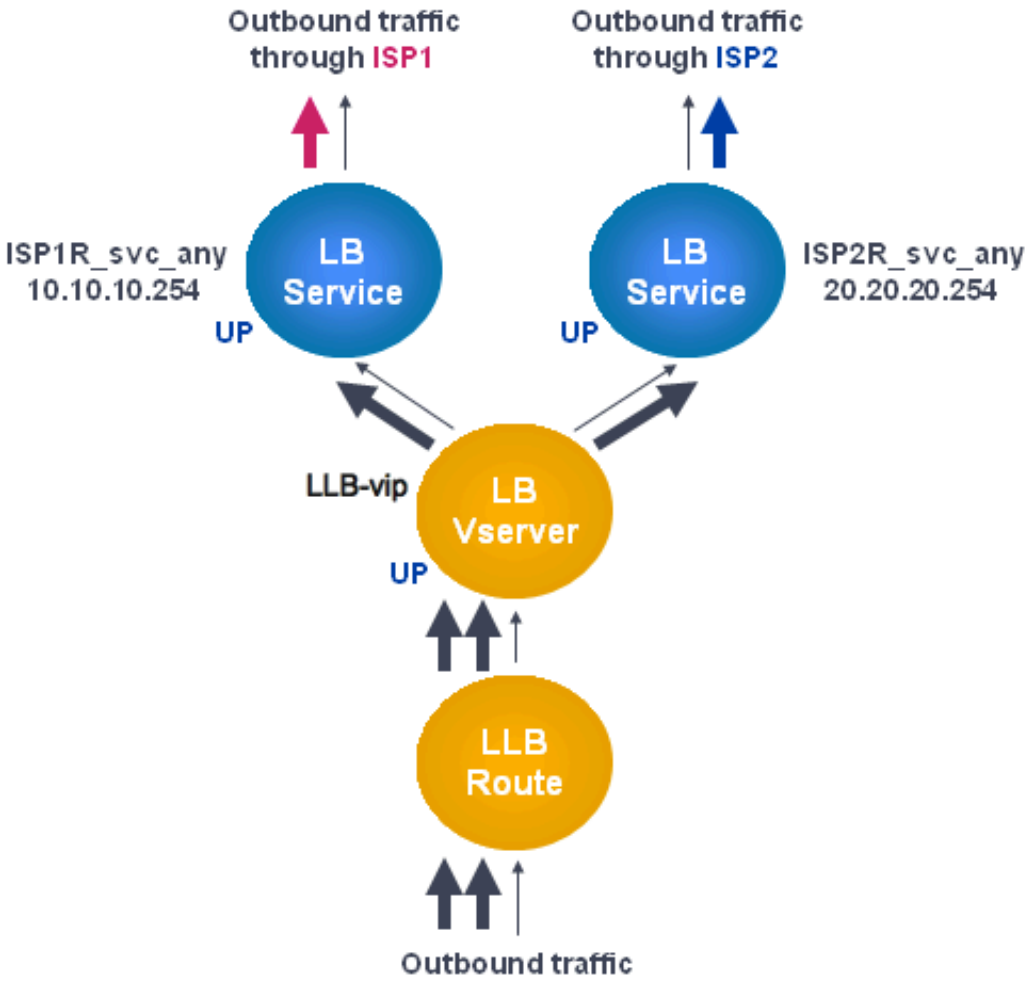
- Click LLB to configure an IPv4 route.
- Click LLBV6 to configure an IPv4 route.

3. In the Create LB Route or Create LB IPV6 Routedialog box, set the following parameters:
   - Network*
   - Netmask*—Required for IPV4 routes.
   - Gateway Name*—gatewayName

     *A required parameter
4. Click Create, and then click Close. The route that you just created appears on the LLB or the LLB6 tab in the Routes pane.

The following diagram shows a basic LLB setup. A service is configured for each of the two links (ISPs) and PING monitors are bound by default to these services. A link is selected based on the LLB method configured.

Figure 1. Basic LLB Setup



> **Note**
>
> If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

Create and bind a transparent monitor

You create a transparent monitor to monitor the health of upstream devices, such as routers. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the Citrix ADC appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the router is UP but one of the next hop devices from that router is down, the appliance includes the router while performing load balancing and forwards the packet to the router. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor,

if any of the devices (including the router) are down, the service is marked as DOWN and the router is not included when the appliance performs link load balancing.

To create a transparent monitor by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent
   YES

show lb monitor [<monitorName>]
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)   Name.......: monitor-1  Type......:      PING   State....:
   ENABLED
Standard parameters:
  Interval.........:            5 sec   Retries..........:
                    3
  Response timeout.:            2 sec   Down time........:
      30 sec
  Reverse..........:              NO   Transparent.......:
                  YES
  Secure...........:              NO   LRTM.............:
      ENABLED
  Action...........:   Not applicable   Deviation........:
      0 sec
  Destination IP...:      10.10.10.11
  Destination port.:    Bound service
  Iptunnel.........:              NO
  TOS..............:              NO   TOS ID...........:
                    0
  SNMP Alert Retries:               0     Success Retries..:
                    1
  Failure Retries..:                0
<!--NeedCopy-->
```

To create a transparent monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors and configure a transparent monitor.

To create a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the Monitors pane, click Add.
3. In the Create Monitor dialog box, set the following parameters:
   - Name*
   - Type*
   - Destination IP
   - Transparent

   *A required parameter
4. Click Create, and then click Close.
5. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed in the Details pane are correct.

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. On the **Monitors** tab, under **Available**, select the monitor that you want to bind to the service, and then click **Add**.

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind lb monitor <monitorName> <serviceName>

show service <name>
<!--NeedCopy-->
```

Example:

```
bind lb monitor monitor-HTTP-1 ISP1R_svc_any
 Done
> show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: UP
        Last state change was at Thu Sep  2 10:51:07 2010
        Time since last state change: 0 days, 18:41:55.130
        Server Name: 10.10.10.254
```



```
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP  Weight: 1
                Probes: 1256       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

To bind a monitor to a service by using the configuration utility

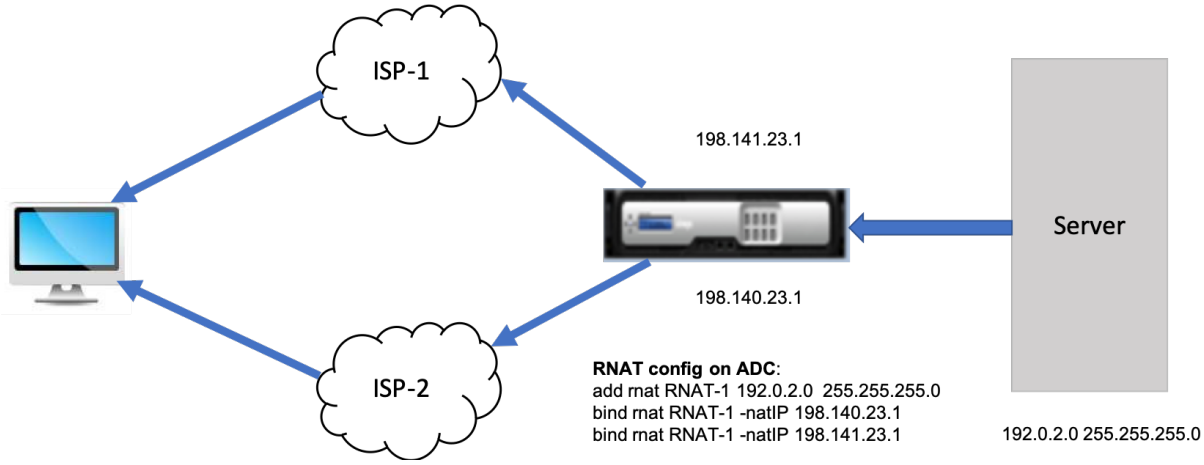
1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select a service to which you want to bind a monitor, and then click Open.
3. In the Configure Service dialog box, on the Monitors tab, under Available, select the monitor that you want to bind to the service, and then click Add.
4. Click OK.
5. In the Services pane, select the service that you just configured and verify that the settings displayed in the Details pane are correct.

Configure RNAT with LLB

September 14, 2021

You can configure an LLB setup for reverse network address translation (RNAT) for outbound traffic. It ensures that the return network traffic for a specific flow is routed through the same path. First configure basic LLB, as described in Configuring a Basic LLB Setup, and then configure RNAT as described in Configure RNAT. Then enable “use subnet IP (USNIP)” mode.

In the following diagram, the Citrix ADC appliance uses LLB to route outbound traffic to different links. During the RNAT operation, the ADC appliance replaces the source IP addresses of the outbound traffic with the public NAT IP address (198.141.23.1) to route the traffic through ISP-1. Similarly, the ADC appliance replaces the source IP addresses with 198.140.23.1 to route the traffic through ISP-2.



To add SNIPs for ISP routers by using the CLI

At the command prompt, type:

```
add NS IP <subnet of first ISP in the IP router> <subnet mask> -type
   SNIP

add NS IP <subnet of second ISP in the IP router> <subnet mask> -type
   SNIP
<!--NeedCopy-->
```

Example:

```
add ns ip 198.140.23.1 255.255.255.0 -type snip

add ns ip 198.141.23.1 255.255.255.0 -type snip
<!--NeedCopy-->
```

To configure RNAT by using the CLI

At the command prompt, type:

```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ ...

```



```
show rnat <name>
<!--NeedCopy-->
```

Example:

```
add rnat RNAT-1 192.0.2.0 255.255.255.0
bind rnat RNAT-1 -natIP 198.140.23.1
bind rnat RNAT-1 -natIP 198.141.23.1

> show rnat RNAT-1
    1) RNAT Name: RNAT-1     Network: 192.0.2.0      Netmask:
       255.255.255.0     Traffic Domain: 0
        UseProxyPort: ENABLED

        NatIP: 198.140.23.1
        NatIP: 198.141.23.1
<!--NeedCopy-->
```

To configure RNAT by using the GUI

1. Navigate to **System** > **Network** > **NATs**.
2. On the **RNAT** tab, click **Configure RNAT**.
3. Specify the network on which to perform RNAT.

> **Note**
>
> You can also configure RNAT by using Access Control Lists (ACLs). Refer Configuring RNAT for details.

To enable Use Subnet IP mode by using the CLI

At the command prompt, type:

```
enable ns mode USNIP

show ns mode
<!--NeedCopy-->
```

Example:

```
enable ns mode USNIP

show ns mode
        Mode                            Acronym             Status
```



```
5          -------                          -------              ------
6    1)    Fast Ramp                        FR                   ON
7    2)    … .
8    8)    Use Subnet IP                    USNIP                ON
9    9)    …
10 <!--NeedCopy-->
```

To enable Use Subnet IP mode by using the GUI

1. Navigate to **System** > **Settings** and, under **Modes and Features**, click **Configure Modes**.
2. In the **Configure Modes** dialog box, select **Use Subnet IP**, and then click **OK**.

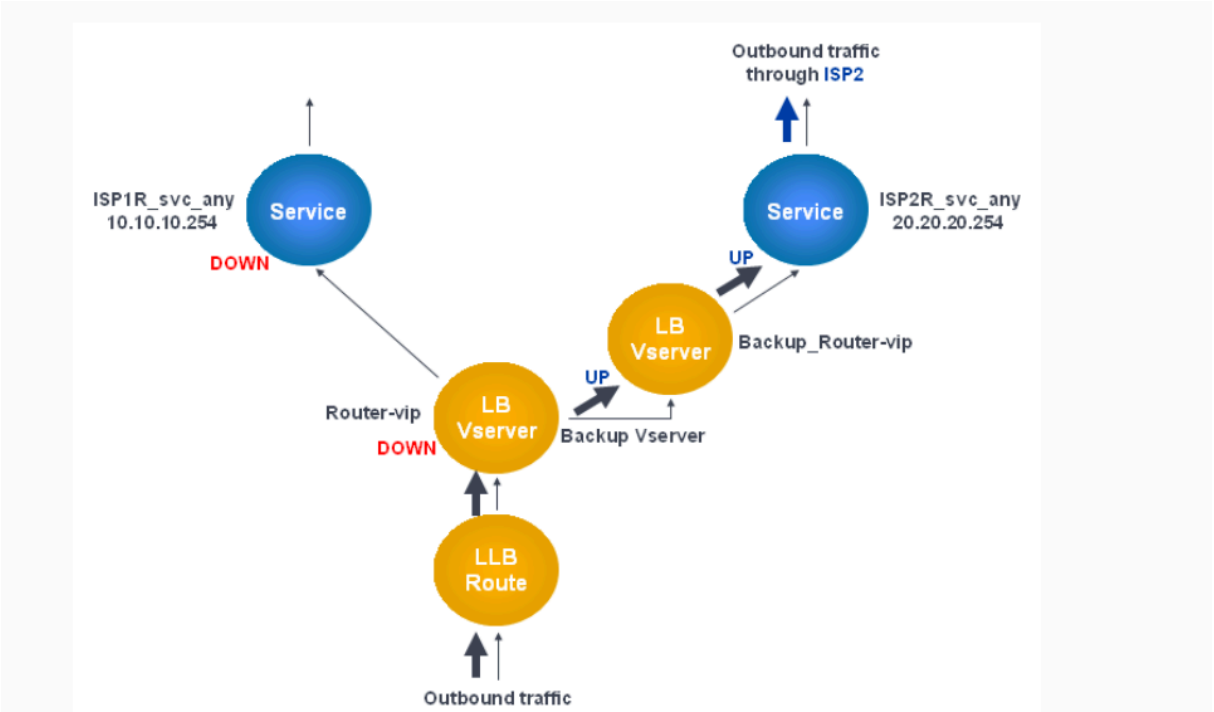
Configure a backup route

September 14, 2021

To prevent disruption in services when the primary route is down, you can configure a backup route. Once the backup route is configured, the Citrix ADC appliance automatically uses it when the primary route fails. First create a primary virtual server as described in Configuring an LLB Virtual Server and Binding a Service. To configure a backup route, create a secondary virtual server similar to a primary virtual server and then designate this virtual server as a backup virtual server (route).

In the following diagram, **Router-vip** is the primary virtual server, and **Backup_Router-vip** is the secondary virtual server designated as the backup virtual server.

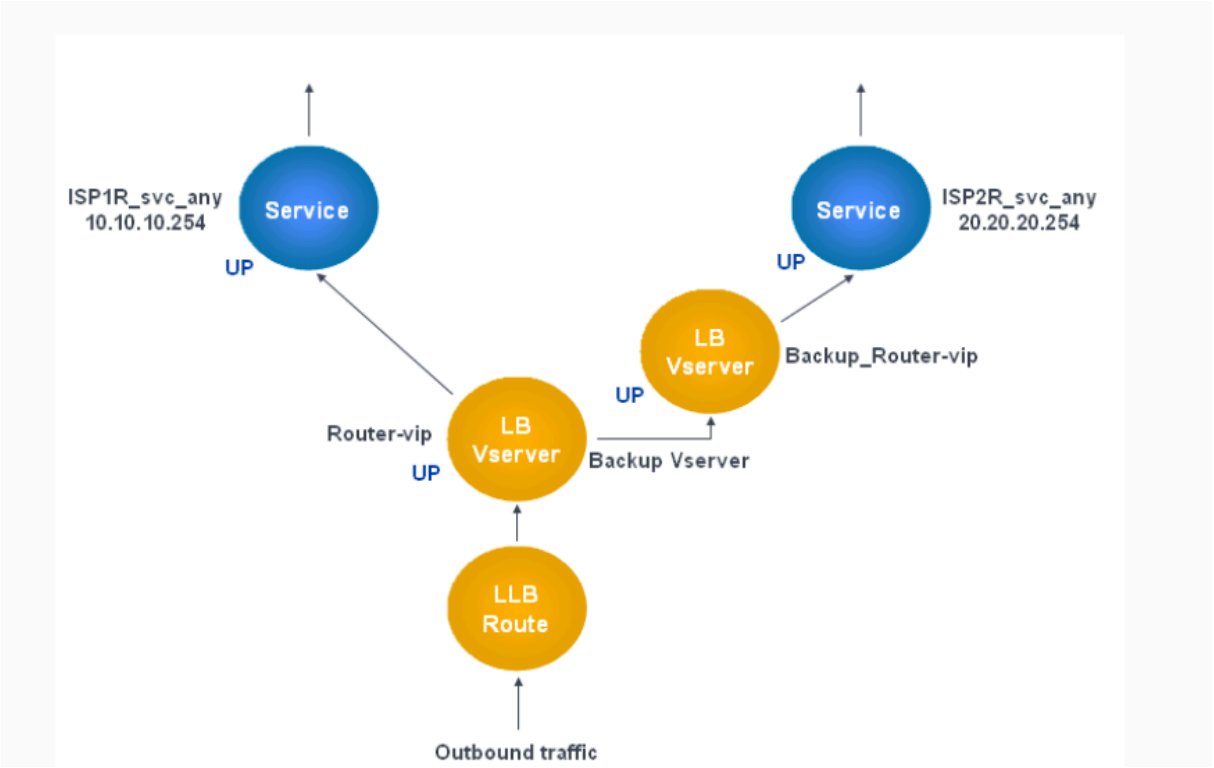
Figure 1. Backup Route Setup



Note: If your ISP has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the preceding figure.

By default, all traffic is sent through the primary route. However, when the primary route fails, all traffic is diverted to the backup route as shown in the following diagram.

Figure 2. Back up Routing in Operation



Note: If your ISP has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the preceding figure.

To set the secondary virtual server as the backup virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

Example:

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 03:09:45.600
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
```



```
12          Configured Method: ROUNDROBIN
13          Mode: IP
14          Persistence: DESTIP     Persistence Mask: 255.255.255.255
                    Persistence v6MaskLength: 128   Persistence Timeout: 2
                min
15          Backup: Router2-vip
16          Connection Failover: DISABLED
17   Done
18  <!--NeedCopy-->
```

To set the secondary virtual server as the backup virtual server by using the configuration utility

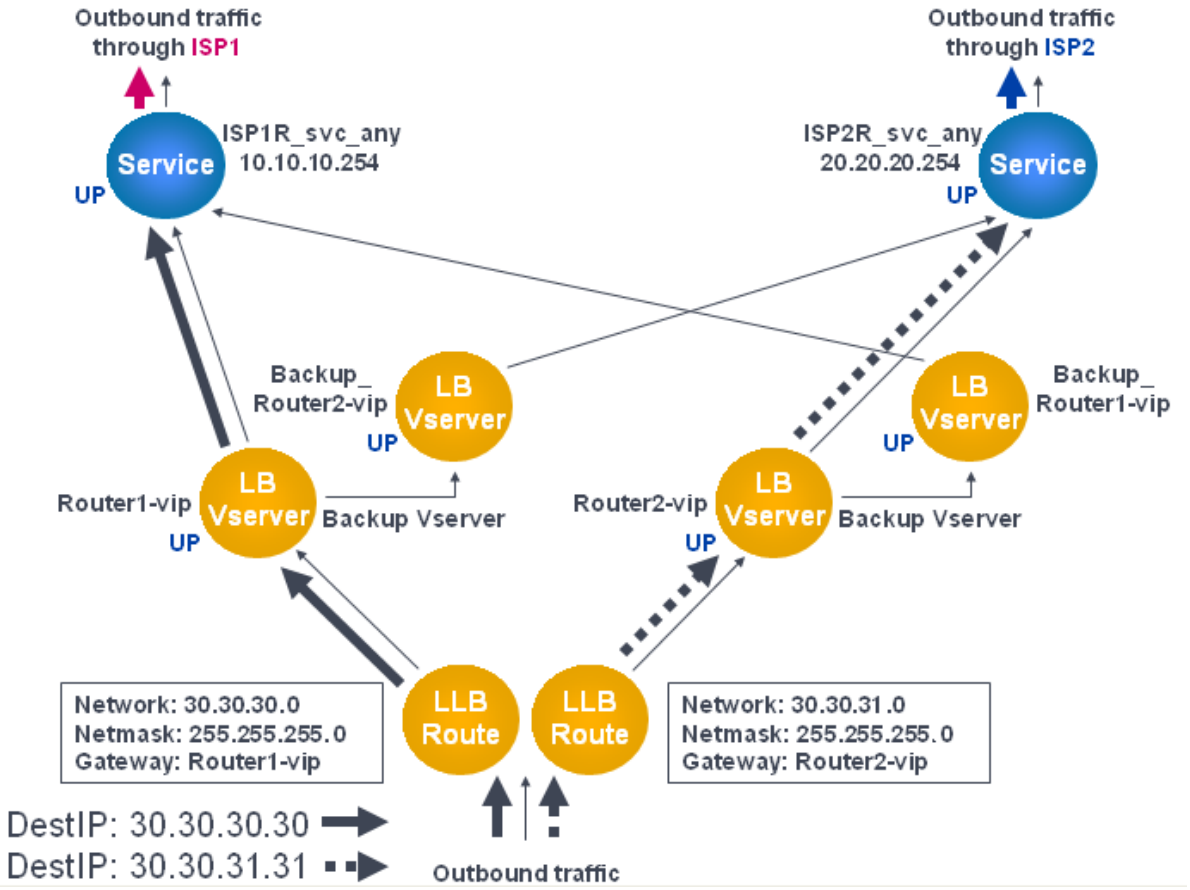
1. Navigate to **Traffic Management > Load Balancing > Virtual Servers** and select the secondary virtual server for which you want to configure the backup virtual server.
2. In the **Load Balancing Virtual Server** dialog box, under **Advanced**, select **Protection**.
3. In the **Backup Virtual Server** drop-down list, select the secondary backup virtual server, and then click **OK**.

Resilient LLB deployment scenario

September 14, 2021

In the following diagram, there are two networks: 30.30.30.0 and 30.30.31.0. Link load balancing is configured based on the destination IP address. Two routes are configured with gateways **Router1-vip** and **Router2-vip**, respectively. **Router1-vip** is configured as a backup to **Router2-vip** and the opposite way. All traffic with the destination IP specified as 30.30.30.30 is sent through **Router1-vip** and traffic with the destination IP specified as 30.30.31.31 is sent through **Router2-vip**.

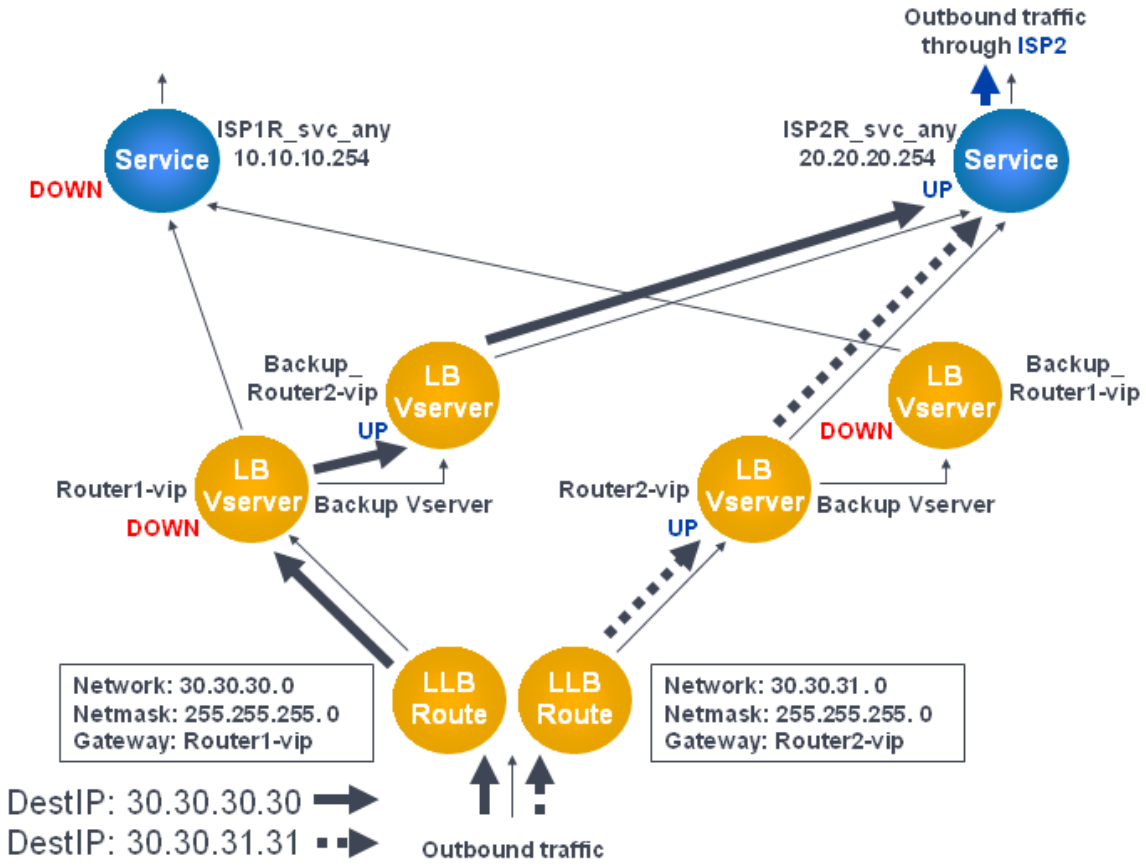
Figure 1. Resilient LLB Deployment Setup



Note: If your ISP has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the preceding figure.

However, if any one of the gateways (**Router1-vip** or **Router2-vip**) is DOWN, traffic is routed through the backup router. In the following diagram, **Router1-vip** for ISP1 is DOWN, so all traffic with the destination IP specified as 30.30.30.30 is also sent through ISP2.

Figure 2. Resilient LLB Deployment Scenario



Note: If your ISP has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the preceding figure.

Monitor an LLB setup

September 14, 2021

After the configuration is up and running, you can view the statistics for each service and virtual server to check for possible problems.

View the statistics of a virtual server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the Citrix ADC appliance. You can display a summary of statistics for all the virtual servers. You can also specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name

- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

Display virtual server statistics by using the CLI

To display a summary of the statistics for all the virtual servers currently configured on the Citrix ADC, or for a single virtual server, at the command prompt, type:

```
stat lb vserver -detail] [<name>]
<!--NeedCopy-->
```

Example:

```
stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
         0/s
Two                          *     0          TCP         DOWN      0/s
         0/s
Three                        *  2598          TCP         DOWN      0/s
         0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
         0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
         0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
         0/s
 Done
<!--NeedCopy-->
```

Display virtual server statistics by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers > Statistics**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

View the statistics of a service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in the surge queue, current server connections, and so forth using the service statistics.

View the statistics of a service by using the CLI

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

View the statistics of a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services > Statistics**.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

Load Balancing

September 14, 2021

The load balancing feature distributes user requests for webpages and other protected applications across multiple servers that all host (or mirror) the same content. You use load balancing primarily to manage user requests to heavily used applications, preventing poor performance and outages and ensuring that users can access your protected applications. Load balancing also provides fault tolerance. When one server that hosts a protected application becomes unavailable, the feature distributes user requests to the other servers that host the same application.

You can configure the load balancing feature to;

- Distribute all requests for a specific protected website, application, or resource between two or more identically configured servers.
- Use any of several different algorithms to determine which server must receive each incoming user request, basing the decision on different factors, such as which server has the fewest current user connections or which server has the lightest load.

The load balancing feature is a core feature of the Citrix ADC appliance. Most users first set up a working basic configuration and then customize various settings, including persistence for connections. In addition, you can configure features for protecting the configuration against failure, managing client traffic, managing and monitoring servers, and managing a large scale deployment.

How load balancing works

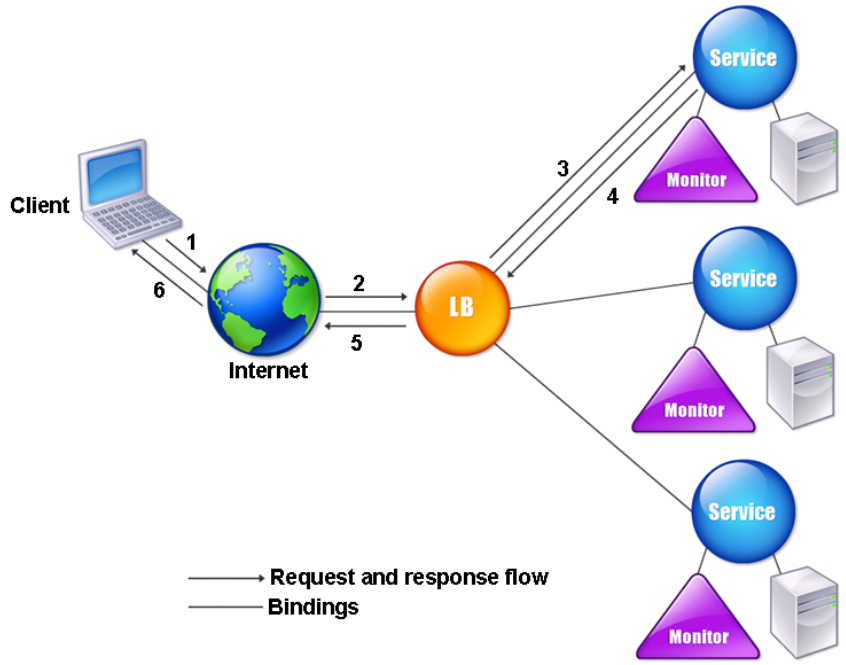
September 14, 2021

In a basic load balancing setup, clients send their requests to the IP address of a virtual server configured on the Citrix ADC appliance. The virtual server distributes them to the load-balanced application servers according to a preset pattern, called the load balancing algorithm. Sometimes, you might want to assign the load balancing virtual server a wildcard address instead of a specific IP address. For instructions about specifying a global HTTP port on the appliance, see **Global HTTP Ports**.

Load balancing basics

A load balancing setup includes a load-balancing virtual server and multiple load-balanced application servers. The virtual server receives incoming client requests, uses the load balancing algorithm to select an application server, and forwards the requests to the selected application server. The following conceptual drawing illustrates a typical load balancing deployment. Another variation involves assigning a global HTTP port.

Figure 1. Load Balancing Architecture



The load balancing virtual server can use several algorithms (or methods) to determine how to distribute load among the load-balanced servers that it manages. The default load balancing method is the least connection method, in which the Citrix ADC appliance forwards each incoming client connection to whichever load-balanced application server currently has the fewest active user connections.

The entities that you configure in a typical Citrix ADC load balancing setup are:

- **Load balancing virtual server.** The IP address, port, and protocol combination to which a client sends connection requests for a particular load-balanced website or application. If the application is accessible from the Internet, the virtual server IP (VIP) address is a public IP address. If the application is accessible only from the LAN or WAN, the VIP is usually a private (ICANN non-routable) IP address.
- **Service.** The IP address, port, and protocol combination used to route requests to a specific load-balanced application server. A service can be a logical representation of the application server itself, or of an application running on a server that hosts multiple applications. After creating a service, you bind it to a load balancing virtual server.
- **Server object.** A virtual entity that enables you to assign a name to a physical server instead of identifying the server by its IP address. If you create a server object, you can specify its name instead of the server's IP address when you create a service. Otherwise, you must specify the server's IP address when you create a service, and the IP address becomes the name of the

server.

- **Monitor.** An entity on the Citrix ADC appliance that tracks a service and ensures that it is operating correctly. The monitor periodically probes (or performs a health check on) each service to which you assign it. If the service does not respond within the time specified by the time-out, and a specified number of health checks fail, that service is marked DOWN. The Citrix ADC appliance then skips that service when performing load balancing, until the issues that caused the service to quit responding are fixed.

The virtual server, services, and load balanced application servers in a load balancing setup can use either Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) IP addresses. You can mix IPv4 and IPv6 addresses in a single load balancing setup.

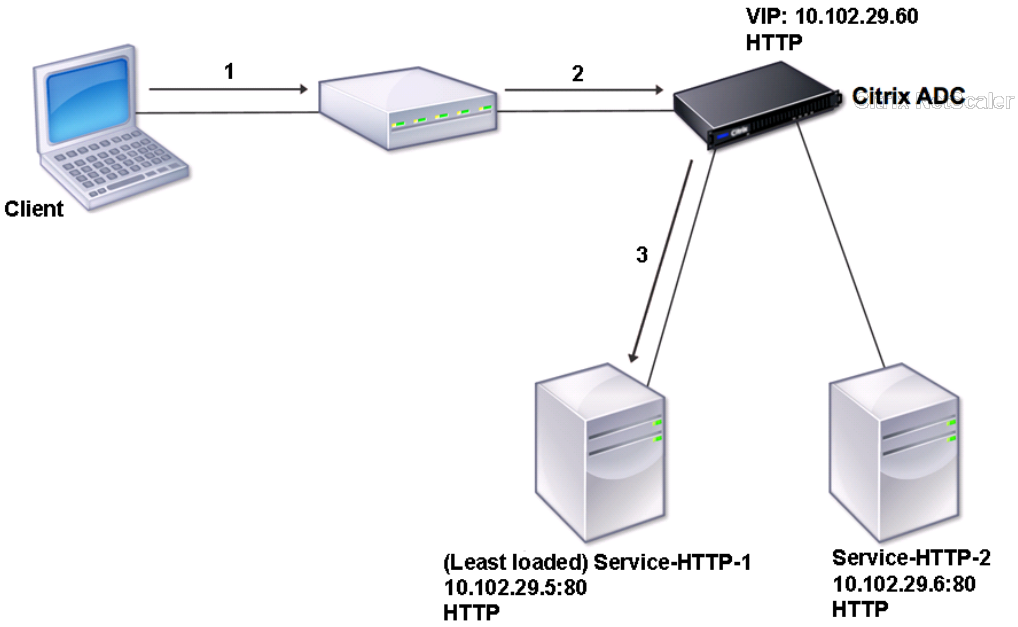
For variations in the load balancing setup, see the following use cases:

- Configuring Load Balancing in Direct Server Return Mode
- Configuring LINUX Servers in DSR Mode
- Configuring DSR Mode When Using TOS
- Configuring Load Balancing in DSR Mode by Using IP Over IP
- Configuring Load Balancing in One-arm Mode
- Configuring Load Balancing in the Inline Mode
- Load Balancing of Intrusion Detection System Servers
- Load balance remote desktop protocol servers

Understanding the topology

In a load balancing setup, the load balancing server is logically located between the client and the server farm, and manages traffic flow to the servers in the server farm. On the Citrix ADC appliance, the application servers are represented by virtual entities called services. The following diagram shows the topology of a basic load balancing configuration.

Figure 2. Basic Load Balancing Topology

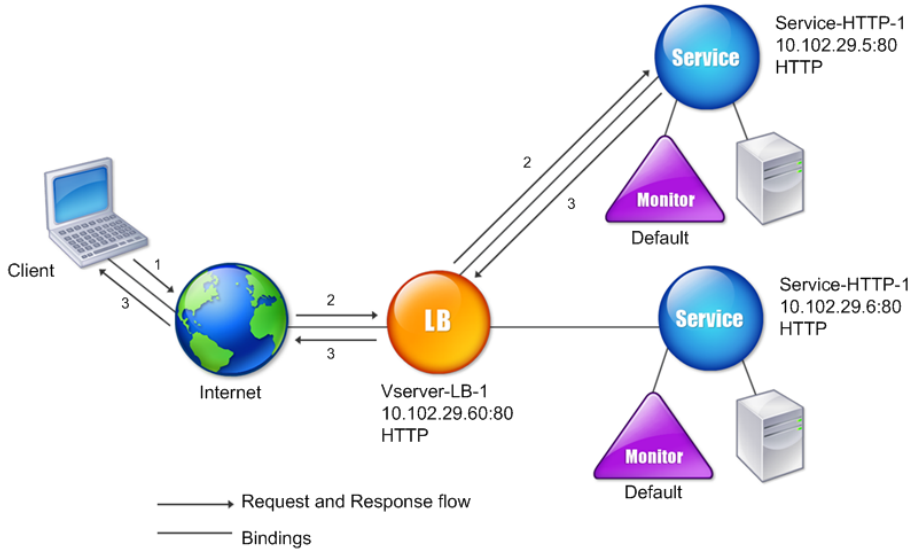


In the diagram, load balancing is used to manage traffic flow to the servers. The virtual server selects the service and assigns it to serve client requests. Consider a scenario where the services Service-HTTP-1 and Service-HTTP-2 are created and bound to the virtual server named Vserver-LB-1. Vserver-LB-1 forwards the client request to either Service-HTTP-1 or Service-HTTP-2. The Citrix ADC appliance uses the least connection load balancing method to select the service for each request. The following table lists the names and values of the basic entities that must be configured on the appliance.

Entity	Name	IPAddress	Port	Protocol
Virtual server	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	80	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP
Monitors	Default	None	None	None

The following diagram shows the load balancing sample values and mandatory parameters that are described in the preceding table.

Figure 3. Load Balancing Entity Model



Use of wildcards instead of IP addresses and ports

Sometimes you might need to use a wildcard for the IP address or the port of a virtual server or for the port of a service. The following cases might require using a wildcard:

- If the Citrix ADC appliance is configured as a transparent pass through, which must accept all traffic that is sent to it regardless of the IP or port to which it is sent.
- If one or more services listen on ports that are not well known.
- If one or more services, over time, change the ports that they listen on.
- If you reach the limit for the number of IP addresses and ports that you can configure on a single Citrix ADC appliance.
- If you want to create virtual servers that listen for all traffic on a specific virtual LAN.

When a wildcard-configured virtual server or service receives traffic, the Citrix ADC appliance determines the actual IP address or port and creates records for the service and associated load balanced application server. These dynamically created records are called dynamically learned server and service records.

For example, a firewall load balancing configuration can use wildcards for both the IP address and port. If you bind a wildcard TCP service to this type of load balancing virtual server, the virtual server

receives and processes all TCP traffic that does not match any other service or virtual server.

The following table describes some of the different types of wildcard configurations and when each must be used.

IP	Port	Protocol	Description
*	*	TCP	A general wildcard virtual server that accepts traffic sent to any IP address and port on the Citrix ADC appliance. When using a wildcard virtual server, the appliance dynamically learns the IP and port of each service and creates the necessary records as it processes traffic.
*	*	TCP	A firewall load balancing virtual server. You can bind firewall services to this virtual server, and the Citrix ADC appliance passes traffic through the firewall to the destination.

IP	Port	Protocol	Description
IP Address	*	TCP, UDP, and ANY	A virtual server that accepts all traffic that is sent to the specified IP address, regardless of the port. You must explicitly bind to this type of virtual server the services to which it will redirect traffic. It does not learn them dynamically. **Note**: You do not configure services or virtual servers for a global HTTP port. In this case, you configure a specific port as a global HTTP port (for example, set ns param -httpPort 80). The appliance then accepts all traffic that matches the port number, and processes it as HTTP traffic. The appliance dynamically learns and creates services for this traffic.

IP	Port	Protocol	Description
*	port	SSL, SSL_TCP	A virtual server that accepts all traffic sent to any IP address on a specific port. Used for global transparent SSL offloading. All SSL, HTTP, and TCP processing that usually is performed for a service of the same protocol type is applied to traffic that is directed to this specific port. The appliance uses the port to dynamically learn the IP of the service it must use. If –cleartext is not specified, the Citrix ADC appliance uses end-to-end SSL.
*	port	Not applicable	All other virtual servers that can accept traffic to the port. You do not bind services to these virtual servers. The Citrix ADC appliance learns them dynamically.

Note: If you have configured your Citrix ADC appliance as a transparent pass through that uses global (wildcard) ports, you might want to turn on Edge mode.
For more information, see "Configuring Edge Mode."

The Citrix ADC appliance attempts to locate virtual servers and services by first attempting an exact

match. If none is found, it continues to search for a match based on wildcards, in the following order:

1. Specific IP address and specific port number
2. Specific IP address and a * (wildcard) port
3. • (wildcard) IP address and a specific port
4. • (wildcard) IP address and a * (wildcard) port

If the appliance is unable to select a virtual server by IP address or port number, it searches for a virtual server based on the protocol used in the request, in the following order:

1. HTTP
2. TCP
3. ANY

Configuring global HTTP ports

You do not configure services or virtual servers for a global HTTP port. Instead, you configure a specific port by using the set ns param command. After configuring this port, the Citrix ADC appliance accepts all traffic that matches the port number, and processes it as HTTP traffic, dynamically learning and creating services for that traffic.

You can configure more than one port number as a global HTTP port. If you are specifying more than one port number in a single set ns param command, separate the port numbers by a single white space. If one or more ports have already been specified as global HTTP ports, and you want to add one or more ports without removing the ports that are currently configured, you must specify all the port numbers, current and new, in the command. Before you add port numbers, use the show ns param command to view the ports that are currently configured.

To configure a global HTTP port by using the command line interface

At the command prompt, type the following commands to configure a global HTTP port and verify the configuration:

```
set ns param -httpPort <port>

show ns param
<!--NeedCopy-->
```

Example 1: Configuring a port as a global HTTP port

In this example, port 80 is configured as a global HTTP port.


```
set ns param -httpPort 80
Done
show ns param
         Global configuration settings:
                          HTTP port(s): 80
                       Max connections: 0
            Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
         Persistence Cookie Secure Flag: ENABLED
         ...
         ...
<!--NeedCopy-->
```

Example 2: Adding ports when one or more global HTTP ports are already configured**

In this example, port 8888 is added to the global HTTP port list. Port 80 is already configured as a global HTTP port.

```
> show ns param
        Global configuration settings:
                          HTTP port(s): 80
                       Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576
        ...
        ...
 Done
> set ns param -httpPort 80 8888
 Done
>  show ns param

        Global configuration settings:
                          HTTP port(s): 80,8888
                       Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576
```



```
25
26          ...
27          ...
28  Done
29 >
30 <!--NeedCopy-->
```

To configure a global HTTP port by using the configuration utility

1. Navigate to **System > Settings > Change HTTP Parameters**, and then add an HTTP port number.

Set up basic load balancing

September 14, 2021

Before configuring your initial load balancing setup, enable the load balancing feature. Then begin by creating at least one service for each server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server, and bind each service to the virtual server. That completes the initial setup. Before proceeding with further configuration, verify your configuration to make sure that each element was configured properly and is operating as expected.

Enabling Load Balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the CLI

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example

```
1      > enable ns feature LoadBalancing
2
3       Done
```



```
4
5      > show ns feature
6
7
8
9              Feature                          Acronym              Status
10
11             -------                          -------              ------
12
13      1)     Web Logging                      WL                   OFF
14
15      2)     Surge Protection                 SP                   ON
16
17      3) Load Balancing LB ON
18
19      .
20
21      .
22
23      .
24
25      24)    NetScaler Push                   push                 OFF
26
27      Done
28 <!--NeedCopy-->
```

To enable load balancing by using the GUI

Navigate to **System** > **Settings** and, in **Configure Basic Features**, select **Load Balancing**.

Configuring a Server Object

Create an entry for your server on the Citrix ADC appliance. The Citrix ADC appliance supports IP address based servers and domain-based servers. If you create an IP address based server, you can specify the name of the server instead of its IP address when you create a service. For information about setting up DNS for a domain-based server, see Domain Name System.

To create a server object by using the CLI

At the command prompt, type:

```
1 add server `<name>`@ `<IPAddress>`@ | `<domain>`
2 <!--NeedCopy-->
```


Example for adding IP address-based name server:

```
add server web_serv 10.102.27.150
<!--NeedCopy-->
```

Example for adding domain-based server:

```
add server web_serv test.com
<!--NeedCopy-->
```

To create a server object by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Servers**, and add a server object.

Configuring Services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the Citrix ADC appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served.

If you create a service without first creating a server object, the IP address of the service is also the name of the server that hosts the service. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

When you create a service that uses UDP as the transport layer protocol, a ping monitor is automatically bound to the service. A ping monitor is the most basic of the built-in monitors. When you create a service that uses TCP as the transport layer protocol, a TCP_default monitor is automatically bound to the service. When you develop a strategy for managing your load balancing setup, you might decide to bind a different type of monitor, or multiple monitors, to the service.

Creating a Service

Before you create a service, you need to understand the different service types and how each is used. The following list describes the types of services supported on the Citrix ADC appliance.

HTTP

Used for load-balanced servers that accept HTTP traffic, such as standard websites and web applications. The HTTP service type enables the Citrix ADC appliance to provide compression, content filtering, caching, and client keep-alive support for your Layer 7 web servers. This service type also

supports virtual server IP port insertion, redirect port rewriting, Web 2.0 Push, and URL redirection support.

Because HTTP is a TCP-based application protocol, you can also use the TCP service type for web servers. If you do so, however, the Citrix ADC appliance is able to perform only Layer 4 load balancing. It cannot provide any of the Layer 7 supports described earlier.

SSL

Used for servers that accept HTTPS traffic, such as ecommerce websites and shopping cart applications. The SSL service type enables the Citrix ADC appliance to encrypt and decrypt SSL traffic (perform SSL offloading) for your secure web applications. It also supports HTTP persistence, content switching, rewrite, virtual server IP port insertion, Web 2.0 Push, and URL redirection.

You can also use the SSL_BRIDGE, SSL_TCP, or TCP service types. If you do so, however, the appliance performs only Layer 4 load balancing. It cannot provide SSL offloading or any of the Layer 7 supports described.

FTP

Used for servers that accept FTP traffic. The FTP service type enables the Citrix ADC appliance to support specific details of the FTP protocol.

You can also use TCP or ANY service types for FTP servers.

TCP

Used for servers that accept many different types of TCP traffic, or that accept a type of TCP traffic for which a more specific type of service is not available.

You can also use the ANY service type for these servers.

SSL_TCP

Used for servers that accept non-HTTP-based SSL traffic, to support SSL offloading.

You can also use the TCP service type for these services. If you do so, the Citrix ADC appliance performs both the Layer 4 load balancing and SSL offloading.

UDP

Used for servers that accept UDP traffic. You can also use the ANY service type.

SSL_BRIDGE

Used for servers that accept SSL traffic when you do not want the Citrix ADC appliance to perform SSL offloading. Alternatively, you can use the SSL_TCP service type.

NNTP

Used for servers that accept Network News Transfer Protocol (NNTP) traffic, typically Usenet sites.

DNS

Used for servers that accept DNS traffic, typically nameservers. With the DNS service type, the Citrix ADC appliance validates the packet format of each DNS request and response. It can also cache DNS responses. You can apply DNS policies to DNS services.

You can also use the UDP service type for these services. If you do, however, the Citrix ADC appliance can only perform Layer 4 load balancing. It cannot provide support for DNS-specific features.

ANY

Used for servers that accept any type of TCP, UDP, or ICMP traffic. The ANY parameter is used primarily with firewall load balancing and link load balancing.

SIP-UDP

Used for servers that accept UDP-based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP).

You can also use the UDP service type for these services. If you do, however, the Citrix ADC appliance performs only Layer 4 load balancing. It cannot provide support for SIP-specific features.

DNS-TCP

Used for servers that accept DNS traffic, where the Citrix ADC appliance acts as a proxy for TCP traffic sent to DNS servers. With services of the DNS-TCP service type, the Citrix ADC appliance validates the packet format of each DNS request and response and can cache DNS responses, as with the DNS service type.

You can also use the TCP service type for these services. If you do, however, the Citrix ADC appliance only performs Layer 4 load balancing of external DNS name servers. It cannot provide support for any DNS-specific features.

RTSP

Used for servers that accept Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data. Select this type to support audio, video, and other types of streamed media.

You can also use the TCP service type for these services. If you do, however, the Citrix ADC appliance performs only Layer 4 load balancing. It cannot parse the RTSP stream or provide support for RTSPID persistence or RTSP NAT.

DHCPRA

Used for servers that accept DHCP traffic. The DHCPRA service type can be used to relay DHCP requests and responses between VLANs.

DIAMETER

Used for load balancing Diameter traffic among multiple Diameter servers. Diameter uses message-based load balancing.

SSL_DIAMETER

Used for load balancing Diameter traffic over SSL.

Services are designated as DISABLED until the Citrix ADC appliance connects to the associated load-balanced server and verifies that it is operational. At that point, the service is designated as ENABLED. Thereafter, the Citrix ADC appliance periodically monitors the status of the servers, and places any that fail to respond to monitoring probes (called health checks) back in the DISABLED state until they respond.

Note: You can create a range of services from a single CLI command or the same dialog box. The names in the range vary by a number used as a suffix/prefix. For example, service1, service2, and so on. From the configuration utility, you can specify a range only in the last octet of the IP address, which is the fourth in case of an IPv4 address and the eighth in an IPv6 address. From the command line, you can specify the range in any octet of the IP address.

QUIC

Used by load balancing servers that accept UDP based QUIC video traffic. The service enables the Citrix ADC appliance to optimize encrypted ABR video traffic over UDP protocol.

To create a service by using the CLI

At the command prompt, type:

```
add service <name> <serverName> <serviceType> <port>

add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

To create a service by using the GUI

1. Navigate to **Traffic Management**> **Load Balancing** > **Services**.
2. In the details pane, click **Add**.
3. In the Create Service dialog box, specify values for the following parameters:
   - Service Name—name
   - Server—serverName
   - Protocol—serviceType
   - Port—port
4. Click **Create**, and then click **Close**. The service you created appears in the Services pane.

Creating a Virtual Server

After you create your services, you must create a virtual server to accept traffic for the load balanced websites, applications, or servers. Once load balancing is configured, users connect to the load-balanced website, application, or server through the virtual server's IP address or FQDN.

Note:

- Virtual server names prefixed with "app_" do not appear in the GUI though they are present in the ns.conf file and are displayed when you run the show command. However, virtual server names prefixed with "app" are displayed in the GUI.
- The virtual server is designated as DOWN until you bind the services that you created to it, and until the Citrix ADC appliance connects to those services and verifies that they are operational. Only then is the virtual server designated as UP.

To create a virtual server by using the CLI

At the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>

```



```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

To create a virtual server by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and then create a virtual server.

Binding Services to the Virtual Server

Note: A service can be bound to a maximum of 500 virtual servers.

After you have created services and a virtual server, you must bind the services to the virtual server. Usually, services are bound to virtual servers of the same type, but you can bind certain types of services to certain different types of virtual servers, as shown below.

Virtual Server Type	Service Type	Comment
HTTP	SSL	You would normally bind an SSL service to an HTTP virtual server to do encryption.
SSL	HTTP	You would normally bind an HTTP service to an SSL virtual server to do SSL offloading.
SSL_TCP	TCP	You would normally bind a TCP service to an SSL_TCP virtual server to do SSL offloading for other TCP (SSL decryption without content awareness).

The state of the services bound to a virtual server determines the state of the virtual server: if all of the bound services are DOWN, the virtual server is marked DOWN, and if any of the bound services is UP or OUT OF SERVICE, the state of the virtual server is UP.

To bind a service to a load balancing virtual server by using the CLI

At the command prompt, type:

```
bind lb vserver <name> <serviceName>

```



```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a service to a load balancing virtual server by using the GUI

1. Navigate to **Traffic Management**> **Load Balancing** > **Virtual Servers**, and select a virtual server.
2. Click in the **Service** section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Verifying the Configuration

After finishing your basic configuration, you can view the properties of each service and load balancing virtual server in your load balancing setup to verify that each is configured correctly. After the configuration is up and running, you can view the statistics for each service and load balancing virtual server to check for possible problems.

Viewing the Properties of a Server Object

You can view properties such as the name, state, and IP address of any server object in your Citrix ADC appliance configuration.

To view the properties of server objects by using the command line interface

At the command prompt, type:

```
show server <serverName>

show server server-1
<!--NeedCopy-->
```

To view the properties of server objects by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Servers**. The parameter values of the available servers appear in the details pane.

Viewing the Properties of a Virtual Server

You can view properties such as the name, state, effective state, IP address, port, protocol, method, and number of bound services for your virtual servers. If you have configured more than the basic

load balancing settings, you can view the persistence settings for your virtual servers, any policies that are bound to them, and any cache redirection and content switching virtual servers that have been bound to the virtual servers.

To view the properties of a load balancing virtual server by using the CLI

At the command prompt, type:

```
show lb vserver <name>

show lb vserver Vserver-LB-1
<!--NeedCopy-->
```

To view the properties of a load balancing virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the details pane, click a virtual server to display its properties at the bottom of the details pane.
3. To view the cache redirection and content switching virtual servers that are bound to this virtual server, click **Show CS/CR Bindings**.

Viewing the Properties of a Service

You can view the name, state, IP address, port, protocol, maximum client connection, maximum requests per connection, and server type of the configured services, and use this information to troubleshoot any mistake in the service configuration.

To view the properties of services by using the CLI

At the command prompt, type:

```
show service <name>

show service Service-HTTP-1
<!--NeedCopy-->
```

To view the properties of services by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Services**. The details of the available services appear on the Services pane.

Viewing the Bindings of a Service

You can view the list of virtual servers to which the service is bound. The binding information also provides the name, IP address, port, and state of the virtual servers to which the services are bound. You can use the binding information to troubleshoot any problem with binding the services to virtual servers.

To view the bindings of a service by using the CLI

At the command prompt, type:

```
show service bindings <name>

show service bindings Service-HTTP-1
<!--NeedCopy-->
```

To view the bindings of a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. In the details pane, select the service whose binding information you want to view.
3. In the **Action** tab, click **Show Bindings**.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the Citrix ADC appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the CLI

To display a summary of the statistics for all the virtual servers currently configured on the appliance, or for a single virtual server, at the command prompt, type:


```
stat lb vserver [`<name>`]
<!--NeedCopy-->
```

Example:

```
stat lb vserver server-1
<!--NeedCopy-->
```

The following figure displays a sample statistic.


```
> stat lbvserver
[
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
vserver1        10.102.20.200    80          SSL         DOWN      0/s

lb1               203.1.113.5   443         DTLS         DOWN      0/s

vicap                       *     0          TCP         DOWN      0/s

lbicap                2.2.3.4  1344          TCP         DOWN      0/s

app_...stest          0.0.0.0     0         HTTP         DOWN      0/s

app_...ttest          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...test1          0.0.0.0     0         HTTP         DOWN      0/s

app_...1test          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...est12          0.0.0.0     0         HTTP         DOWN      0/s

app_...sting          0.0.0.0     0         HTTP         DOWN      0/s

test                  2.2.2.2    80         HTTP         DOWN      0/s

shar...lt-lb          0.0.0.0     0         HTTP         DOWN      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ts-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ns-lb          0.0.0.0     0         HTTP           UP      0/s

shar...as-lb          0.0.0.0     0         HTTP           UP      0/s

forward-vs            0.0.0.0     0          TCP         DOWN      0/s

tcpcs                 0.0.0.0     0          TCP         DOWN      0/s

test124               0.0.0.0     0          SSL         DOWN      0/s

testssl               0.0.0.0     0          SSL         DOWN      0/s
```


To display virtual server statistics by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.
3. In the details pane, click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in the surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the CLI

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. In the details pane, select the service whose statistics you want to view (for example, Service-HTTP-1).
3. Click **Statistics**. The statistics appear in a new window.

Load balance virtual server and service states

September 14, 2021

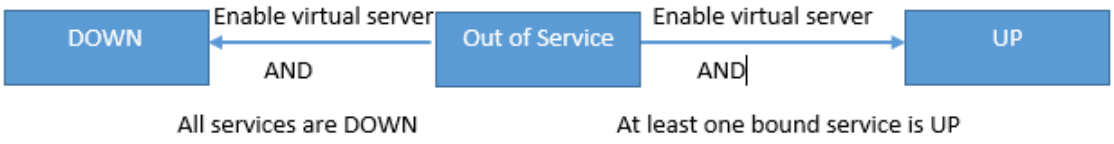
A load balancing virtual server that does not have a backup virtual server can take the following states, depending on the states of the services bound to it and whether it is administratively disabled:

- **UP**: At least one of the services bound to the virtual server is UP.

- **DOWN**: All the services bound to the virtual server are DOWN, or the load balancing feature is not enabled.
- **Out of Service (OFS)**: If you administratively disable the virtual server, it enters the OFS state but its effective state is DOWN. Administrator can control the transitioning to the OFS state from the DOWN or UP state, or to the DOWN or UP state from the OFS state.

The state and effective state of a virtual server are the same if a backup virtual server is not configured. However, if a backup virtual server or a chain of backup virtual servers is configured, the effective state is derived from the states of the services that are bound to the primary virtual server and the backup virtual servers. If any of the backup virtual servers in the chain is UP, the effective state of the primary virtual server is UP, even if all the services bound to the primary virtual server are DOWN.

The following diagrams show the conditions under which a virtual server transitions from one state to another.



A service can take the following states:

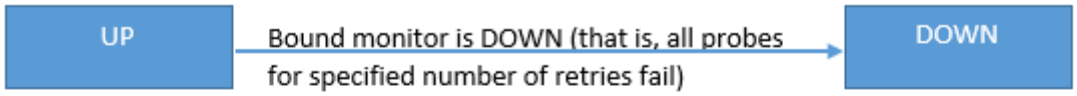
- **UP:** If probes from all the monitors bound to the service are successful.
- **DOWN**: If monitoring probes to the service are not answered within the configured time limit.
- **OUT OF SERVICE**: If you administratively disable the service, or if you gracefully shut down the service and there are no active transactions to the service

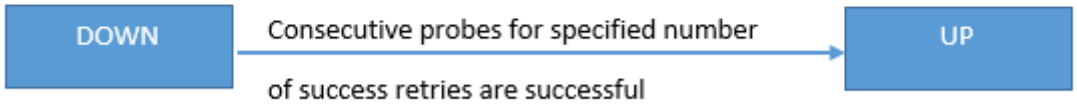
- **GOING OUT OF SERVICE (TROFS)**: If you administratively disable the service with delay, or gracefully shut down the service and there are active transactions to the service. For more information, see Graceful Shut down of Services.
- **DOWN WHEN GOING OUT OF SERVICE (TROFS_DOWN)[]** A monitoring probe fails while the service is in the GOING OUT OF SERVICE state.

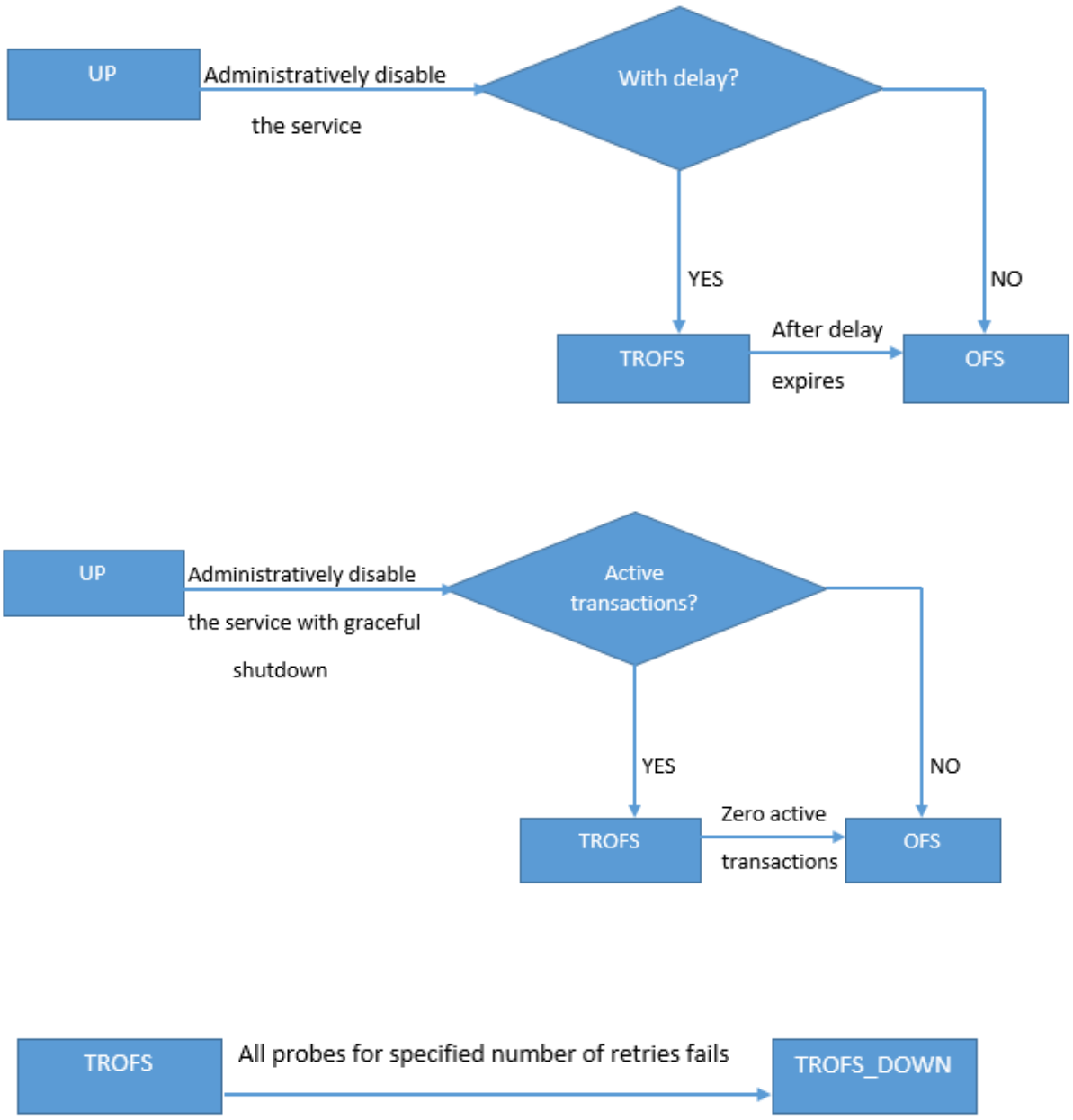
A service in the process of transitioning from UP to OFS is in the GOING OUT OF SERVICE state. A service transitioning from DOWN to OFS is in the DOWN WHEN GOING OUT OF SERVICE state. For example, if a service is DOWN and you disable it with delay, the service transitions to DOWN WHEN GOING OUT OF SERVICE and then to the OUT OF SERVICE state. If a service is UP and you disable it with delay, the service transitions to GOING OUT OF SERVICE. During this time, if a monitoring probe to the server fails, the service transitions to DOWN WHEN GOING OUT OF SERVICE and, after the delay time expires, enters the OFS state.

> **Note**
>
> You can configure spillover to a backup virtual server by setting the “healthThreshold” parameter to a non-zero positive value. Then, if a single service bound to the primary virtual server transitions to the DOWN WHEN GOING OUT OF SERVICE state and the health threshold is not reached, the primary virtual server is marked DOWN and new connections are directed to the backup virtual server.

The following diagrams show the conditions under which a service transitions from one state to another.







Support for load balancing profile

September 14, 2021

A load balancing configuration has many parameters, so setting the same parameters on several virtual servers can become tedious. From release 11.1, a load balancing (LB) profile makes this task

easier. You can now set load balancing parameters in a profile and associate this profile with virtual servers, instead of setting these parameters on each virtual server.

The following parameters are presently supported in an LB profile:

- `HTTPonlyflag`—Include the HttpOnly attribute in persistence cookies. The HttpOnly attribute limits the scope of a cookie to HTTP requests and helps mitigate the risk of cross-site scripting attacks.
- UseSecuredPersistenceCookie—Encrypt the persistence cookie values by using the SHA2 hash algorithm.
- `Cookiepassphrase`—Specify the passphrase used to generate a secured persistence cookie value.
- DBS_LB—Enable database specific load balancing for MySQL and MSSQL service types.
- Cl_process_local—Packets destined to a virtual server in a cluster are not steered. Enable the option for single packet request response mode or when the upstream device is performing a proper RSS for connection based distribution.
- lbHashAlgorithm—Specify the hashing algorithm to be used for the following hash-based load balancing methods:
  - URL hash method
  - Domain hash method
  - Destination IP hash method
  - Source IP hash method
  - Source IP Destination IP hash method
  - Source IP Source Port hash method
  - Call ID hash method
  - Token method

  Possible values: DEFAULT, PRAC, JARH
  Default value: DEFAULT
- lbHashFingers—Specify the number of fingers to be used in PRAC and JARH algorithms for hash-based LB methods. Increasing the number of fingers provides better distribution of traffic at the expense of additional memory.

Default value: 256
Minimum value: 1
Maximum value: 1024

> **Note**
>
> You can set DBS_LB and Cl_process_local parameters on a virtual server and in the profile. If you enable these parameters on a virtual server and then set a profile to this virtual server, the

> parameters appear as disabled in the output of the "show lb vserver" command for that virtual server. Check the profile to see the actual status of these parameters. In addition, if you set and then unset a profile to a virtual server, the parameters are set with default values for that virtual server.

To create an LB profile by using the CLI

At the command prompt, type:

```
add lb profile <lbprofilename> -dbsLb ( ENABLED | DISABLED ) -
    processLocal ( ENABLED | DISABLED ) -httpOnlyCookieFlag ( ENABLED |
    DISABLED ) -cookiePassphrase -useSecuredPersistenceCookie ( ENABLED
    | DISABLED ) -lbHashAlgorithm <lbHashAlgorithm> -lbHashFingers <
    positive_integer>
<!--NeedCopy-->
```

Example:

```
> sh lb profile p1
LB Profile name:  p1
DBS LB : DISABLED   Process Local: DISABLED
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
No of vservers bound: 0
Store MQTT clientid and username in transactional logs: NO
Hash LB algorithm used in LB decision: DEFAULT
Number of fingers for Hash LB algorithm: 256
Done

<!--NeedCopy-->
```

To create an LB profile by using the GUI

Navigate to **System** > **Profiles** > **LB Profile**, and add a profile.

To associate an LB profile with an LB virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -lbprofilename <string>
<!--NeedCopy-->
```

Example


```
set lbvserver lbvip1 -lbprofile p1

Done

sh lb vserver lbvip1

lbvip1 (203.0.113.1:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Wed May 25 12:36:20 2016
Time since last state change: 0 days, 00:01:26.140
Effective State: UP  ARP:DISABLED
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: ENABLED
Port Rewrite : DISABLED
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION      BackupMethod: ROUNDROBIN
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule: none
L2Conn: OFF
Skip Persistency: None
Listen Policy: NONE
IcmpResponse: PASSIVE
RHIstate: PASSIVE
New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
Mac mode Retain Vlan: DISABLED
DBS_LB: DISABLED
Process Local: DISABLED
Traffic Domain: 0
LB Profile: p1
Done
<!--NeedCopy-->
```

To associate an LB profile with an LB virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Select a virtual server, and click **Edit**.
3. In **Advanced Settings**, click **Profiles**.

4. In the **LB Profile** list, select the profile to associate with this virtual server.

Load balancing algorithms

September 14, 2021

The load balancing algorithm defines the criteria that the Citrix ADC appliance uses to select the service to which to redirect each client request. Different load balancing algorithms use different criteria. For example, the least connection algorithm selects the service with the fewest active connections, while the round robin algorithm maintains a running queue of active services, distributes each connection to the next service in the queue, and then sends that service to the end of the queue.

Some load balancing algorithms are best suited to handling traffic on websites, others to managing traffic to DNS servers, and others to handling complex web applications used in e-commerce or on company LANs or WANs. The following table lists each load balancing algorithm that the Citrix ADC appliance supports, with a brief description of how each operates.

Name	Server selection based on
LEASTCONNECTION	Which service currently has the fewest client connections. This is the default load balancing algorithm.
ROUNDROBIN	Which service is at the top of a list of services. After that service is selected for a connection, it moves to the bottom of the list.
LEASTRESPONSETIME	Which load balanced server currently has the quickest response time.
URLHASH	A hash of the destination URL.
DOMAINHASH	A hash of the destination domain.
DESTINATIONIPHASH	A hash of the destination IP address.
SOURCEIPHASH	A hash of the source IP address.
SRCIPDESTIPHASH	A hash of the source and destination IP addresses.
CALLIDHASH	A hash of the call ID in the SIP header.
SRCIPSRCPORTHASH	A hash of the client's IP address and port.
LEASTBANDWIDTH	Which service currently has the fewest bandwidth constraints.

Name	Server selection based on
LEASTPACKETS	Which service currently is receiving the fewest packets.
CUSTOMLOAD	Data from a load monitor.
TOKEN	The configured token.
LRTM	Fewest active connections and the lowest average response time.

Depending on the protocol of the service that it is load balancing, the Citrix ADC appliance sets up each connection between client and server to last for a different time interval. This is called load balancing granularity, of which are three types: request-based, connection-based, and time-based granularity. The following table describes each type of granularity and when each is used.

Granularity	Types of Load Balanced Service	Specifies
Request -based	HTTP or HTTPS	A new service is chosen for each HTTP request, independent of TCP connections. As with all HTTP requests, after the Web server fulfills the request, the connection is closed.
Connection-based	TCP and TCP-based protocols other than HTTP	A service is chosen for every new TCP connection. The connection persists until terminated by either the service or the client.

Granularity	Types of Load Balanced Service	Specifies
Time-based	UDP and other IP protocols	A new service is chosen for each UDP packet. Upon selection of a service, a session is created between the service and a client for a specified period. When the time expires, the session is deleted and a new service is chosen for any additional packets, even if those packets come from the same client.

During startup of a virtual server, or whenever the state of a virtual server changes, the virtual server can initially use the round robin method to distribute the client requests among the physical servers. This type of distribution, referred to as *startup round robin,* helps prevent unnecessary load on a single server as the initial requests are served. After using the round robin method at the startup, the virtual server switches to the load balancing method specified on the virtual server.

The Startup RR Factor works in the following manner:

- If the Startup RR Factor is set to zero, the appliance switches to the specified load balancing method depending on the request rate.
- If the Startup RR Factor is any number other than zero, the appliance uses the round robin method for the specified number of requests before switching to the specified load balancing method.
- By default, the Startup RR Factor is set to zero.

Note: You cannot set the startup RR Factor for an individual virtual server. The value you specify applies to all the virtual servers on the Citrix ADC appliance.

To set the startup round-robin factor by using the CLI

At the command prompt, type:

```
set lb parameter -startupRRFactor <positive_integer>
```

Example

```
set lb parameter -startupRRFactor 25000
```


To set the startup round-robin factor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Configure Load Balancing Parameters**, and set the Startup RR Factor.

Least connection method

September 14, 2021

When a virtual server is configured to use the least connection load balancing algorithm (or method), it selects the service with the fewest active connections. This is the default method, because, in most circumstances, it provides the best performance.

For TCP, HTTP, HTTPS, and SSL_TCP services, the Citrix ADC appliance includes the following connection types in its list of existing connections:

- **Active connections to a service.** Connections representing requests that a client has sent to the virtual server and that the virtual server has forwarded to a service. For HTTP and HTTPS services, active connections represent only those HTTP or HTTPS requests that have not yet received a response.
- **Waiting connections in the surge queue.** Any connections to the virtual server that are waiting in a surge queue and have not yet been forwarded to a service. Connections can build up in the surge queue at any time, for any of the following reasons:
  - Your services have connection limits, and all services in your load balancing configuration are at that limit.
  - The surge protection feature is configured and has been activated by a surge in requests to the virtual server.
  - The load-balanced server has reached an internal limit and therefore does not open any new connections. (For example, an Apache server's connection limit is reached.)

When a virtual server uses the least connection method, it considers the waiting connections as belonging to the specific service. Therefore, it does not open new connections to those services.

For UDP services, the connections that the least connection algorithm considers include all sessions between the client and a service. These sessions are logical, time-based entities. When the first UDP packet in a session arrives, the Citrix ADC appliance creates a session between the source IP address and port and the destination IP address and port.

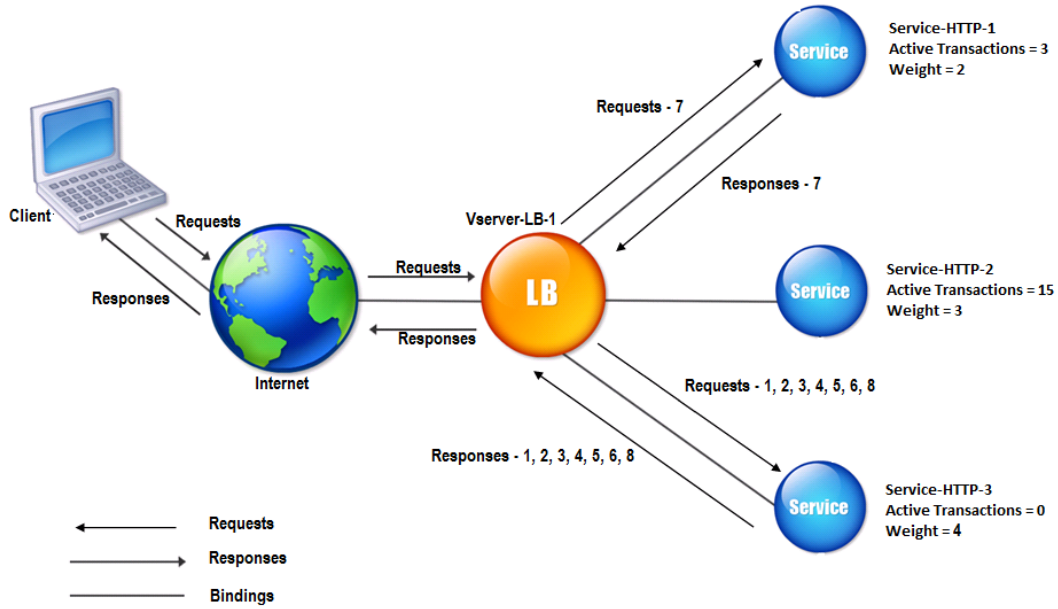
For Real-Time Streaming Protocol (RTSP) connections, the Citrix ADC appliance uses the number of active control connections to determine the lowest number of connections to an RTSP service.

The following example shows how a virtual server selects a service for load balancing by using the least connection method. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions.
- Service-HTTP-2 is handling 15 active transactions.
- Service-HTTP-3 is not handling any active transactions.

The following diagram illustrates how the Citrix ADC appliance forwards incoming requests when using the least connection method.

Figure 1. Mechanism of the Least Connections Load Balancing Method



In this diagram, the virtual server selects the service for each incoming connection by choosing the server with the fewest active transactions.

Connections are forwarded as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  Note: The service with no active transaction is selected first.

- Service-HTTP-3 receives the second and third requests because the service has the next least number of active transactions.
- Service-HTTP-1 receives the fourth request because Service-HTTP-1 and Service-HTTP-3 have the same number of active transactions, the virtual server uses the round robin method to choose between them.
- Service-HTTP-3 receives the fifth request.

- Service-HTTP-1 receives the sixth request, and so on, until both Service-HTTP-1 and Service-HTTP-3 are handling the same number of requests as Service-HTTP-2. Then, the Citrix ADC appliance starts forwarding requests to Service-HTTP-2 when it is the least loaded service or its turn comes up in the round robin queue.

  Note: If connections to Service-HTTP-2 close, it might get new connections before each of the other two services has 15 active transactions.

The following table explains how connections are distributed in the three-service load balancing setup described earlier.

Incoming Connection	Service Selected	Current Number of Active Connections	Remarks
Request-1	Service-HTTP-3; (N = 0)	1	Service-HTTP-3 has the fewest active connections.
Request-2	Service-HTTP-3; (N = 1)	2	Service-HTTP-3 has the fewest active connections.
Request-3	Service-HTTP-3; (N = 2)	3	-
Request-4	Service-HTTP-1; (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.
Request-5	Service-HTTP-3; (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.
Request-6	Service-HTTP-1;(N = 4)	5	-
Request-7	Service-HTTP-3; (N = 4)	5	-
Request-8	Service-HTTP-1; (N = 5)	6	-

Service-HTTP-2 is selected for load balancing when it completes its active transactions and the current connections to it close, or when the other services (Service-HTTP-1 and Service-HTTP-3) have 15 or more connections each.

The Citrix ADC appliance can also use the least connection method when weights are assigned to services. It selects a service by using the value (Nw) of the following expression:

Nw = (Number of active transactions) * (10000 / weight)

The following example shows how the Citrix ADC appliance selects a service for load balancing by using the least connection method when weights are assigned to services. In the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. Connections are forwarded as follows:

- Service-HTTP-3 receives the first because the service is not handling any active transactions.

  Note: If the services are not handling any active transactions, the Citrix ADC appliance uses the round robin method regardless of the weights assigned to each of the services.
- Service-HTTP-3 receives the second, third, fourth, fifth, sixth, and seventh requests because the service has the lowest Nw value.
- Service-HTTP-1 receives the eighth request. Because Service-HTTP-1 and Service-HTTP-3 now have the same Nw value, the appliance performs load balancing in a round robin manner. Therefore, Service-HTTP-3 receives the ninth request.

The following table explains how connections are distributed on the three-service load balancing setup that is described earlier.

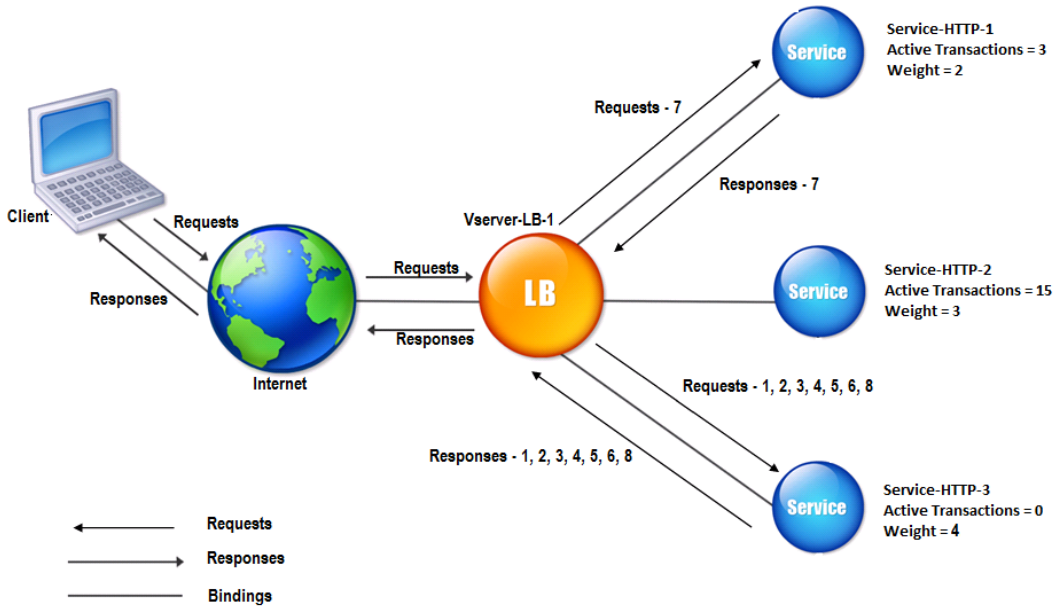
Request Received	Service Selected	Current Nw (Number of active transactions) * (10000 / weight) value	Remarks
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 2500	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 2500)	Nw = 5000	
Request-3	Service-HTTP-3; (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3; (Nw = 7500)	Nw = 10000	
Request-5	Service-HTTP-3; (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3; (Nw = 12500)	Nw = 15000	

Request Received	Service Selected	Current Nw (Number of active transactions) * (10000 / weight) value	Remarks
Request-7	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw values
Request-8	Service-HTTP-3; (Nw = 15000)	Nw = 17500	

Service-HTTP-2 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-1 and Service-HTTP-3) is equal to 50000.

The following diagram illustrates how the Citrix ADC appliance uses the least connection method when weights are assigned to the services.

Figure 2. Mechanism of the Least Connections Load Balancing Method when Weights are Assigned



To configure the least connection method, see Configuring a Load Balancing Method that Does Not Include a Policy.

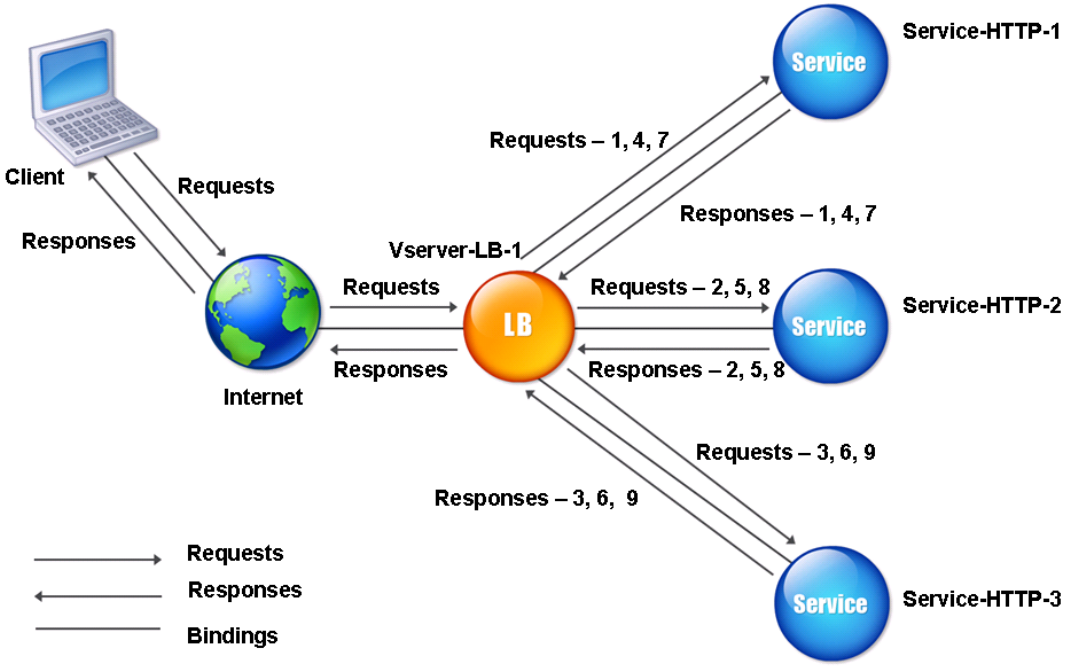
Round robin method

September 14, 2021

When a load balancing virtual server is configured to use the round robin method, it continuously rotates a list of the services that are bound to it. When the virtual server receives a request, it assigns the connection to the first service in the list, and then moves that service to the bottom of the list.

The following diagram illustrates how the Citrix ADC appliance uses the round robin method with a load balancing setup that contains three load-balanced servers and their associated services.

Figure 1. How the Round Robin Load Balancing Method Works



If you assign a different weight to each service, the Citrix ADC appliance performs the weighted round robin distribution of incoming connections. It does this by skipping the lower-weighted services at appropriate intervals.

For example, assume that you have a load balancing setup with three services. You set Service-HTTP-1 to a weight of 2, Service-HTTP-2 to a weight of 3, and Service-HTTP-3 to a weight of 4. The services are bound to Vserver-LB-1, which is configured to use the round robin method. With this setup, incoming requests are delivered as follows:

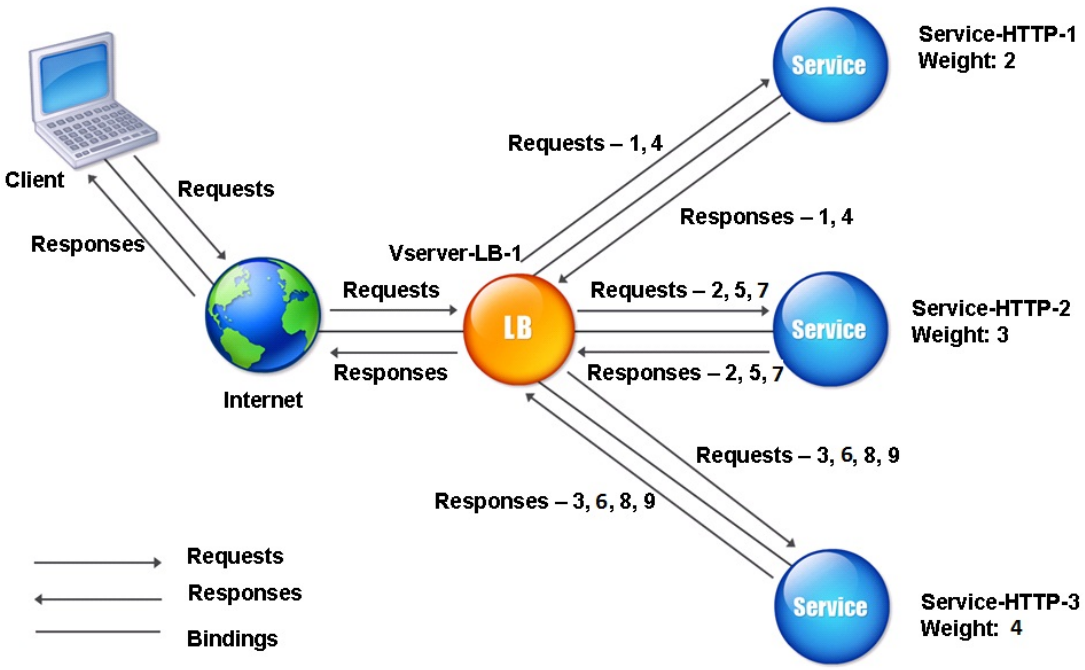
- Service-HTTP-1 receives the first request.
- Service-HTTP-2 receives the second request.
- Service-HTTP-3 receives the third request.
- Service-HTTP-1 receives the fourth request.
- Service-HTTP-2 receives the fifth request.
- Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh request.
- Service-HTTP-3 receives both the eighth and the ninth requests.

Note: You can also configure weights on services to prevent multiple services from using the same server and overloading the server.

A new cycle then begins, using the same pattern.

The following diagram illustrates the weighted round robin method.

Figure 2. How the Round Robin Load Balancing Method supports Weighted Services



To configure the round robin method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Least response time method

September 14, 2021

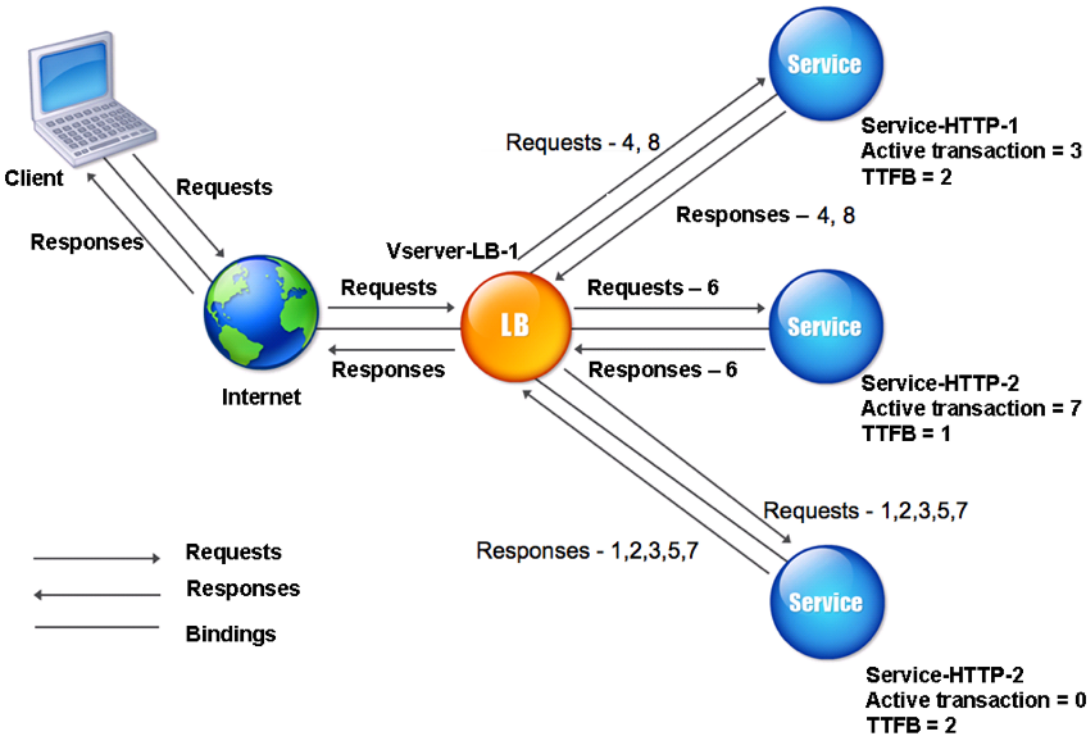
When the load balancing virtual server is configured to use the least response time method, it selects the service with the fewest active connections and the lowest average response time. You can configure this method for HTTP and Secure Sockets Layer (SSL) load balancing virtual servers only. The response time (also called Time to First Byte, or TTFB) is the time interval between sending a request packet to a service and receiving the first response packet from the service. The Citrix ADC appliance uses response code 200 to calculate TTFB.

The following example shows how a virtual server selects a service for load balancing by using the least response time method. Consider the following three services:

- Service-HTTP-1 is handling three active transactions and TTFB is two seconds.
- Service-HTTP-2 is handling seven active transactions and TTFB is one second.
- Service-HTTP-3 is not handling any active transactions and TTFB is two seconds.

The following diagram illustrates how the Citrix ADC appliance uses the least response time method to forward the connections.

Figure 1. How the Least Response Time Load Balancing Method Works



The virtual server selects a service by multiplying the number of active transactions by the TTFB for each service and then selecting the service with the lowest result. For the example shown above, the virtual server forwards requests as follows:

- Service-HTTP-3 receives the first request, because the service is not handling any active transactions.
- Service-HTTP-3 also receives the second and third requests, because the result is the lowest of the three services.
- Service-HTTP-1 receives the fourth request. Since Service-HTTP-1 and Service-HTTP-3 have the same result, the Citrix ADC appliance chooses between them by applying the Round Robin method.
- Service-HTTP-3 receives the fifth request.
- Service-HTTP-2 receives the sixth request, because at this point it has the lowest result.
- Because Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all have the same result at this point, the appliance switches to the round robin method, and continues to distribute connections using that method.

The following table explains how connections are distributed in the three-service load balancing setup described earlier.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 has the lowest N value.
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 has the lowest N value.
Request-4	Service-HTTP-1; (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values. The appliance uses the round robin method to distribute the requests.
Request-5	Service-HTTP-3; (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.

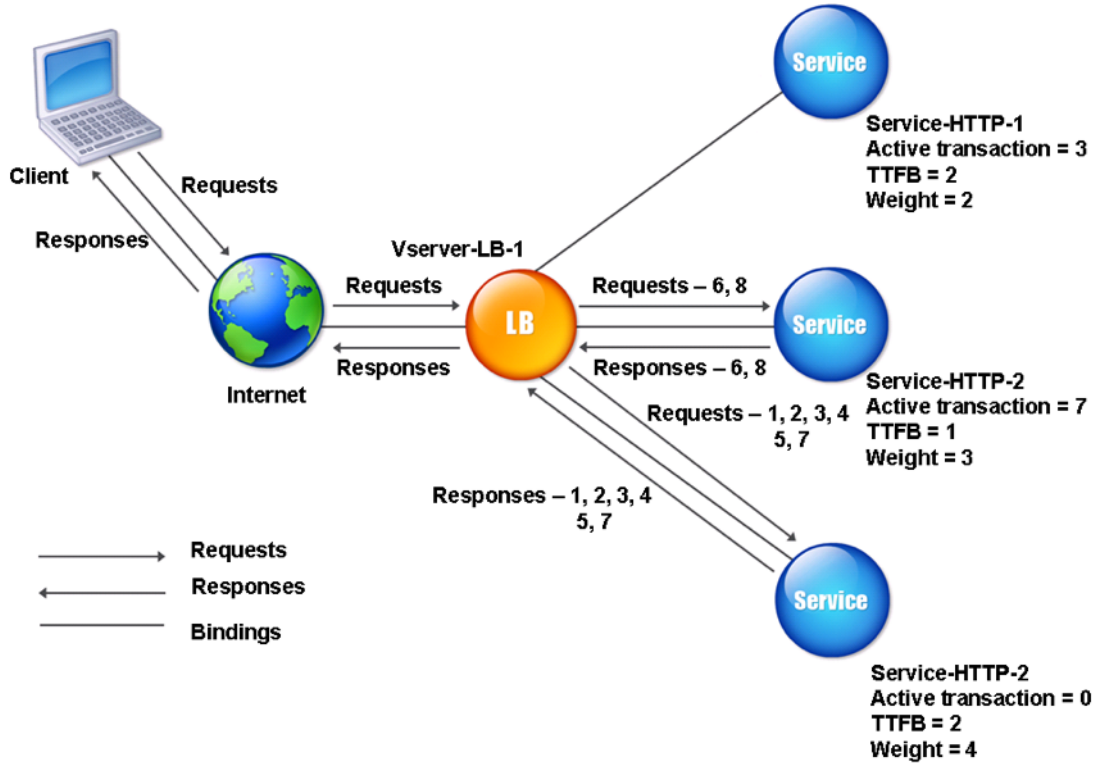
Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-6	Service-HTTP-2; (N = 7)	N = 8	Service-HTTP-2 has the lowest N value.
Request-7	Service-HTTP-3; (N = 8)	N = 10	Service-HTTP-1,Service-HTTP-2 and Service-HTTP-3 have the same N values. The Citrix ADC appliance uses the round robin method to distribute the requests.
Request-8	Service-HTTP-1; (N = 8)	N = 10	Service-HTTP-1 and Service-HTTP-2 have the same N values, the appliance uses the round robin method to distribute the requests.

Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when its N value is less than the other services (Service-HTTP-2 and Service-HTTP-3).

Selection of services when weights are assigned

The following diagram illustrates how the Citrix ADC appliance uses the least response time method when weights are assigned.

Figure 2. How the Least Response Time Load Balancing Method Works When Weights Are Assigned



The virtual server selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight), where N = (number of active transactions * TTFB)

Suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned weight of 3, and Service-HTTP-3 is assigned weight of 4.

The Citrix ADC appliance distributes requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  If the services are not handling any active transactions, the appliance selects them regardless of the weights assigned to them.
- Service-HTTP-3 receives the second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and therefore the highest Nw value, so the virtual server does not select it for load balancing.

The following table explains how connections are distributed in the three-service load balancing setup described earlier.

Request Received	Service Selected	Current Nw Value = (N) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 has the lowest Nw value.
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 has the lowest Nw value.
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 has the lowest Nw value.
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 has the lowest Nw value.
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 has the lowest Nw value.
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 is selected for load balancing when it completes its active transactions or when its Nw value is less than other services (Service-HTTP-2 and Service-HTTP-3).

To configure the least response time load balancing method by using the CLI

At the command prompt type;

```
set lb vserver <name> -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```


To configure the least response time load balancing method by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **LEASTRESPONSETIME**.

For more information about configuring monitors, see Configuring Monitors in a Load Balancing Setup.

LRTM method

September 14, 2021

Note: LRTM stands for Least response time method using monitors (LRTM).

When a load balancing virtual server is configured to use the LRTM method, it uses the existing monitoring infrastructure to get the fastest response time. The load balancing virtual server then selects the service with the smallest number of active transactions and lowest response time. Before you use the LRTM method, you must bind application-specific monitors to each service and enable LRTM mode on these monitors. The Citrix ADC appliance then makes load balancing decisions based on the response times it calculates from monitoring probes.

You can use the LRTM method to load balance non-HTTP and non-HTTPS services also. You can also use this method when several monitors are bound to a service. Each monitor determines the response time by using the protocol that it measures for the service that it is bound to. The virtual server then calculates an average response time for that service by averaging the results.

The following table summarizes how response times are calculated for the various monitors.

Monitor	Response time calculation
PING	Time difference between the ICMP ECHO request and the ICMP ECHO response.
TCP	Time difference between the SYN request and the SYN+ACK response.
HTTP	Time difference between the HTTP request (after the TCP connection is established) and the HTTP response.
TCP-ECV	Time difference between the time the data send string is sent and the data receive string is returned. A TCP-ECV monitor without the send and receive strings is considered to have an incorrect configuration.

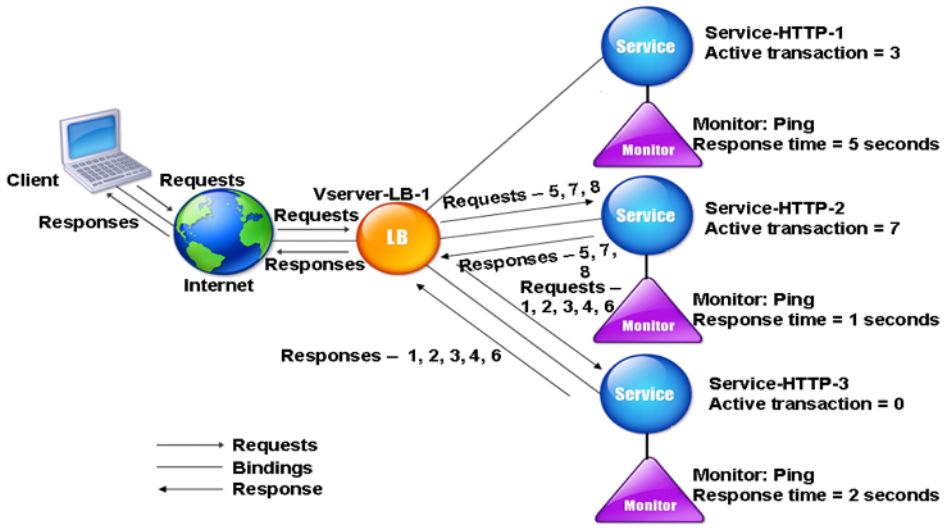
Monitor	Response time calculation
HTTP-ECV	Time difference between the HTTP request and the HTTP response.
UDP-ECV	Time difference between the UDP's send string and the receive string. A UDP-ECV monitor without the receive string is considered to have an incorrect configuration.
DNS	Time difference between a DNS query and the DNS response.
TCPS	Time difference between a SYN request and the SSL handshake completion.
FTP	Time difference between the sending of the user name and the completion of user authentication.
HTTPS (monitors HTTPS requests)	Time difference is the same as for the HTTP monitor.
HTTPS-ECV (monitors HTTPS requests)	Time difference is the same as for the HTTP-ECV monitor
USER	Time difference between the time when a request is sent to the dispatcher and the time when the dispatcher response is received.

The following example shows how the Citrix ADC appliance selects a service for load balancing by using the LRTM method. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions and the response time is five seconds.
- Service-HTTP-2 is handling 7 active transactions and the response time is one second.
- Service-HTTP-3 is not handling any active transactions and the response time is two seconds.

The following diagram illustrates the process that the Citrix ADC appliance follows when it forwards requests.

Figure 1. How the LRTM Method Works



The virtual server selects a service by using the value (N) in the following expression:

N = (Number of active transactions * Response time that is determined by the monitor)

The virtual server delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service is not handling any active transaction.
- Service-HTTP-3 receives the second, third, and fourth requests, because this service has the lowest N value.
- Service-HTTP-2 receives the fifth request, because this service has the lowest N value.
- Since both Service-HTTP-2 and Service-HTTP-3 currently have the same N value, the Citrix ADC appliance switches to the round robin method. Therefore, Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh and eighth requests, because this service has the lowest N value.

Service-HTTP-1 is not considered for load balancing, because it is more heavily loaded (has the highest N value) when compared to the other two services. However, if Service-HTTP-1 completes its active transactions, the Citrix ADC appliance again considers that service for load balancing.

The following table summarizes how N is calculated for the services.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 has the lowest N value.
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 has the lowest N value.
Request-4	Service-HTTP-3; (N = 6)	N = 8	Service-HTTP-3 has the lowest N value.
Request-5	Service-HTTP-2; (N = 7)	N = 8	Service-HTTP-2 has the lowest N value.
Request-6	Service-HTTP-3; (N = 8)	N = 10	Service-HTTP-2 and Service-HTTP-3 have the same N values. Citrix ADC appliance switches to the round robin method and selects Service-HTTP-3
Request-7	Service-HTTP-2; (N = 8)	N = 9	Service-HTTP-2 has the lowest N value.
Request-8	Service-HTTP-2; (N = 9)	N = 10	Service-HTTP-2 has the lowest N value.

Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when its N value is less than the other services (Service-HTTP-2 and Service-HTTP-3).

Selection of services when weights are assigned

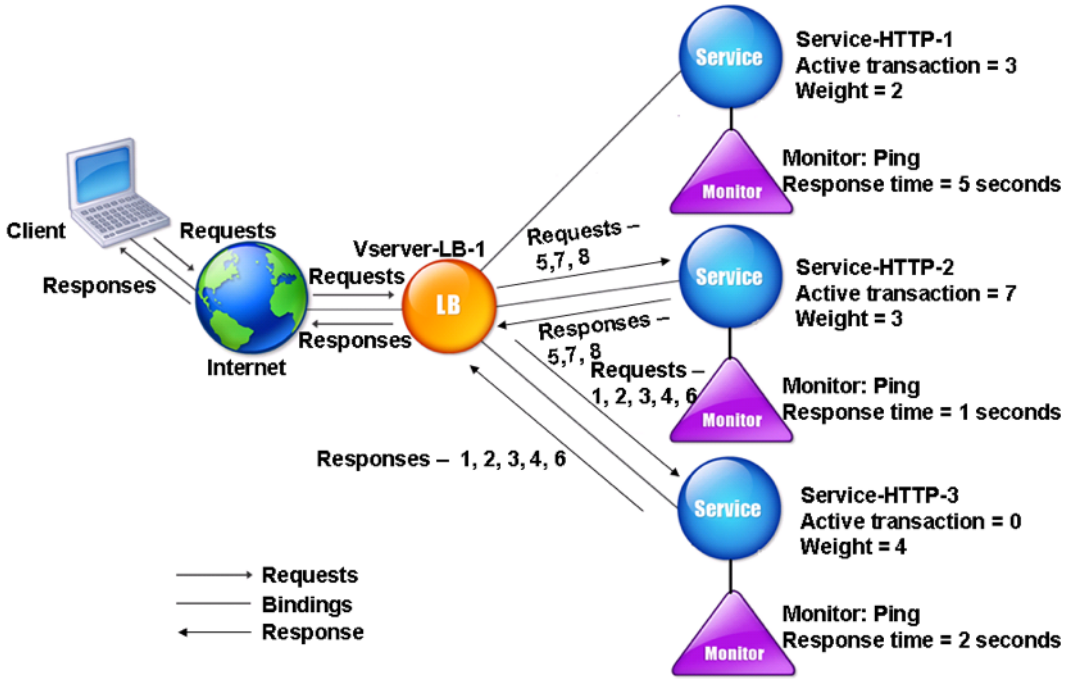
The Citrix ADC appliance also performs load balancing by using the number of active transactions, response time, and weights if different weights are assigned to services. The Citrix ADC appliance selects the service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

Where N = (Number of active transactions * Response time that is determined by the monitor)

The following diagram illustrates how the virtual server uses the LRTM method when weights are assigned.

Figure 2. How the Least Response Time Load Balancing Method Works When Weights Are Assigned



In this example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4.

The Citrix ADC appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.
- Service-HTTP-3 receives the second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth requests, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and the highest Nw value, so the Citrix ADC appliance does not select it for load balancing.

The following table summarizes how Nw is calculated for various monitors.

Request Received	Service Selected	Current Nw Value (N) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 has the lowest Nw value.
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 has the lowest Nw value.
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 has the lowest Nw value.
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 has the lowest Nw value.
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 has the lowest Nw value.
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 is selected for load balancing when it completes its active transactions or when its Nw value is less than other services (Service-HTTP-2 and Service-HTTP-3).

To configure the LRTM load balancing method by using the CLI

At the command prompt type;

```
set lb vserver <name> [-lbMethod <lbMethod>]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod LRTM
<!--NeedCopy-->
```

To configure the LRTM load balancing method by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **LRTM**.

To enable the LRTM option in monitors by using the CLI

At the command prompt type;

```
set lb monitor <monitorName> <type> [-LRTM ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

Example:

```
set lb monitor monitor-HTTP-1 HTTP -LRTM ENABLED
<!--NeedCopy-->
```

To enable the LRTM option in monitors by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**, and open a monitor.
2. In Advanced Parameters, select **LRTM (Least Response Time using Monitoring)**.

For more information about configuring monitors, see Configuring Monitors in a Load Balancing Setup.

Hashing methods

September 14, 2021

Load balancing methods based on hashes of certain connection information or header information constitute most the Citrix ADC appliance's load balancing methods. Hashes are shorter and easier to use than the information that they are based on, while retaining enough information to ensure that no two different pieces of information generate the same hash and are therefore confused with one another.

You can use the hashing load balancing methods in an environment where a cache serves a wide range of content from the Internet or specified origin servers. Caching requests reduces request and response latency, and ensures better resource (CPU) utilization, making caching popular on heavily used websites and application servers. Since these sites also benefit from load balancing, hashing load balancing methods are widely useful.

The Citrix ADC appliance provides the following hashing methods:

- URL hash method
- Domain hash method
- Destination IP hash method
- Source IP hash method
- Source IP Destination IP hash method

- Source IP Source Port hash method
- Call ID hash method
- Token method

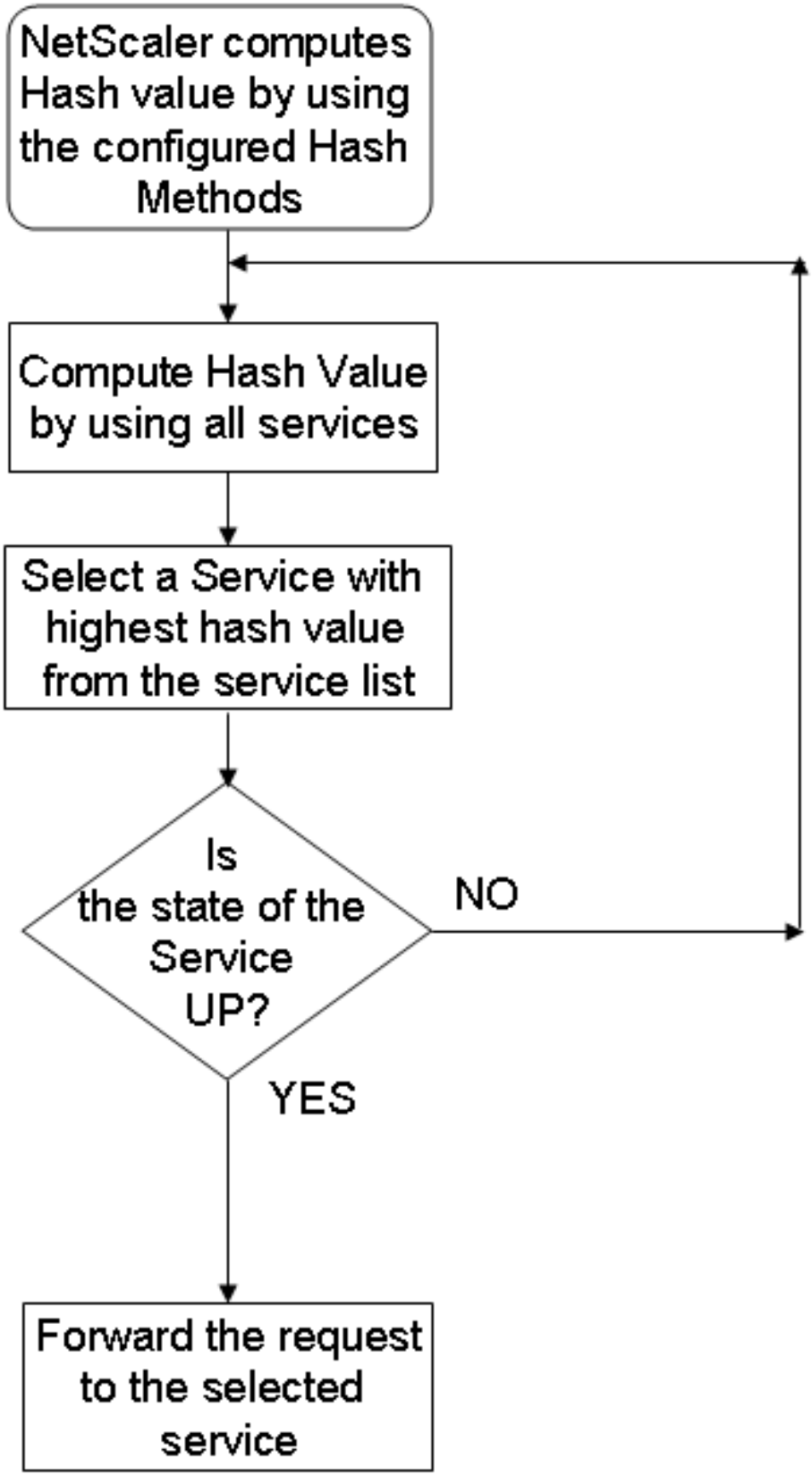
Most of the hashing algorithms calculate two hash values:

- A hash of the service's IP address and port.
- A hash of the incoming URL, the domain name, the source IP address, the destination IP address, or the source and destination IP addresses, depending on the configured hash method.

The Citrix ADC appliance then generates a new hash value by using both of those hash values. Finally, it forwards the request to the service with the highest hash value. As the appliance computes a hash value for each request and selects the service that processes the request, it populates a cache. Subsequent requests with the same hash value are sent to the same service. The following flow chart illustrates this process.

> Note
>
> Starting from Citrix ADC release 13.0 build 79.x, Prime Re-Shuffled Assisted CARP (PRAC) and Jump table Assisted Ring Hash (JARH) consistent hashing algorithms are supported. The consistent hashing algorithms ensure minimal disruption when services are added to or deleted from your load balancing setup, or during a service flap event in load balancing setup. For more details, see Consistent hashing algorithms.

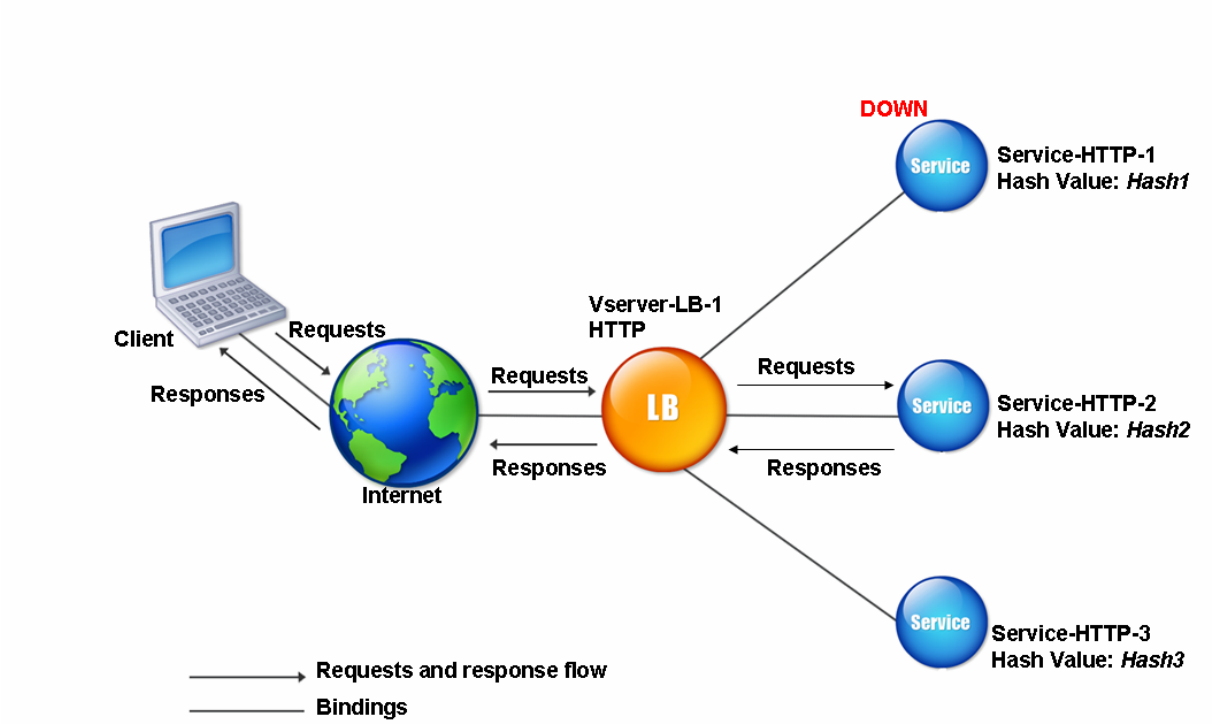
Figure 1. How the Hashing Methods Distribute Requests



Hashing methods can be applied to IPv4 and IPv6 addresses.

Consider a scenario where three services (Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3) are bound to a virtual server, any hash method is configured, and the hash value is Hash1. When the configured services are UP, the request is sent to Service-HTTP-1. If Service-HTTP-1 is down, the Citrix ADC appliance calculates the hash value for the last log of the number of services. The appliance then selects the service with the highest hash value, such as Service-HTTP-2. The following diagram illustrates this process.

Figure 2. Entity Model for Hashing Methods



> Note
>
> If the Citrix ADC appliance fails to select a service by using a hashing method, it defaults to the least connection method to select a service for the incoming request. Adjust server pools by removing services during periods of low traffic to enable the caches to repopulate without affecting performance on your load balancing setup.

Consistent hashing algorithms

The consistent hashing algorithms are used to achieve stateless persistency. The hash-based LB methods use one of the following three consistent hashing algorithms:

- **Cache Array Routing Protocol (CARP)**

  The CARP algorithm is used in load-balancing HTTP requests across multiple proxy cache

servers. This algorithm is enabled by default.

- **Prime Re-Shuffled Assisted CARP (PRAC)**

  The Citrix ADC appliance uses the proprietary PRAC algorithm to provide uniform traffic distribution.

- **Jump table Assisted Ring Hash (JARH)**

  The Citrix ADC appliance uses the proprietary JARH algorithm to provide consistency and uniform distribution of the traffic. This algorithm uses hash fingers. Higher number of fingers provide better distribution of traffic. However, increasing the number of fingers increases the memory usage as well.

To choose the consistent hashing algorithm by using CLI

```
set lb parameter [-lbHashAlgorithm [DEFAULT|JARH|PRAC] [-lbHashFingers
    <positive_integer>]
<!--NeedCopy-->
```

Example:

```
set lb parameter -lbHashAlgorithm JARH -lbHashFingers 10
<!--NeedCopy-->
```

ARGUMENTS:

- **lbHashAlgorithm**-Specify the hashing algorithm to be used for the following hash-based load balancing methods:
  - URL hash method
  - Domain hash method
  - Destination IP hash method
  - Source IP hash method
  - Source IP Destination IP hash method
  - Source IP Source Port hash method
  - Call ID hash method
  - Token method

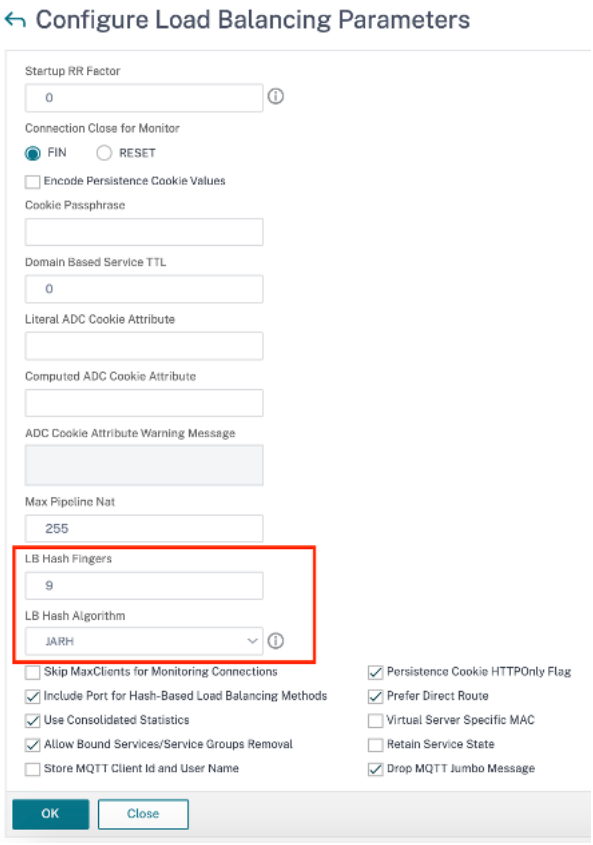
  Possible values: DEFAULT, PRAC, JARH
  Default value: DEFAULT

- **lbHashFingers**-Specify the number of fingers to be used in PRAC and JARH algorithms for hash-based LB methods. Increasing the number of fingers provides better distribution of traffic at the expense of extra memory.

Default value: 256
Minimum value: 1
Maximum value: 1024

To choose the consistent hashing algorithm by using GUI

1. Navigate to **Traffic Management > Load Balancing > Change Load Balancing parameters**.
2. In the **Configure Load Balancing Parameters** pane, enter appropriate values for the following fields based on your requirement:
   - LB Hash Fingers
   - In the **LB Hash Algorithm** field, choose the consistent hashing algorithm from the drop-down menu.



The URL Hash Method

When you configure the Citrix ADC appliance to use the URL hash method for load balancing the services, for selecting a service, the appliance generates a hash value of the HTTP URL present in the incoming request. If the service selected by the hash value is DOWN, the algorithm has a method to select another service from the list of active services. The appliance caches the hashed value of the

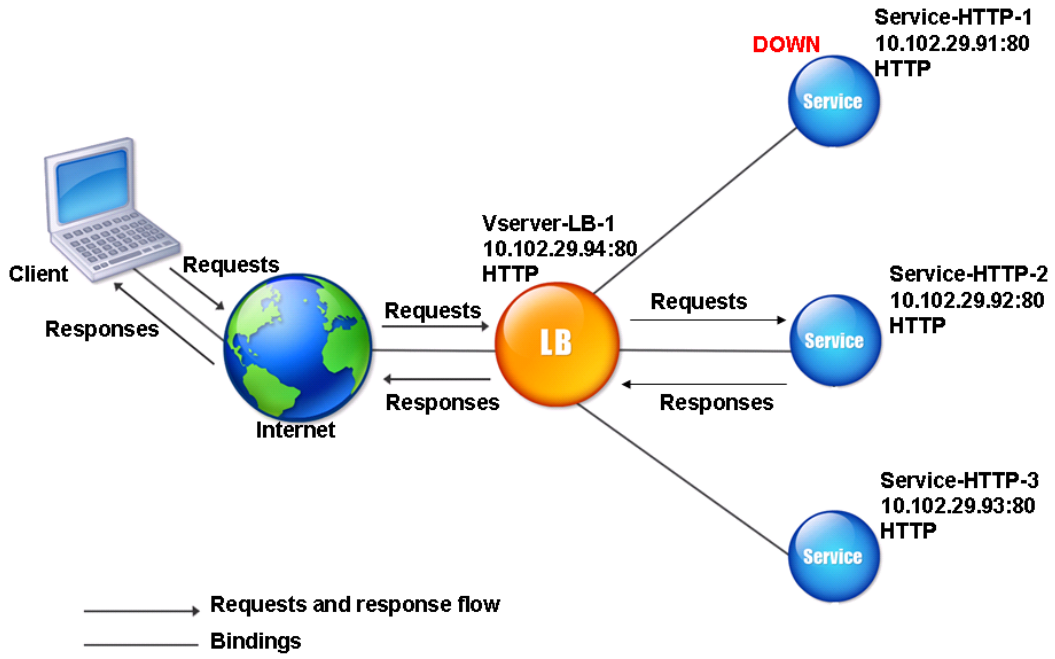
URL, and when it receives subsequent requests that use the same URL, it forwards them to the same service. If the appliance cannot parse an incoming request, it uses the round robin method for load balancing instead of the URL hash method.

For generating the hash value, the appliance uses a specific algorithm and considers a part of the URL. By default, the appliance considers the first 80 bytes of the URL. If the URL is of less than 80 bytes, the complete URL is used. You can specify a different length. The hash length can be from 1 byte to 4096 bytes. Generally, if long URLs are used where only a few characters are different, it is a good idea to make the hash length as high as possible to try to ensure a more even load distribution.

Consider a scenario where three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3, are bound to a virtual server, and the load balancing method configured on the virtual server is the URL hash method. The virtual server receives a request and the hash value of the URL is U1. Appliance selects Service-HTTP-1. If Service-HTTP-1 is DOWN, the appliance selects Service-HTTP-2.

The following diagram illustrates this process.

Figure 3. How URL Hashing Operates



If both Service-HTTP-1 and Service-HTTP-2 are DOWN, the appliance sends requests with hash value U1 to Service-HTTP-3.

If Service-HTTP-1 and Service-HTTP-2 are down, requests that generate the hash URL1 are sent to

Service-HTTP-3. If these services are UP, the requests that generate the hash URL1 are distributed in the following manner:

- If the Service-HTTP-2 is up, the request is sent to Service-HTTP-2.
- If the Service-HTTP-1 is up, the request is sent to Service-HTTP-1.
- If Service-HTTP-1 and Service-HTTP-2 are up at the same time, the request is sent to Service-HTTP-1.

To configure the URL hash method, see Configuring a Load Balancing Method that Does Not Include a Policy. Select the load balancing method as URL Hash, and set the hash length to the number of bytes to be used for generating the hash value.

The Domain Hash Method

A load balancing virtual server configured to use the domain hash method uses the hashed value of the domain name in the HTTP request to select a service. The domain name is taken from either the incoming URL or the Host header of the HTTP request. If the domain name appears in both the URL and the Host header, the appliance gives preference to the URL.

If you configure domain name hashing, and an incoming HTTP request does not contain a domain name, the Citrix ADC appliance defaults to the round robin method for that request.

The hash-value calculation uses the name length or hash length value, whichever is smaller. By default, the Citrix ADC appliance calculates the hash value from the first 80 bytes of the domain name. To specify a different number of bytes in the domain name when calculating the hash value, you can set the hashLength parameter (Hash Length in the configuration utility) to a value of from 1 to 4096 (bytes).

To configure the domain hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Destination IP Hash Method

A load balancing virtual server configured to use the destination IP hash method uses the hashed value of the destination IP address to select a server. You can mask the destination IP address to specify which part of it to use in the hash value calculation, so that requests that are from different networks but destined for the same subnet are all directed to the same server. This method supports IPv4 and IPv6-based destination servers.

This load balancing method is appropriate for use with the cache redirection feature.

To configure the destination IP hash method for an IPv4 destination server, you set the netMask parameter. To configure this method for an IPv6 destination server, you use the v6NetMaskLen parameter. In the configuration utility, text boxes for setting these parameters appear when you select the **Destination IP Hash method**.

To configure the destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Hash Method

A load balancing virtual server configured to use the source IP hash method uses the hashed value of the client IPv4 or IPv6 address to select a service. To direct all requests from source IP addresses that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Destination IP Hash Method

A load balancing virtual server configured to use the source IP destination IP hash method uses the hashed value of the source and destination IP addresses (either IPv4 or IPv6) to select a service. Hashing is symmetric. The hash value is the same regardless of the order of the source and destination IPs. This ensures that all packets flowing from a particular client to the same destination are directed to the same server.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Source Port Hash Method

A load balancing virtual server configured to use the source IP source port hash method uses the hash value of the source IP (either IPv4 or IPv6) and the source port to select a service. This ensures that all packets on a particular connection are directed to the same service.

This method is used in connection mirroring and firewall load balancing. For more information about connection mirroring, see Connection Failover.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP source port hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Call ID Hash Method

A load balancing virtual server configured to use the call ID hash method uses the hash value of the call ID in the SIP header to select a service. Packets for a particular SIP session are therefore always directed to the same proxy server.

This method is applicable to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure the call ID hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Least bandwidth method

September 14, 2021

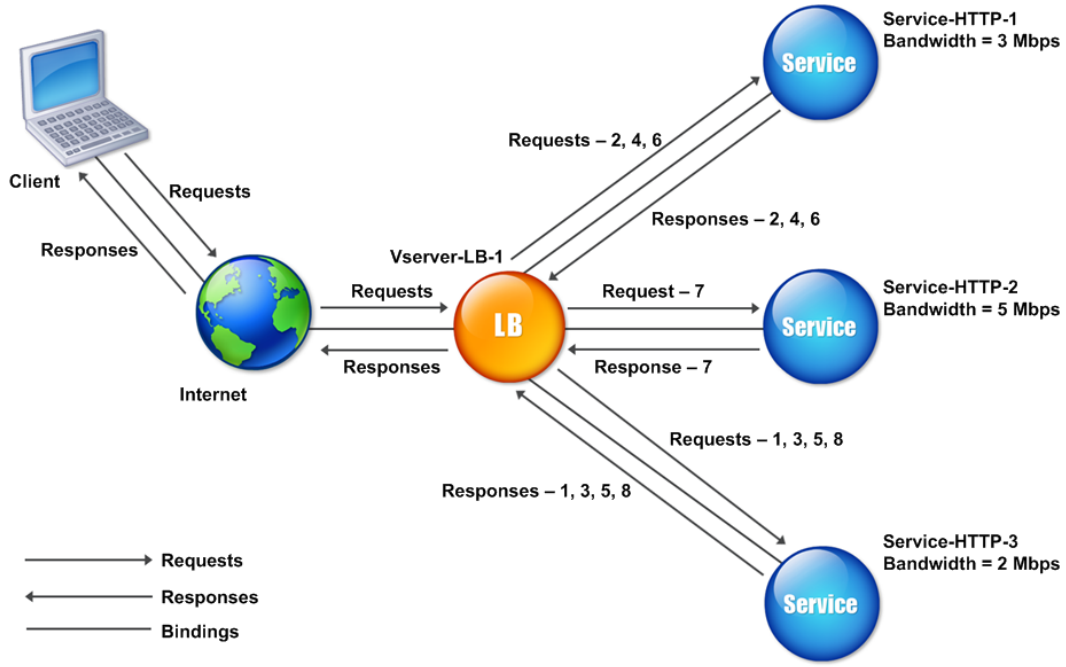
A load balancing virtual server configured to use the least bandwidth method selects the service that is currently serving the least amount of traffic, measured in megabits per second (Mbps). The following example shows how the virtual server selects a service for load balancing by using the least bandwidth method.

Consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has 3 Mbps bandwidth.
- Service-HTTP-2 has 5 Mbps bandwidth.
- Service-HTTP-3 has 2 Mbps bandwidth.

The following diagram illustrates how the virtual server uses the least bandwidth method to forward requests to the three services.

Figure 1. How the Least Bandwidth Load Balancing Method Works



The virtual server selects the service by using the bandwidth value (N), which is the sum of the number of bytes transmitted and received over the previous 14 seconds. If each request requires 1 Mbps bandwidth, the Citrix ADC appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Because Service-HTTP-1 and Service-HTTP-3 now have the same N value, the virtual server switches to the round robin method for these servers, alternating between them. Service-HTTP-1 receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 now all have the same N value, the virtual server includes Service-HTTP-2 in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

Request Received	Service Selected	Current N Value	Remarks
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.

Request Received	Service Selected	Current N Value	Remarks
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3;(N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

Note: If you enable the RTSP NAT option on the virtual server, the Citrix ADC appliance uses the number of data and control bytes exchanged to determine the bandwidth usage for RTSP services. For more information about the RTSP NAT option, see Managing RTSP Connections.

The Citrix ADC appliance also performs load balancing by using the bandwidth and weights if different weights are assigned to the services. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The Citrix ADC appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.

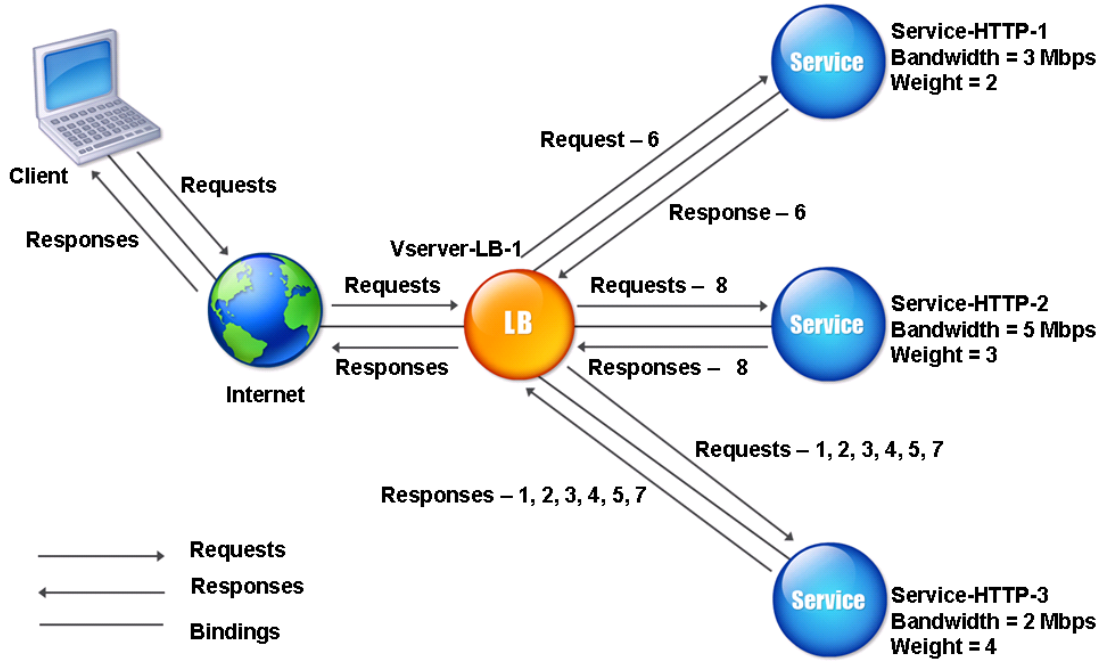
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3; (Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least bandwidth method when weights are assigned to the services.

Figure 2. How the Least Bandwidth Load Balancing Method Works When Weights Are Assigned



To configure the least bandwidth method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Least packets method

September 14, 2021

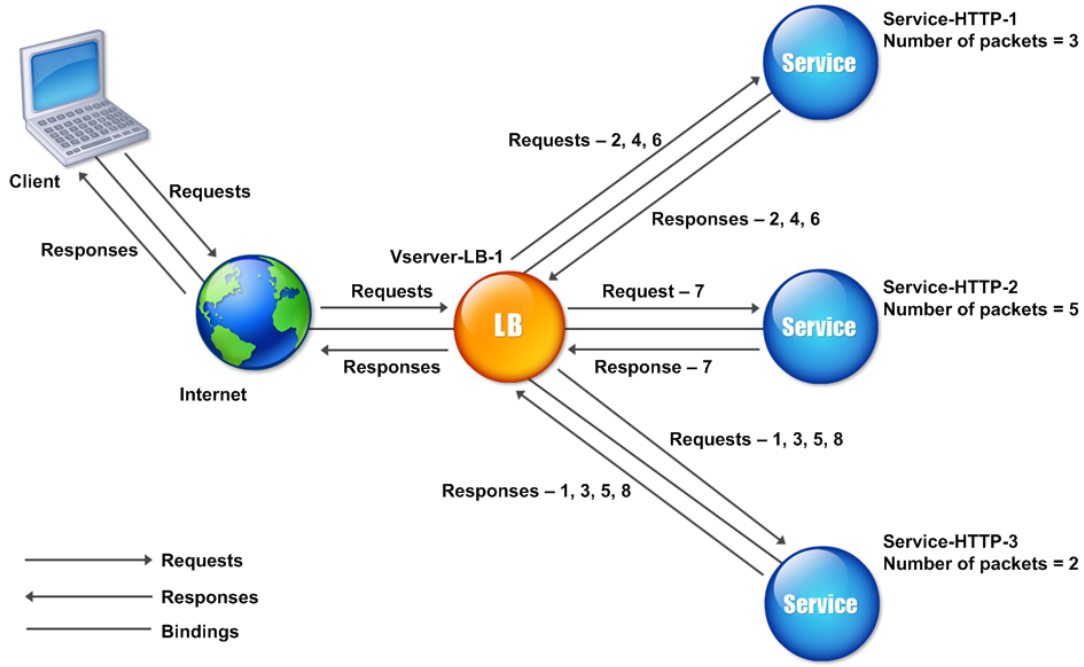
A load balancing virtual server configured to use the least packets method selects the service that has received the fewest packets in the last 14 seconds.

For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has handled three packets in the last 14 seconds.
- Service-HTTP-2 has handled five packets in the last 14 seconds.
- Service-HTTP-3 has handled two packets in the last 14 seconds.

The following diagram illustrates how the Citrix ADC appliance uses the least packets method to choose a service for each request that it receives.

Figure 1. How the Least Packets Load Balancing Method Works



The Citrix ADC appliance selects a service by using the number of packets (N) transmitted and received by each service in the last 14 seconds. Using this method, it delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Since Service-HTTP-1 and Service-HTTP-3 now have the same N value, the virtual server switches to the round robin method. Service-HTTP-1 therefore receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have the same N value, the virtual server switches to the round robin method for Service-HTTP-2 as well, including it in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

Request Received	Service Selected	Current N Value	Remarks
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.

Request Received	Service Selected	Current N Value	Remarks
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3; (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

Note: If you enable the RTSP NAT option on the virtual server, the appliance uses the number of data and control packets to calculate the number of packets for RTSP services. For more information about the RTSP NAT option, see Managing RTSP Connections.

The Citrix ADC appliance also performs load balancing by using the number of packets and weights when a different weight is assigned to each service. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The Citrix ADC appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.

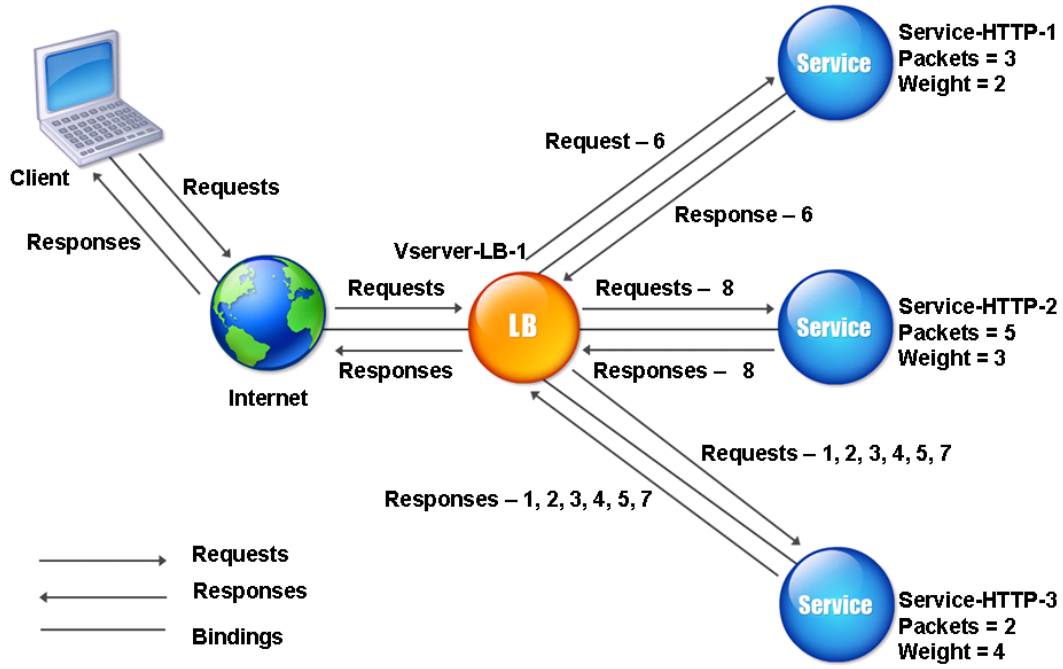
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3;(Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least packets method when weights are assigned.

Figure 2. How the Least Packets Method Works When Weights Are Assigned



To configure the least packets method, see Configuring a Load Balancing Method that Does Not Include a Policy.

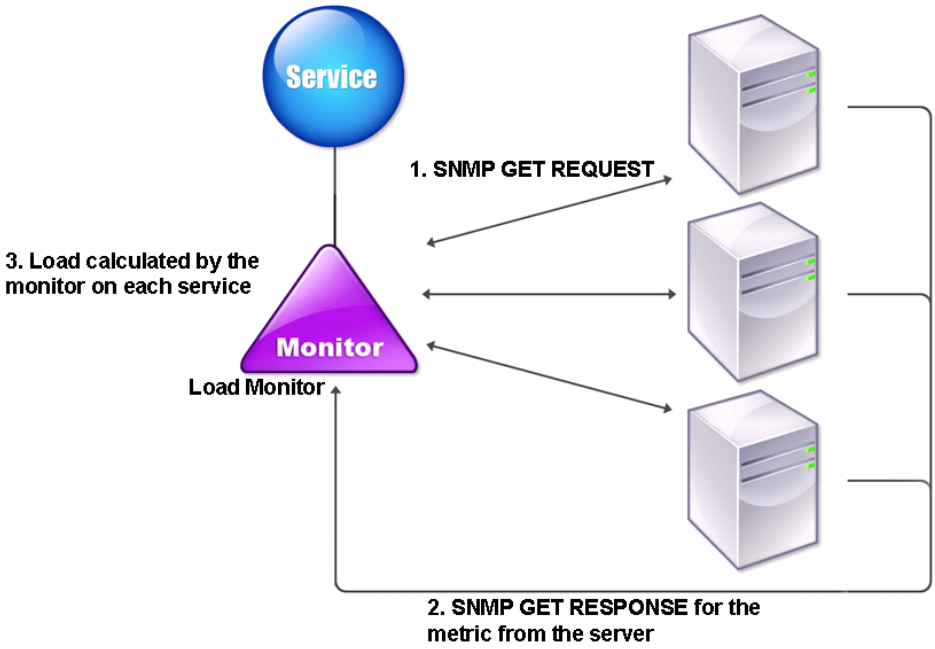
Custom load method

September 14, 2021

Custom load balancing is performed on server parameters such as CPU usage, memory, and response time. When using the custom load method, the Citrix ADC appliance usually selects a service that is not handling any active transactions. If all the services in the load balancing setup are handling active transactions, the appliance selects the service with the smallest load. A special type of monitor, known as a load monitor, calculates the load on each service in the network. The load monitors do not mark the state of a service, but they do take services out of the load balancing decision when those services are not UP.

For more information about load monitors, see Understanding Load Monitors. The following diagram illustrates how a load monitor operates.

Figure 1. How Load Monitors Operate



The load monitor uses SNMP probes to calculate the load on each service by sending an SNMP GET request to the service. This request contains one or more object IDs (OIDs). The service responds with an SNMP GET response, with metrics corresponding to the SNMP OIDs. The load monitor uses the response metrics to calculate the load on the service.

The load monitor calculates the load on a service by using the following parameters:

- Metrics values retrieved through SNMP probes that exist as tables in the Citrix ADC appliance.
- Threshold value set for each metric.
- Weight assigned to each metric.

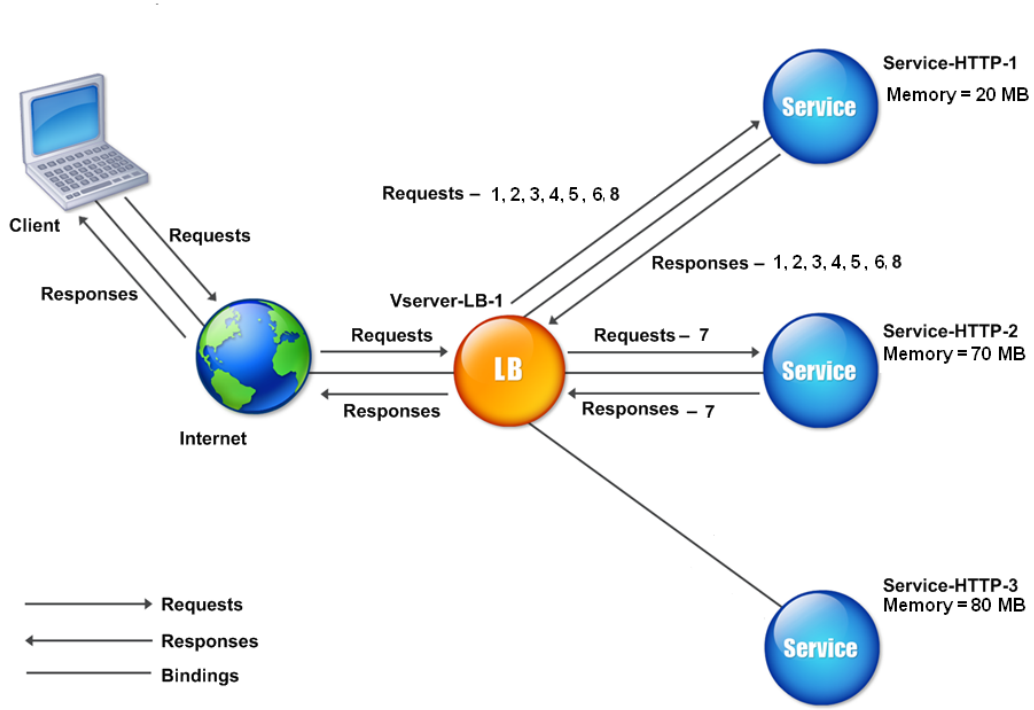
For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 is using 20 MB of memory.
- Service-HTTP-2 is using 70 MB of memory.
- Service-HTTP-3 is using 80 MB of memory.

The load balanced servers can export metrics such as CPU and memory usage to the services, which can in turn provide them to the load monitor. The load monitor sends an SNMP GET request containing the OIDs 1.3.6.1.4.1.5951.4.1.1.41.1.5, 1.3.6.1.4.1.5951.4.1.1.41.1.4, and 1.3.6.1.4.1.5951.4.1.1.41.1.3 to the services. SNMP OIDs of type STRING are not supported, because you cannot calculate the load by using a STRING OID. Loads can be calculated by using other data types, such as INT and gauge32. The

three services respond to the request. The Citrix ADC appliance compares the exported metrics, and then selects Service-HTTP-1 because it has more available memory. The following diagram illustrates this process.

Figure 2. How the Custom Load Method Works



If each request uses 10 MB memory, the Citrix ADC appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, and fifth requests, because this service has the lowest N value.
- Service-HTTP-1 and Service-HTTP-2 now have the same load, so the virtual server reverts to the round robin method for these servers. Therefore, Service-HTTP-2 receives the sixth request, and Service-HTTP-1 receives the seventh request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have the same load, the virtual server reverts to the round robin method for Service-HTTP-3 as well. Therefore, Service-HTTP-3 receives the eighth request.

The following table summarizes how N is calculated.

Request received	Service selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-1; (N = 20)	N = 30	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1; (N = 30)	N = 40	-
Request-3	Service-HTTP-1; (N = 40)	N = 50	-
Request-4	Service-HTTP-1; (N = 50)	N = 60	-
Request-5	Service-HTTP-1; (N = 60)	N = 70	-
Request-6	Service-HTTP-1; (N = 70)	N = 80	Service-HTTP-2 and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2; (N = 70)	N = 80	Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-1; (N = 80)	N = 90	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.

If different weights are assigned to the services, the custom load algorithm considers both the load on each service and the weight assigned to each service. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 4, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 2. If each request uses 10 MB memory, the Citrix ADC appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, fifth, sixth, seventh, and eighth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the ninth request, because this service has the lowest Nw value.

Service-HTTP-3 has the highest Nw value, and is therefore not considered for load balancing.

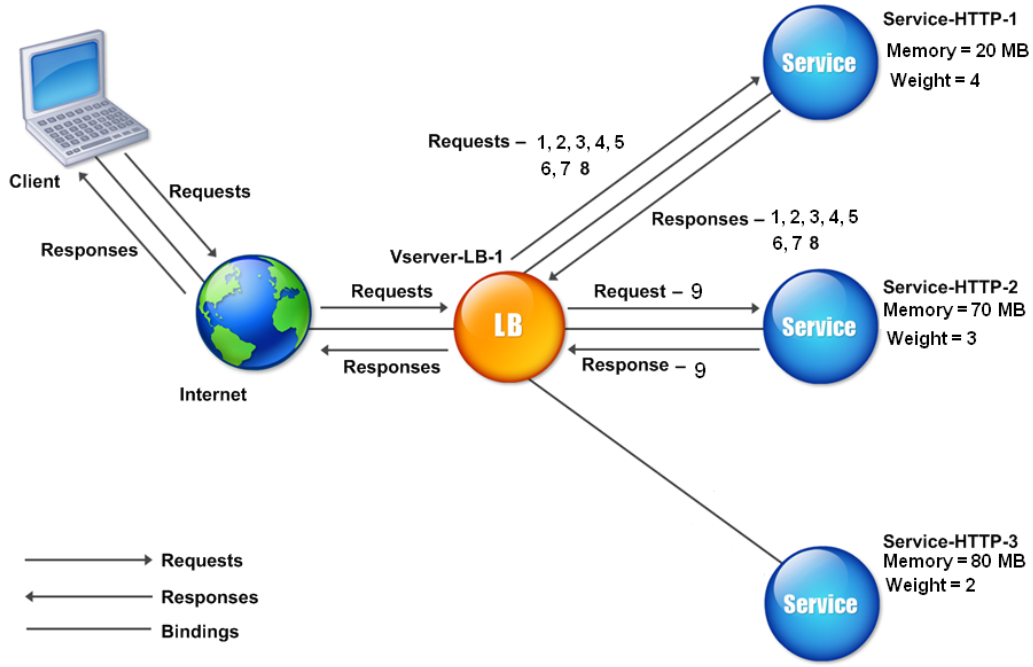
The following table summarizes how Nw is calculated.

Request received	Service selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-1; (Nw = 50000)	Nw = 75000	Service-HTTP-1 has the lowest Nw value.
Request-2	Service-HTTP-1; (Nw = 5000)	Nw = 100000	-
Request-3	Service-HTTP-1; (Nw = 15000)	Nw = 125000	-
Request-4	Service-HTTP-1; (Nw = 20000)	Nw = 150000	-
Request-5	Service-HTTP-1; (Nw = 23333.34)	Nw = 175000	-
Request-6	Service-HTTP-1; (Nw = 25000)	Nw = 200000	-
Request-7	Service-HTTP-1; (Nw = 23333.34)	Nw = 225000	-
Request-8	Service-HTTP-1; (Nw = 25000)	Nw = 250000	-
Request-9	Service-HTTP-2; (Nw = 233333.34)	Nw = 266666.67	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-2 and Service-HTTP-3) is equal to 400,000.

The following diagram illustrates how the Citrix ADC appliance uses the custom load method when weights are assigned.

Figure 3. How the Custom Load Method Works When Weights Are Assigned



To configure the custom load method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Static proximity method

September 14, 2021

When a virtual server is configured to use the static proximity method, it selects the service that best matches the proximity criteria.

For the static proximity method to work, you must either configure the Citrix ADC appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the load balancing method.

For more details, see the following topics.

- Adding a Location File to Create a Static Proximity Database
- Adding Custom Entries to a Static Proximity Database
- Setting the Location Qualifiers

- Specifying the Static Proximity method

Specifying the Proximity Method

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the command line interface

At the command prompt, type the following commands to configure static proximity and verify the configuration:

```
set lb vserver <name> -lbMethod STATICPROXIMITY

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod STATICPROXIMITY

show lb vserver
<!--NeedCopy-->
```

To specify static proximity by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and select the virtual server.
2. Click **Edit** and expand the **Method** section.
3. In the **Load Balancing Method** list, select **STATICPROXIMITY**.

Token method

September 14, 2021

A load balancing virtual server configured to use the token method bases its selection of a service on the value of a data segment extracted from the client request. The data segment is called the token. You configure the location and size of the token. For subsequent requests with the same token, the virtual server chooses the same service that handled the initial request.

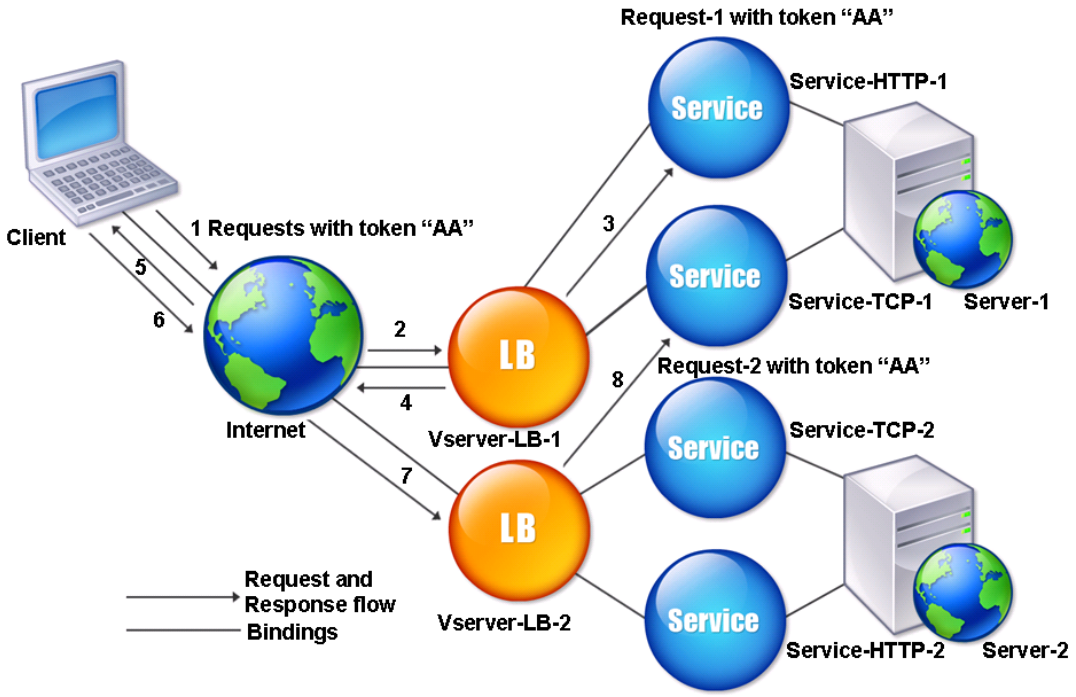
This method is content aware. It operates differently for TCP, HTTP, and HTTPS connections. For HTTP or HTTPS services, the token is found in the HTTP headers, the URL, or the BODY. To locate the token, you specify or create a classic or advanced expression. For more information on classic or advanced expressions, see Policy Configuration and Reference.

For HTTP services, the virtual server searches for the configured token in the first 24 kilobytes (KB) of the TCP payload. For non-HTTP (TCP, SSL, and SSL_TCP) services, the virtual server searches for the configured token in the first 16 packets if the total size of the 16 packets is less than 24 KB. But if the total size of the 16 packets is greater than 24 KB, the appliance searches for the token in the first 24 KB of payload. You can use this load balancing method across virtual servers of different types to make sure that requests presenting the same token are directed to appropriate services, regardless of the protocol used.

For example, consider a load balancing setup consisting of servers that contain Web content. You want to configure the Citrix ADC appliance to search for a specific string (the token) inside the URL query portion of the request. Server-1 has two services, Service-HTTP-1 and Service-TCP-1, and Server-2 has two services, Service-HTTP-2 and Service-TCP-2. The TCP services are bound to Vserver-LB-2, and the HTTP services are bound to Vserver-LB-1.

If Vserver-LB-1 receives a request with the token AA, it selects the service Service-HTTP-1 (bound to server-1) to process the request. If Vserver-LB-2 receives a different request with the same token (AA), it directs this request to the service Service-TCP-1. The following diagram illustrates this process.

Figure 1. How the Token Method Works



To configure the Token load balancing method by using the command line interface

At the command prompt, type the following commands to configure the token load balancing method and verify the configuration:

```
set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length>
     -dataoffset <offset>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -
   dataoffset 25

show lb vserver LB-VServer-1
<!--NeedCopy-->
```


To configure the token load balancing method by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, click Method
3. In the Load Balancing Method list, select Token, and specify an expression.

Configure a load balancing method that does not include a policy

September 14, 2021

After you select a load balancing algorithm for your load balancing setup, you must configure the Citrix ADC appliance to use that algorithm. You can configure it by using the CLI or by using the configuration utility.

Note:

The token method is policy based and requires more configuration than is described here. To configure the token method, see Token method.

For some hash-based methods, you can mask an IP address to direct requests belonging to the same subnet to the same server. For more information, see Hashing methods.

To set the load balancing method by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -lbMethod <method>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
<!--NeedCopy-->
```

To set the load balancing method by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, click **Method**, and in the Load Balancing Method list, select a method.

Persistence and persistent connections

September 14, 2021

A load balancing stateless protocol, such as HTTP, disrupts the maintenance of state information about client connections if persistence is not configured. Different transmissions from the same client might be directed to different servers even though all the transmissions are part of the same session. You can configure persistence on a load balancing virtual server that handles certain types of Web applications, such as shopping cart applications.

Before you can configure persistence, you need to understand the different types of persistence, how they are used, and what the implications of each type is. You then need to configure the Citrix ADC appliance to provide persistent connections for those websites and Web applications that require them.

You can also configure backup persistence, which takes effect if the primary type of persistence configured for a load balancing virtual server fails. You can configure persistence groups, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

For information about persistence with RADIUS load balancing, see Configuring RADIUS Load Balancing with Persistence.

About Persistence

September 14, 2021

You can choose from among any of several types of persistence for a given load balancing virtual server, which then routes to the same service all connections from the same user to your shopping cart application, web-based email, or other network application. The persistence session remains in effect for the time which you specify.

If a server participating in a persistence session goes DOWN, the load balancing virtual server uses the configured load balancing method to select a new service, and establishes a new persistence session with the server represented by that service. If the server goes OUT OF SERVICE, it continues to process existing persistence sessions, but the virtual server does not direct any new traffic to it. After the shutdown period elapses, the virtual server ceases to direct connections from existing clients to the service, closes existing connections, and redirects those clients to new services if necessary.

Depending on the persistence type you configure, the Citrix ADC appliance might examine the source IPs, destination IPs, SSL session IDs, Host or URL headers, or some combination of these things to place each connection in the proper persistence session. It might also base persistence on a cookie issued by the Web server, on an arbitrarily assigned token, or on a logical rule. Almost anything that

allows the appliance to match connections with the proper persistence session and is used as the basis for persistence.

The following table summarizes the persistence types available on the Citrix ADC appliance.

Persistence Type	Description
Source IP	SOURCEIP. Connections from the same client IP address are parts of the same persistence session.
HTTP Cookie	COOKIEINSERT. Connections that have the same HTTP Cookie header are parts of the same persistence session.
SSL Session ID	SSLSESSION. Connections that have the same SSL Session ID are parts of the same persistence session.
URL Passive	URLPASSIVE. Connections to the same URL are treated as parts of the same persistence session.
Custom Server ID	CUSTOMSERVERID. Connections with the same HTTP HOST header are treated as parts of the same persistence session.
Destination IP	DESTIP. Connections to the same destination IP are treated as parts of the same persistence session.
Source and Destination IPs	SRCIPDESTIP. Connections that are both from the same source IP and to the same destination IP are treated as parts of the same persistence session.
SIP Call ID	CALLID. Connections that have the same call ID in the SIP header are treated as parts of the same persistence session.
RTSP Session ID	RTSPSID. Connections that have the same RTSP Session ID are treated as parts of the same persistence session.
User-Defined Rule	RULE. Connections that match a user-defined rule are treated as parts of the same persistence session.

Table 1. Types of Persistence

Depending on the type of persistence that you have configured, the virtual server can support either 250,000 simultaneous persistent connections or any number of persistent connections up to the limits imposed by the amount of RAM on your Citrix ADC appliance. The following table shows which types of persistence fall into each category.

Persistence Type	Number of Simultaneous Persistent Connections Supported
Source IP, SSL Session ID, Rule, destination IP, source IP/destination IP, SIP Call ID, RTSP Session ID	250 K
Cookie, URL Server ID, Custom Server ID	Memory limit. In CookieInsert, if timeout is not 0, the number of connections is limited by memory.

Table 2. Persistence Types and Numbers of Simultaneous Connections Supported

Some types of persistence are specific to particular types of virtual server. The following table lists each type of persistence and indicates which types of persistence are supported on which types of virtual server.

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEI	YES	YES	YES	YES	YES	YES	NO	NO
COOKIEINSERT	YES	YES	NO	NO	NO	NO	NO	NO
SSLSESS	NO	YES	NO	NO	YES	YES	NO	NO
URLPASSIVE	S	YES	NO	NO	NO	NO	NO	NO
CUSTOM	YES	YES	NO	NO	NO	NO	NO	NO
RULE	YES	YES	YES	NO	NO		NO	NO
SRCIPDE	YES	YES	YES	YES	YES	YES	NO	NO
DESTIP	YES	YES	YES	YES	YES	YES	NO	NO
CALLID	NO	NO	NO	NO	NO	NO	NO	YES
RTSPID	NO	NO	NO	NO	NO	NO	YES	NO

Table 3. Relationship of Persistence Type to Virtual Server Type

Source IP address persistence

September 14, 2021

When source IP persistence is configured, the load balancing virtual server uses the configured load balancing method to select a service for the initial request, and then uses the source IP address (client IP address) to identify subsequent requests from that client and send them to the same service. You can set a time-out value, which specifies the maximum inactivity period for the session. When the time-out value expires, the session is discarded, and the configured load balancing algorithm is used to select a new server.

Caution: In some circumstances, using persistence based on source IP address can overload your servers. All requests to a single website or application are routed through the single gateway to the Citrix ADC appliance, even though they are then redirected to multiple locations. In multiple proxy environments, client requests frequently have different source IP addresses even when they are sent from the same client, resulting in rapid multiplication of persistence sessions where a single session must be created. This issue is called the "Mega Proxy problem." You can use HTTP cookie-based persistence instead of Source IP-based persistence to prevent this from happening.

To configure persistence based on Source IP Address, see Configuring Persistence Types That Do Not Require a Rule.

Note: If all incoming traffic comes from behind a Network Address Translation (NAT) device or proxy, the traffic appears to the Citrix ADC appliance to come from a single source IP address. This prevents Source IP persistence from functioning properly. Where this is the case, you must select a different persistence type.

HTTP cookie persistence

September 14, 2021

When HTTP cookie persistence is configured, the Citrix ADC appliance sets a cookie in the HTTP headers of the initial client request. The cookie contains the IP address and port of the service selected by the load balancing algorithm. As with any HTTP connection, the client then includes that cookie with any subsequent requests.

When the Citrix ADC appliance detects the cookie, it forwards the request to the service IP and port in the cookie, maintaining persistence for the connection. You can use this type of persistence with virtual servers of type HTTP or HTTPS. This persistence type does not consume any appliance resources and therefore can accommodate an unlimited number of persistent clients.

Note: If the client's Web browser is configured to refuse cookies, HTTP cookie-based persistence does not work. It might be advisable to configure a cookie check on the website, and warn clients that do not appear to be storing cookies properly that they need to enable cookies for the website if they want to use it.

The format of the cookie that the Citrix ADC appliance inserts is:

```
NSC_XXXX=<ServiceIP ><ServicePort>
```

Where:

- NSC_XXXX is the virtual server ID that is derived from the virtual server name.
- ServiceIP and ServicePort are encoded representations of the service IP address and service port, respectively. The IP address and port are encoded separately.

You can set a time-out value for this type of persistence to specify an inactivity period for the session. When the connection has been inactive for the specified period, the Citrix ADC appliance discards the persistence session. Any subsequent connection from the same client results in a new server being selected based on the configured load balancing method, and a new persistence session being established.

Note: If you set the time-out value to 0, the Citrix ADC appliance does not specify an expiration time, but sets a session cookie that is not saved when the client's browser is shut down.

By default, the Citrix ADC appliance sets HTTP version 0 cookies for maximum compatibility with client browsers. (Only certain HTTP proxies understand version 1 cookies; most commonly used browsers do not.) You can configure the appliance to set HTTP version 1 cookies, for compliance with RFC2109. For HTTP version 0 cookies, the appliance inserts the cookie expiration date and time as an absolute Coordinated Universal Time (GMT). It calculates this value as the sum of the current GMT time on the appliance and the time-out value. For HTTP version 1 cookies, the appliance inserts a relative expiration time by setting the "Max-Age" attribute of the HTTP cookie. In this case, the client's browser calculates the actual expiration time.

To configure persistence based on a cookie inserted by the appliance, see Configuring Persistence Types That Do Not Require a Rule.

In the HTTP cookie, the appliance by default sets the `HTTPOnly` flag to indicate that the cookie is nonscriptable and must not be revealed to the client application. Therefore, a client-side script cannot access the cookie, and the client is not susceptible to cross-site scripting.

Certain browsers, however, do not support the `HTTPOnly` flag and, therefore, might not return the cookie. As a result, persistence is broken. For browsers that do not support the flag, you can omit the `HTTPOnly` flag in the persistence cookie.

To change the HTTPOnly flag setting by using the CLI

At the command prompt, type:


```
set lb parameter -httpOnlyCookieFlag (ENABLED|DISABLED)
<!--NeedCopy-->
```

Example:

```
>  set lb parameter -httpOnlyCookieFlag disabled
   Done
> show lb parameter
  Global LB parameters:
       Persistence Cookie HttpOnly Flag: DISABLED
       Use port for hash LB: YES
 Done
<!--NeedCopy-->
```

To change the HTTPOnly flag setting by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Configure Load Balancing Parameters**, and select or clear the **Persistence Cookie HTTPOnly** flag.

Encrypting the Cookie

From release 10.5 build 55.8, you can encrypt the cookie in addition to any SSL encryption.

To encrypt the cookie by using the command line interface, at the command prompt, type

```
set lb parameter -UseEncryptedPersistenceCookie ENABLED -
   cookiePassphrase test
<!--NeedCopy-->
```

To encrypt the cookie by using the configuration utility

1. Navigate to **Traffic Management** > **Change Load Balancing Parameters**, and select **Encode Persistence Cookie Values** and enter a passphrase in **Cookie Passphrase**.

SSL session ID persistence

September 14, 2021

When SSL session ID persistence is configured, the Citrix ADC appliance uses the SSL session ID, which is part of the SSL handshake process, to create a persistence session before the initial request is directed to a service. The load balancing virtual server directs subsequent requests that have the same SSL session ID to the same service. This type of persistence is used for SSL bridge services.

Note:

There are two issues that users must consider before choosing this type of persistence. First, this type of persistence consumes resources on the Citrix ADC appliance, which limits the number of concurrent persistence sessions that it can support. If you expect to support multiple persistence sessions, you might want to choose another type of persistence.

Second, if the client and the load-balanced server must renegotiate the session ID during their transactions, persistence is not maintained, and a new persistence session is created when the client's next request is received. This might result in the client's activity on the website being interrupted and the client might be asked to reauthenticate or restart the session. It might also result in large numbers of abandoned sessions if the timeout is set to too large a value.

To configure persistence based on SSL session ID, see Configuring Persistence Types That Do Not Require a Rule.

> **Note**
>
> SSL session ID persistence is not supported with session tickets.

Back up persistence support for SSL session ID

From NetScaler release 12.0 build 56.20, source IP persistence is supported as a backup persistence type for SSL session ID persistence. If the client and load-balanced server renegotiate the session, and source IP persistence is configured as the backup persistence, client requests are forwarded to the same server.

To support backup persistence for SSL session ID, the Citrix ADC appliance creates session entries for both source IP and SSL session ID when a client request is received for the first time. For the subsequent requests containing the same session ID, the SSL session ID is used. However, when the client and the load-balanced server renegotiate the session, the client request is forwarded to the same server by using the Source IP persistence and a new SSL Session ID persistence entry is created.

For information about configuring backup persistence, see Configuring Backup Persistence.

Diameter AVP number persistence

September 14, 2021

You can use persistence based on the Attribute-Value Pair (AVP) number of a Diameter message to create persistent Diameter sessions. When the Citrix ADC appliance finds the AVP in the Diameter message, it creates a persistence session based on the value of the AVP. All subsequent messages that match the value of the AVP are directed to the previously selected server. If the value of the AVP does not match the persistence session, a new session is created for the new value.

Note: If the AVP number is not defined in Diameter base-protocol RFC 6733, and if the number is nested inside a grouped AVP, you must define a sequence of AVP numbers (maximum of 3) in parent-to-child order. For example, if the persist AVP number X is nested inside AVP Y, which is nested in Z, define the list as Z Y X.

To configure Diameter-based persistence on a virtual server by using the command line interface

At the command prompt, type the following command:

```
1 set lb vserver <name> -PersistenceType <type-> persistAVPno <
     positive_integer>
2 <!--NeedCopy-->
```

Example:

```
1 set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263
2 <!--NeedCopy-->
```

Custom server ID persistence

September 14, 2021

In the Custom Server ID persistence method, the Server ID specified in the client request is used to maintain persistence. For this type of persistence to work, you must first set a server ID on the services. The Citrix ADC appliance checks the URL of the client request and connects to the server associated with the specified server ID. The service provider must make sure that the users are aware of the server IDs to be provided in their requests for specific services.

For example, if your site provides different types of data, such as images, text, and multimedia, from different servers, you can assign each server a server ID. On the Citrix ADC appliance, you specify those server IDs for the corresponding services, and you configure custom server ID persistence on the corresponding load balancing virtual server. When sending a request, the client inserts the server ID into the URL indicating the required type of data.

To configure custom server ID persistence:

- In your load balancing setup, assign a server ID to each service for which you want to use the user-defined server ID to maintain persistence. Alphanumeric server IDs are allowed.
- Specify rules, in the default-syntax expression language, to examine the URL queries for the server ID and forward traffic to the corresponding server.
- Configure custom server ID persistence.

Note: The persistence time-out value does not affect the Custom Server ID persistence type. There is no limit on the maximum number of persistent clients because this persistence type does not store any client information.

Example:

In a load balancing setup with two services, assign server ID 2345-photo-56789 to Service-1, and server ID 2345-drawing-abb123 to Service-2. Bind these services to a virtual server named Web11.

```
set service Service-1 10.102.29.5 -CustomServerID 2345-photo-56789

set service Service-2 10.102.29.6 -CustomServerID 2345-drawing-abb123
<!--NeedCopy-->
```

On virtual server Web11, enable Custom Server ID persistence.

Create the following expression so that all URL queries containing the string “sid=” are examined.

HTTP.REQ.URL.AFTER_STR(“sid=”)

Example:

```
set lb vserver Web11 -persistenceType customserverID -rule "HTTP.REQ.
   URL.AFTER_STR("sid=")"

bind lb vserver Web11 Service-[1-2]
<!--NeedCopy-->
```

When a client sends a request with the following URL to the IP address of Web11, the appliance directs the request to Service-2 and honors persistence.

Example:

```
http://www.example.com/index.asp?&sid=2345-drawing-abb123
```

For more information about default-syntax policy expressions, see the Policy Configuration and Reference.

To configure custom server ID persistence by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Services**.

2. Open the service and set a server ID.
3. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
4. In Advanced Settings, select Persistence.
5. Select CUSTOMESERVERID, and specify an expression.

IP address persistence

September 14, 2021

You can base persistence on Destination IP addresses, or on both Source IP and Destination IP Addresses.

Persistence Based on Destination IP Addresses

With destination IP address-based persistence, when the Citrix ADC appliance receives a request from a new client, it creates a persistence session based on the IP address of the service selected by the virtual server (the destination IP address). Later, it directs requests to the same destination IP to the same service. This type of persistence is used with link load balancing. For more information about link load balancing, see Link Load Balancing.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on the destination IP address, see Configuring Persistence Types That Do Not Require a Rule.

Persistence Based on Source and Destination IP Addresses

With source and destination IP address-based persistence, when the Citrix ADC appliance receives a request, it creates a persistence session based on both the IP address of the client (the source IP address) and the IP address of the service selected by the virtual server (the destination IP address). Later, it directs requests from the same source IP and to the same destination IP to the same service.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on both source and destination IP addresses, see Configuring Persistence Types That Do Not Require a Rule.

SIP Call ID persistence

September 14, 2021

With SIP Call ID persistence, the Citrix ADC appliance chooses a service based on the call ID in the SIP header. This enables it to direct packets for a particular SIP session to the same service and, therefore, to the same load balanced server. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure persistence based on SIP Call ID, see Configuring Persistence Types That Do Not Require a Rule.

RTSP session ID persistence

September 14, 2021

With RTSP Session ID persistence, when the Citrix ADC appliance receives a request from a new client, it creates a persistence session based on the Real-Time Streaming Protocol (RTSP) session ID in the RTSP packet header, and then directs the request to the RTSP service selected by the configured load balancing method. It directs subsequent requests that contain the same session ID to the same service. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

Note: RTSP Session ID persistence is configured by default on RTSP virtual servers, and you cannot modify that setting.

Sometimes different RTSP servers issue the same session IDs. When this happens, unique sessions cannot be created between the client and the RTSP server by using only the RTSP session ID. If you have multiple RTSP servers that might issue the same session IDs, you can configure the appliance to append the server IP address and port to the session ID, creating a unique token that can be used to establish persistence. This is called session ID mapping.

To configure persistence based on RTSP Session IDs, see Configuring Persistence Types That Do Not Require a Rule.

Important: If you need to use session ID mapping, you must set the following parameter when configuring each service within the load balancing setup. Also, make sure that no non-persistent connections are routed through the RTSP virtual server.

Configure URL passive persistence

September 14, 2021

With URL Passive persistence, when the Citrix ADC appliance receives a request from a client, it extracts the server IP address-port information (expressed as a single hexadecimal number) from the client request.

URL passive persistence requires configuring an advanced expression that specifies the query element that contains the server IP address-port information. For more information about classic and advanced policy expressions, see Policies and Expressions.

The following expression configures the appliance to examine requests for URL queries that contain the string "urlp=", extract the server IP address-port information, convert it from a hexadecimal string to an IP and port number, and forward the request to the service configured with this IP address and port number.

HTTP.REQ.URL.AFTER_STR("urlp=")

If URL passive persistence is enabled and the previous expression is configured, a request with the following URL and server IP address-port string is directed to 10.102.29.10:80.

```
http://www.example.com/index.asp?&urlp=0A661D0A0050
```

The persistence time-out value does not affect this persistence type. Persistence is maintained as long as the server IP address-port information can be extracted from client requests. This persistence type does not consume any appliance resources, so it can accommodate an unlimited number of persistent clients.

To configure URL passive persistence, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule. You set the persistence type to URLPASSIVE. You then perform the following procedures.

To configure URL passive persistence by using the CLI

At the command prompt, type:

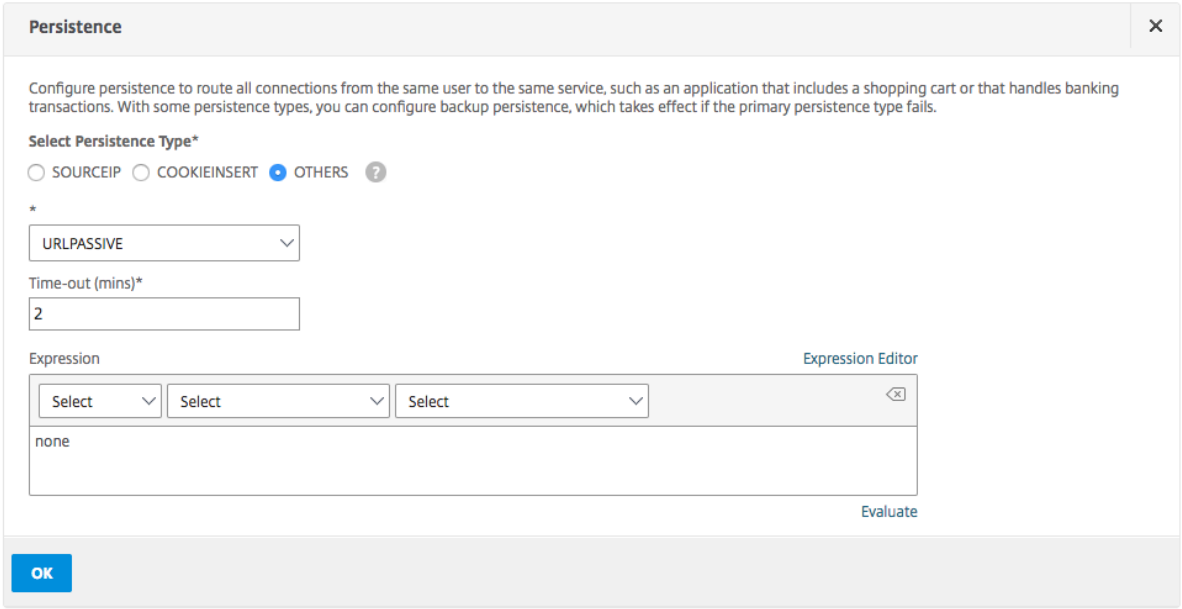
```
set lb vserver <vserverName>  [-persistenceType <persistenceType>]  [-
    rule <expression>]
<!--NeedCopy-->
```

Example:

```
set lb vserver LB-VServer-1 -persistenceType URLPASSIVE  -rule HTTP.REQ
    .URL.AFTER_STR( "urlp=" )
<!--NeedCopy-->
```


To configure persistence on a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In the Persistence section, choose the persistence type that meets your requirement. The most suitable persistence type for the virtual server is available as option buttons. Other persistence types that are applicable to the specific virtual server type can be selected from the Others list.



> **Note:**
>
> Prior to NetScaler release 12.0 build 56.20, all persistence types are available in a single Persistence drop-down list without any option buttons.

Configure persistence based on user-defined rules

September 20, 2021

> **Warning:**
>
> The use of Classic expressions for the persistence rule in the load balancing feature is deprecated. However Classic expressions are removed and no longer available for the filter rule on the Citrix ADC appliance release 13.1 onwards.

When rule based persistence is configured, the Citrix ADC appliance creates a persistence session based on the contents of the matched rule before directing the request to the service selected by the configured load balancing method. Later, it directs all requests that match the rule to the same service. You can configure rule based persistence for services of type HTTP, SSL, RADIUS, ANY, TCP,

and SSL_TCP.

Rule based persistence requires a classic or default syntax expression. You can use a classic expression to evaluate request headers, or you can use a default syntax expression to evaluate request headers, Web form data in a request, response headers, or response bodies. For example, you can use a classic expression to configure persistence based on the contents of the HTTP Host header. You can also use a default syntax expression to configure persistence based on application session information in a response cookie or custom header. For more information on creating and using classic and default syntax expressions, see Policies and Expressions.

The expressions that you can configure depend on the type of service for which you are configuring rule based persistence. For example, certain RADIUS-specific expressions are not allowed for protocols other than RADIUS, and TCP-option based expressions are not allowed for service types other than the ANY type. For TCP and SSL_TCP service types, you can use expressions that evaluate TCP/IP protocol data, Layer 2 data, TCP options, and TCP payloads.

Note: For a use case that involves configuring rule based persistence based on Financial Information eXchange ("FIX") Protocol data transmitted over TCP, see Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream.

Rule based persistence can be used for maintaining persistence with entities such as Citrix SD-WAN appliances, Citrix SD-WAN plug-ins, cache servers, and application servers.

Note: On an ANY virtual server, you cannot configure rule-based persistence for the responses.

To configure persistence based on a user-defined rule, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule, and set the persistence type to RULE. You can then perform the following procedures. You can configure rule based persistence by using the configuration utility or the CLI.

To configure persistence based on user-defined rules by using the CLI

At the command prompt, type:

```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression
    >]
<!--NeedCopy-->
```

Example:

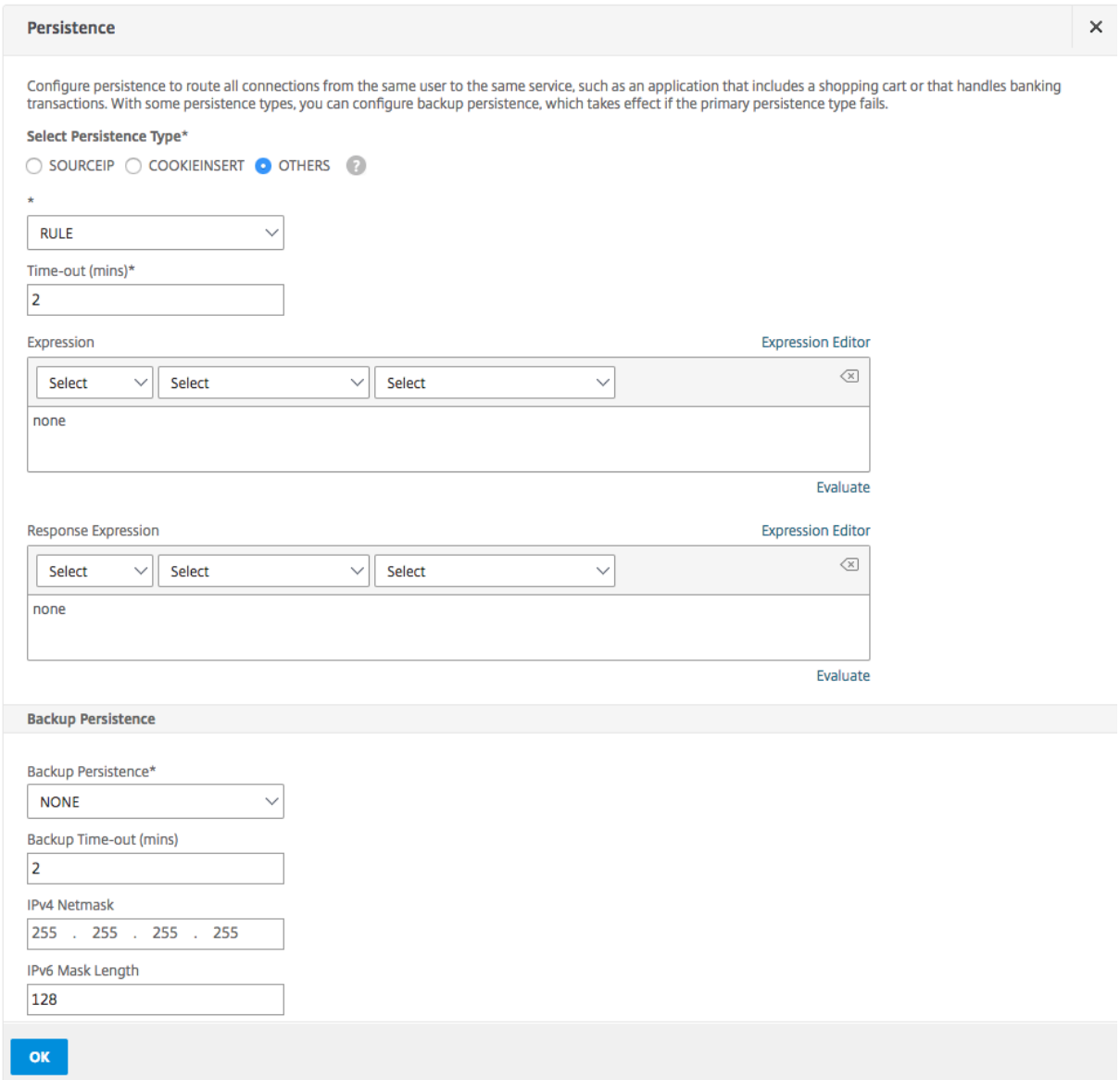
```
set lb vserver vsvr_name -rule http.req.header("cookie").value(0).
    typecast_nvlist_t('=',';').value("server")

set lb vserver vsvr_name -resrule http.res.header("set-cookie").value
    (0).typecast_nvlist_t('=',';').value("server")
```



```
4
5 <!--NeedCopy-->
```

To configure persistence based on user-defined rules by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In the Persistence section, choose the persistence type that meets your requirement. The most suitable persistence type for the virtual server is available as option buttons. Other persistence types that are applicable to the specific virtual server type can be selected from the Others list.



> **Note**
>
> Prior to NetScaler release 12.0 build 56.20, all persistence types are available in a single Persistence drop-down list without any option buttons.

Example: Classic Expression for a Request Payload

The following classic expression creates a persistence session based on the presence of a User-Agent HTTP header that contains the string, “MyBrowser”, and directs any subsequent client requests that contain this header and string to the same server that was selected for the initial request.

```
http header User-Agent contains MyBrowser
<!--NeedCopy-->
```

Example: Default syntax Expression for a Request Header

The following default syntax expression does the same thing as the previous classic expression.

HTTP.REQ.HEADER(“User-Agent”).CONTAINS (“MyBrowser”)

Example: Default syntax Expression for a Response Cookie

The following expression examines responses for “server” cookies, and then directs any requests that contain that cookie to the same server that was selected for the initial request.

HTTP.RES.HEADER(“SET-COOKIE”).VALUE(0).TYPECAST_NVLIST_T(‘=’,’;’).VALUE(“server”)

Configure persistence types that do not require a rule

September 14, 2021

To configure persistence, you must first set up a load balancing virtual server, as described in Setting Up Basic Load Balancing. You then configure persistence on the virtual server.

To configure persistence on a virtual server by using the CLI

At the command prompt, type the following commands to configure persistence and verify the configuration:

```
set lb vserver <name> -PersistenceType <type> [-timeout <integer>]

```



```
3 show lb vserver
4 <!--NeedCopy-->
```

Example:

```
1 set lb vserver Vserver-LB-1 -persistenceType SOURCEIP -timeout 60
2
3 show lb vserver
4 <!--NeedCopy-->
```

Timeout is the time period for which a persistence session is in effect. The timeout default and minimum values (in minutes) vary based on the persistence type as listed in the following table.

persistence type	Default value	Minimum value	Maximum value
Cookie insert/group cookie insert	2	0	1440
Other persistence types	2	2	1440

Note

- Group cookie insert persistence type can be set on the load balancing group.
- For IP-based persistence, you can also set the persistMask parameter.
- The persistence type by default is set to NONE.

To configure persistence on a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In the Persistence section, choose the persistence type that meets your requirement. The most suitable persistence type for the virtual server is available as option buttons. Other persistence types that are applicable to the specific virtual server type can be selected from the **Others** list.

> **Note** Prior to Citrix ADC release 12.0 build 56.20, all persistence types are available in a single Persistence drop-down list without any option buttons.

Configure backup persistence

September 14, 2021

You can configure a virtual server to use the source IP persistence type when the primary persistence type fails.

The following table describes the combinations of primary and secondary backup persistence types, and the conditions when the backup persistence is used.

Primary persistence	Backup persistence	When the primary persistence lookup fails…
Cookie Insert	Source IP	The appliance falls back to source-IP based persistence only when the client browser does not return any cookie in the request. However, if the browser returns a cookie (not necessarily the persistence cookie) it is assumed that the browser supports cookies and hence backup persistence is not triggered.
Rule	Source IP	The appliance uses source-IP based persistence when the parameter specified in the rule is missing in the incoming request.

Note

- If the primary persistence type is HTTP-cookie based persistence, and the backup persistence type is Source IP-based, you can set a timeout value for backup persistence. For instructions, see Setting a Timeout Value for Idle Client Connections.
- You cannot set a timeout value for backup persistence when the primary persistence is rule based, because in that case the timeout value for secondary persistence must be the same as for the primary persistence. Therefore, the primary and secondary expire at the same time.

To set backup persistence for a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -persistenceType <PersistenceType> -
    persistenceBackup <BackupPersistenceType>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -
    persistenceBackup SourceIP
```



```
2
3 set lb vserver Vserver-LB-1 -persistenceType sslsession -
     persistenceBackup SourceIP
4
5 set lb vserver Vserver-LB-1  - persistenceType RULE  - rule http.req.
     header("User-Agent").value(0).contains("MyBrowser")  -
     persistenceBackup SOURCEIP
6
7 set lb vserver Vserver-LB-1 -persistenceType sslsession -
     persistenceBackup SourceIP
8 <!--NeedCopy-->
```

To set backup persistence for a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In **Advanced Settings**, select **Persistence**, and specify a backup persistence type.

Note: The primary persistence must be set to COOKIEINSERT, RULE, or SSLSESSION.

Configure persistence groups

September 14, 2021

When you have load-balanced servers that handle several different types of connections (such as Web servers that host multimedia), you can configure a virtual server group to handle these connections. To create a virtual server group, you bind different types of virtual servers, one for each type of connection that your load balanced servers accept, into a single group. You then configure a persistence type for the entire group.

You can configure either source IP-based persistence or HTTP cookie-based persistence for persistence groups. After you set persistence for the entire group, you cannot change it for individual virtual servers in the group. If you configure persistence on a group and then add a new virtual server to the group, the persistence of the new virtual server is changed to match the persistence setting of the group.

When persistence is configured on a group of virtual servers, persistence sessions are created for initial requests, and subsequent requests are directed to the same service as initial request, regardless of the virtual server in the group that receives each client request.

When you add a virtual server that has persistence sessions to a load balancing group with a different persistence type, the existing persistent sessions specific to an old persistence type are deleted.

The persistent sessions decide whether the traffic must go to the same virtual server or to a different server. Therefore, existing established connections are not impacted.

The persistence type of a load balancing group is applied to all the virtual servers bound to that group, irrespective of the virtual servers' protocol type. A load balancing group supports the following persistence types:

- SourceIP
- CookieInsert
- Rule

Some virtual servers support only certain persistence types. For example, a virtual server of type SSL_BRIDGE can use only SourceIP persistence type for an LB group.

If you configure HTTP cookie-based persistence, the domain attribute of the HTTP cookie is set. This setting causes the client software to add the HTTP cookie into client requests if different virtual servers have different public host names. For more information about CookieInsert persistence type, see Persistence Based on HTTP Cookies.

To create a virtual server persistency group by using the command line interface

At the command prompt, type:

```
bind lb group <vServerGroupName> <vServerName> -persistenceType <
    PersistenceType>
<!--NeedCopy-->
```

Example:

```
bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType
    CookieInsert
<!--NeedCopy-->
```

To modify a virtual server group by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Persistency Groups**, create a persistency group, and specify the virtual servers that must be part of this group.

To modify a virtual server group by using the command line interface

At the command prompt, type:

```
set lb group <vServerGroupName> -PersistenceBackup <
    BackupPersistenceType> -persistMask <SubnetMaskAddress>
```



```
<!--NeedCopy-->
```

Example:

```
set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask
   255.255.255.255
<!--NeedCopy-->
```

Share persistent sessions between virtual servers

September 14, 2021

In some customer environments (telecom and ISP), a single server handles both control and data traffic. For a given client IP address, both control and data traffic have to be directed to the same back-end server. For this, one virtual server is required for handling client authentication traffic, and usually rule based persistency is configured on it. For example, Radius.req.avp(8).value.typecast_text_t'. The second virtual server for handling data traffic. Usually, SourceIP persistence is configured on it.

Previously, persistence entries were local to the virtual server. If you had to apply persistence across multiple virtual servers, you had to add the virtual server to a load balancing group and then apply a common persistence type to the group. This requirement cannot be achieved, because all the virtual servers bound to a load balancing group inherited the persistency configured on the group.

With the persistency sharing between virtual server feature, you can set the new `useVserverPersistency` parameter for a load balancing group to allow the virtual server in the group to use their own persistency parameters instead of inheriting them from the group settings. You can configure separate rule-based persistency on each virtual server.

Optionally, you can also designate one of the virtual servers in the group as a main virtual server. When a virtual server is designated as a main virtual server, only that virtual server creates the persistence entries, which are used by all the virtual server in the group. If the main virtual server is down, the Citrix ADC appliance does not create any persistence entries.

Note: Persistence sharing across the virtual servers is supported only for rule based persistency methods. Configure compatible rule based persistence parameters on the member virtual servers.

Example:

Assume v1 and v2 are bound to a load balancing group, v1 is a RADIUS type virtual server and v2 is an HTTP type virtual server. 'Radius.req.avp(8).value.typecast_text_t' persistency is configured on v1 and 'client.ip.src' is configured on v2.

When traffic flows through the RADIUS virtual server v1, it creates a persistent entry based on the evaluated rule string. Later, when traffic reaches the HTTP type virtual server v2, v2 checks for the

persistence entries on the load balancing group and uses the same persistence session to direct traffic to the same back-end server.

Configuring Sharing of Persistent Sessions

To share persistency parameters across the virtual server in a load balancing group, you must first enable the useVserverPersistency parameter and then designate one of the virtual servers in the group as a main server.

To enable the useVserverPersistency parameter by using the command line interface

At the command prompt, type:

```
set lb group <name> -useVserverPersistency ( ENABLED)
<!--NeedCopy-->
```

Example:

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

To enable the useVserverPersistency parameter by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Persistency Groups**.
2. Click **Add** to add a new group or select an existing group and click **Edit**.
3. Select **Use Vserver Persistence**.

To designate a virtual server as a main virtual server by using the command line interface

At the command prompt, type:

```
set lb group <name> -useVserverPersistency ( ENABLED ) -masterVserver <
    string>
<!--NeedCopy-->
```

Example:

```
set lb group lb_grp1 -useVserverPersistency ENABLED - masterVserver vs1
<!--NeedCopy-->
```


To designate a virtual server as a main virtual server by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Persistency Groups**.
2. Click **Add** to add a new group or select an existing group and click **Edit**.
3. Select **Use Vserver Persistence**.
4. In the **Virtual Server Name** box, click **+** to add the virtual server to the group. You can select the available virtual server or create a virtual server.
5. Click **Create** if you are adding a new group or **Close** if you are modifying an existing group.
6. Select the group for which you have enabled the useVserverPersistency parameter and click **Edit** to set a virtual server as a main to create persistence entries.
7. From the **Master vServer** list, select the virtual server that has to be designated as a main virtual server.

Arguments

useVserverPersistency

Allow the virtual servers in a group to use their own persistency parameters to create persistent sessions, instead of inheriting the persistency settings from the group settings. When this parameter is enabled, persistence cannot be set on the load balancing group.

When this parameter is disabled, the group's virtual servers inherit the persistency parameters from the group settings.

When this parameter is toggled on the load balancing group, the Citrix ADC appliance flushes all the corresponding persistence entries of the group and the member virtual servers.

Possible values: ENABLED, DISABLED

Default: DISABLED

Example:

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

masterVserver

Designate a virtual server as a main virtual server in its load balancing group. Once designated, only the main virtual server can create the persistent entries used by the group.

Note: This parameter can be set only if the useVserverPersistency parameter is enabled.

Example:


```
set lb group lb_grp1 -masterVserver vs1
<!--NeedCopy-->
```

Example Configuration of Sharing Persistent Sessions by Using the Command Line Interface

The virtual servers are created

```
add lb vs vs1 http 10.1.10.11 80 -persistence rule -rule 'client.ip.src'

add lb vs vs2 radius 10.2.2.2 1812 -persistenceType rule -rule 'Radius.req.avp(8).value.typecast_text_t'
<!--NeedCopy-->
```

The groups are created.

```
add lb group lb_grp1 -persistenceType NONE -useVserverPersistency ENABLED
<!--NeedCopy-->
```

A virtual server in a group is designated as the main virtual server.

```
set lb group lb_grp1 -masterVserver vs1
<!--NeedCopy-->
```

The virtual servers are bound to the group.

```
bind lb group lb_grp1 vs1
bind lb group lb_grp1 vs2
<!--NeedCopy-->
```

For more details, see Setting Up Basic Load Balancing and Configuring Persistence Groups.

Configure RADIUS load balancing with persistence

September 14, 2021

Today's complex networking environment often requires coordinating a high-volume, high-capacity load balancing configuration with robust authentication and authorization. Application users might connect to a VPN through mobile access points such as consumer-grade DSL or Cable connections,

WiFi, or even dial-up nodes. Those connections usually use dynamic IPs, which can change during the connection.

If you configure RADIUS load balancing on the Citrix ADC appliance to support persistent client connections to RADIUS authentication servers, the appliance uses the user logon or the specified RADIUS attribute instead of the client IP as the session ID, directing all connections and records associated with that user session to the same RADIUS server. Users are therefore able to log on to your VPN from mobile access locations without experiencing disconnections when the client IP or WiFi access point changes.

To configure RADIUS load balancing with persistence, you must first configure RADIUS authentication for your VPN. For information and instructions, see the Authentication, Authorization, Auditing (AAA) chapter in AAA Application Traffic. Also choose either the Load Balancing or Content Switching feature as the basis for your configuration, and make sure that the feature you chose is enabled. The configuration process with either feature is almost the same.

Then, you configure either two load balancing, or two content switching, virtual servers, one to handle RADIUS authentication traffic and the other to handle RADIUS accounting traffic. Next, you configure two services, one for each load balancing virtual server, and bind each load balancing virtual server to its service. Finally, you create a load balancing persistency group and set the persistency type to RULE.

Enabling the Load Balancing or Content Switching Feature

To use the Load Balancing or Content Switching feature, you must first ensure that the feature is enabled. If you are configuring a new Citrix ADC appliance that has not previously been configured, both of these features are already enabled, so you can skip to the next section. If you are configuring a Citrix ADC appliance with a previous configuration on it, and you are not certain that the feature you use is enabled, you must do that now.

- For instructions on enabling the load balancing feature, see Enabling Load Balancing.
- For instructions on enabling the content switching feature, see Enabling Content Switching

Configuring Virtual Servers

After enabling the load balancing or content switching feature, you must next configure two virtual servers to support RADIUS authentication:

- **RADIUS authentication virtual server.** This virtual server and its associated service handles authentication traffic to your RADIUS server. Authentication traffic consists of connections associated with users logging onto your protected application or virtual private network (VPN).
- **RADIUS accounting virtual server.** This virtual server and its associated service handles accounting connections to your RADIUS server. Accounting traffic consists of connections that

track an authenticated user's activities on your protected application or VPN.

Important: You must create either a pair of load balancing virtual servers or a pair of content switching virtual servers to use in your RADIUS persistence configuration. You cannot mix virtual server types.

To configure a load balancing virtual server by using the command line interface

At the command prompt type the following commands to create a load balancing virtual server and verify the configuration:

```
1 add lb vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
     <rule>
2
3 show lb vserver <name>
4 <!--NeedCopy-->
```

To configure an existing load balancing virtual server, replace the preceding `add lb virtual server` command with the `set lb vserver` command, which takes the same arguments.

To configure a content switching virtual server by using the command line interface

At the command prompt type the following commands to create a content switching virtual server and verify the configuration:

```
1 add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
     <rule>
2
3 show cs vserver <name>
4 <!--NeedCopy-->
```

To configure an existing content switching virtual server, replace the preceding `add cs vserver` command with the `set cs vserver` command, which takes the same arguments.

Example:

```
1 add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
     TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
2
3 add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
     TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
4
5 set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
     TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
6
```



```
set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813 -lbmethod
   TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
<!--NeedCopy-->
```

To configure a load balancing or content switching virtual server by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Virtual Servers** or navigate to **Traffic Management > Content Switching > Virtual Servers»**, and configure a virtual server.

Configuring Services

After configuring your virtual servers, you must next configure two services, one for each of the virtual servers that you created.

Note: Once configured, these services are in the DISABLED state until the Citrix ADC appliance can connect to your RADIUS server's authentication and accounting IPs and monitor their status. For instructions, see Configuring Services.

Binding Virtual Servers to Services

After configuring your services, you must next bind each of the virtual servers that you created to the appropriate service.For instructions, see Binding Services to the Virtual Server.

Configuring a Persistency Group for Radius

After binding your load balancing virtual servers to the corresponding services, you must set up your RADIUS load balancing configuration to support persistence. To do so, you configure a load balancing persistency group that contains your RADIUS load balancing virtual servers and services, and configure that load balancing persistency group to use rule-based persistence. A persistency group is required because the authentication and accounting virtual servers are different and both the authentication & accounting message for a single user should reach the same RADIUS server. Persistency group enables to use the same session for both virtual servers. For instructions, see Configuring Persistence Groups.

Configuring RADIUS Shared Secret

From release 12.0, a Citrix ADC appliance supports RADIUS shared secret. A RADIUS client and server communicate with each other by using a shared secret that is configured on the client and on the server. Transactions between a RADIUS client and server are authenticated by using a shared secret. This secret is also used to encrypt some of the information in the RADIUS packet.

RADIUS shared secret key validation scenarios

The validation of the **RADIUS shared secret** key happens in the following scenarios:

- **RADIUS shared secret key is configured for both the radius client and the radius server:** The Citrix ADC appliance uses the RADIUS secret key for both the client side and the server side. If the verification succeeds, the appliance allows the RADIUS message to go through. Otherwise, it drops the RADIUS message.
- **RADIUS shared secret key is not configured for either the radius client or the radius server:** The Citrix ADC appliance drops the RADIUS message, because shared-secret-key validation cannot be performed on a node that has no radkey configured.
- **RADIUS shared secret key is not configured for both the RADIUS client and the RADIUS server:** The Citrix ADC appliance bypasses the RADIUS secret key validation and allows the RADIUS messages to go through.

You can configure a default RADIUS shared secret or configure on a per client or a subnet basis. It is recommended to add a RADIUS shared secret key for all deployments with RADIUS policy configured. The appliance uses the source IP address of the RADIUS packet to decide which shared secret to use. You might configure a RADIUS client and server and the corresponding shared secret as follows:

At the CLI prompt, type:

```
add radiusNode <clientPrefix/Subnet> -radKey <Shared_secret_key>
<!--NeedCopy-->
```

Arguments

IPaddress

IP address or subnet of the RADIUS client in CIDR format. The appliance uses the source IP address of an incoming request packet to match the client IP address. Instead of configuring a client IP address, you can configure the client network address. Longest prefix is matched to identify the shared secret for an incoming client request.

Radkey

Shared secret between the client, Citrix ADC appliance, and the server. Maximum length: 31.

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

```



```
add service radius_auth_service1 192.168.41.68 RADIUS 1812

add service radius_acct_service1 192.168.41.70 RADIUS 1813

bind lb vserver radius_auth_vs1 radius_auth_service1

bind lb vserver radius_acct_vs1 radius_acct_service[1-3]

add radiusNode 192.168.41.0/24 -radKey  serverkey123

add radiusNode 203.0.113.0/24 -radkey clientkey123
<!--NeedCopy-->
```

A shared secret must be configured both for a RADIUS client and server. The command is the same. The subnet determines whether the shared secret is for a client or for a server.

For example, if the subnet specified is a client subnet, the shared secret is for the client. If the subnet specified is a server subnet (192.168.41.0/24 in the earlier example), the shared secret is for the server.

A subnet of 0.0.0.0/0 implies that it is the default shared secret for all clients and servers.

> **Note:**
>
> Only the PAP and CHAP authentication methods are supported with RADIUS shared secret.

View persistence sessions

September 14, 2021

You can view the different persistence sessions that are in effect globally or for a particular virtual server.

> **Note**: A Citrix ADC nCore appliance uses multiple CPU cores for packet handling. The CPU core owns every session on the appliance. If the appliance receives a request for which a session does not exist, a session is created, and one of the cores is designated as the owner of that session.
>
> Subsequent requests that belong to that session might not always arrive at and be handled by the owner core. In that case, inter-core messaging ensures that the session information on the owner core is always current.

However, when a core receives a request that belongs to a persistence session owned by another core, the inter-core messaging does not refresh the timeout value for the persistence session.

Therefore, in the output of successively run show lb persistentSessions commands, which display timeout values from owner cores only, the timeout value for a persistence session might diminish to

0 (zero), even if the persistence session remains active.

To view persistence sessions by using the command line interface

At the command prompt, to view persistence sessions related to all virtual servers, type:

```
show lb persistentSessions [<vServer>]
<!--NeedCopy-->
```

At the command prompt, to view persistence sessions related a virtual server, type:

```
show lb persistentSessions <vServername>
<!--NeedCopy-->
```

Example:

```
show lb persistentSessions myVserver
<!--NeedCopy-->
```

To view persistence sessions by using the GUI

Navigate to **Traffic Management** > **Virtual server persistent sessions**.

Clear persistence sessions

September 14, 2021

You might need to clear persistence sessions from the Citrix ADC appliance if sessions fail to time out. You can do one of the following:

- Clear all sessions for all virtual servers at once.
- Clear all sessions for a given virtual server at once.
- Clear a particular session that is associated with a given virtual server.

To clear a persistence session by using the command line interface

At the command prompt, type the following commands to clear persistence sessions and verify the configuration:

```
clear lb persistentSessions [<vServer> [-persistenceParam <string>]]

show persistentSessions <vServer>
```



```
<!--NeedCopy-->
```

Examples:

Example 1 clears all persistence sessions for load balancing virtual server lbvip1.

Example 2 first displays the persistence sessions for load balancing virtual server lbvip1, clears the session with the persistence parameter xls, and then displays the persistence sessions to verify that the session was cleared.

Example 1:

```
> clear persistentSessions lbvip1
 Done
> show persistentSessions
 Done
>
<!--NeedCopy-->
```

Example 2:

```
> show persistentSessions lbvip1
Type        SRC-IP    ...    PERSISTENCE-PARAMETER
RULE        0.0.0.0   ...    xls
RULE        0.0.0.0   ...    txt
RULE        0.0.0.0   ...    html
 Done
> clear persistentSessions lbvip1 -persistenceParam xls
 Done
> show persistentSessions lbvip1
Type        SRC-IP    ...    PERSISTENCE-PARAMETER
RULE        0.0.0.0   ...    txt
RULE        0.0.0.0   ...    html
 Done
>
<!--NeedCopy-->
```

To clear persistence sessions by using the configuration utility

1. Navigate to **Traffic Management > Clear Persistent Sessions**.

Override persistence settings for overloaded services

September 14, 2021

When a service is loaded or is otherwise unavailable, service to clients is degraded. In this case, you might have to configure the Citrix ADC appliance to temporarily forward to other services the requests that would otherwise be included in the persistence session that is associated with the overloaded service. In other words, you might have to override the persistence setting that is configured for the load balancing virtual server. You can achieve this functionality by setting the skippersistency parameter. When this skippersistence parameter is set, and if the virtual server receives new connections for an overloaded service, the following happens.

- The virtual server disregards any existing persistence sessions that are associated with that service, until the service returns to a state at which it can accept requests.
- Persistence sessions associated with other services are not affected.

This functionality is available for only virtual servers whose type is ANY or UDP.

In Branch Repeater load balancing configurations, you must also configure a load monitor and bind it to the service. The monitor takes the service out of subsequent load balancing decisions until the load on the service is brought below the configured threshold. For information about configuring a load monitor for your virtual server, see Understanding Load Monitors.

You can configure the virtual server to perform one of the following actions with the requests that would otherwise form a part of the persistence session:

- **Send each request to one of the other services.** The virtual server takes a load balancing decision and sends each request to one of the other services based on the load balancing method. If all the services are overloaded, requests are dropped until a service becomes available.

  Both wildcard and IP address–based virtual servers support this option. This action is appropriate for all deployments, including deployments in which the virtual server is load balancing Branch Repeater appliances or firewalls.

- **Bypass the virtual server-service configuration.** The virtual server does not take a load balancing decision. Instead, it simply bridges each request through to a physical server based on the destination IP address in the request.

  Only wildcard virtual servers of type ANY and UDP support the bypass option. Wildcard virtual servers have a *:* IP and port combination. This action is appropriate for deployments in which you are using the virtual server to load balance Branch Repeater appliances or firewalls. In these deployments, the Citrix ADC appliance first forwards a request to a Branch Repeater appliance or firewall, and then forwards the processed response to a physical server. The virtual server sends requests directly to their destination IP addresses in the following conditions.

- You configure the virtual server to bypass the virtual server–service configuration for overloaded services.
- The Branch Repeater appliance or firewall gets overloaded.

The virtual server sends requests directly to their destination IP addresses until the Branch Repeater appliance or firewall can accept requests.

To override persistence settings for overloaded services by using the CLI

At the command prompt, type the following commands to override persistence settings for overloaded services and verify the configuration:

```
set lb vserver <name> -skippersistency <skippersistency>

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
> set lb vserver mylbvserver -skippersistency ReLb
 Done
> show lb vserver mylbvserver
    mylbvserver (*:*) - ANY Type: ADDRESS
       . . .
       . . .
 Skip Persistency: ReLb
       . . .
 Done
>
<!--NeedCopy-->
```

To override persistence settings for overloaded services by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers** and select the virtual server of type UDP or ANY.
2. In the Advanced Settings pane, select Traffic Settings, and specify the type of Skip Persistency.

Troubleshooting

September 14, 2021

- **The statistics from the Citrix ADC VPX appliance indicate that the appliance has reached the session persistence limit. As a result, persistence sessions are failing. Is possible to increase the session persistence limit?**

  Cause: The Citrix ADC appliance has the system limit of 250,000 persistence session for a core.

  Resolution: To resolve this issue, you can perform any of the following tasks:

  - Reduce the time-out value for persistence
  - Increase the number of cores for the appliance

- **After configuring Cookie Insert persistence on the Citrix ADC appliance, the users report that the connections work fine for some time, but then start getting disconnected. What best practice should I follow when configuring persistence?**

  Cause: By default, the time-out value for Cookie Insert persistence is 120 seconds.

  Resolution: When you configure persistence for applications for which idle time cannot be determined, set the Cookie Insert persistence time-out value to 0. With this setting, the connection does not time out.

- **After configuring an HTTP virtual server on the Citrix ADC appliance, I need to make sure that a user always connects to the same server for the requested content, so I configured SourceIP persistence. Now, increasing the time-out value for persistence introduces latency. How can I increase the timeout value without affecting performance?**

  Resolution: Consider using Cookie Insert persistence with the time-out value set to 0. This setting enables long-duration persistence settings, because the appliance does not specify a time for expiring the cookie.

- **After configuring Cookie Insert persistence on the Citrix ADC appliance, it works as expected when clients from the same time zone access the content. However, when a client from another time zone makes an attempt to connect, the connection is immediately timed out.**

  Cause: Time based Cookie Insert persistence works as expected when a client from the same time zone makes a connection. However, when the client machine and Citrix ADC appliance are in different time zones, the cookie is not valid. For example, when a client in the EST time zone sends a cookie at 11:00 AM EST to a Citrix ADC appliance in the PST time zone, the appliance receives the cookie at 2:00 PM PST. As a result of the difference in time, the cookie is not valid, and the connection is immediately timed-out.

  Resolution: Set the time-out value for Cookie Insert persistence to 0.

- **A Citrix ADC appliance is used to load balance application servers, such as Oracle Weblogic server. To make sure that clients get persistent connections to these servers, SourceIP persistence is configured. It works as expected when a connection is made from a computer. However, when thin clients attempt a connection through a terminal server and,**

as a result, the appliance receives requests from multiple clients from the same IP address (the terminal server IP address). Therefore, the connections from all thin clients are directed to the same application server. Is it possible to configure persistency for requests from individual thin clients based on the client IP address?

Cause: The Citrix ADC appliance receives requests from the terminal server and the source IP address of the request remains the same. As a result, the appliance cannot distinguish among the requests received from the thin clients and provide persistence according to the requests from thin clients.

Resolution: To avoid this problem, you can configure Rule persistence based on some unique parameter value for each thin client.

- **The Citrix ADC appliance is used to load balance Web Interface servers. When accessing the servers, the user receives the "State Error" error message. Additionally, when one of the Web Interface servers is shut down or not available, some of the users receive an error message.**

  Cause: Lack of persistence to the Web Interface servers can result in error messages when a user attempts to connect to the server.

  Resolution: Citrix recommends that you specify the Cookie Insert persistence method on the Citrix ADC appliance when load balancing Web Interface servers.

Insert cookie attributes to ADC generated cookies

September 14, 2021

The web administrators can insert other cookie attributes to the cookies generated by the Citrix ADC appliance. These additional cookie attributes help in enforcing the required policies for the ADC generated cookies based on the application access pattern.

The following features use the ADC generated cookies to achieve persistency.

- Load balancing cookie persistence
- Load balancing group cookie persistence
- GSLB site persistence
- Content switching cookie persistence

You can insert other cookie attributes to the ADC generated cookies using the following parameters:

- **LiteralADCCookieAttribute:** Append other cookie attributes to the ADC generated cookie, as a string.

- **ComputedADCCookieAttribute:** Use an ADC ns variable to conditionally append cookie attributes to the ADC generated cookie, based on the client or server attributes, for example, user agent version.

> **Note**
>
> You cannot configure both Literal ADC Cookie Attribute and Computed ADC Cookie Attribute, simultaneously on the load balancing parameter or in a single load balancing profile.

Use case: Configure SameSite cookie attribute

Every cookie has a domain associated with it. When a cookie's domain matches the website domain in the user's address bar, this is considered a same-site (or first party) context. If the domain associated with a cookie matches an external service and not the website in the user's address bar, this is considered a cross-site (or third party) context.

The **SameSite** attribute indicates the browser whether the cookie can be used for cross-site context or only for same-site context. Also, if an application intends to be accessed in the cross-site context then it can do so only via the HTTPS connection. For details, see RFC6265.

Until Feb 2020, the **SameSite** property was not explicitly set in Citrix ADC. The browser took the default value as None, and did not impact the Citrix ADC deployments.

However with certain browsers upgrade, such as Google Chrome 80, there is a change in the default cross-domain behavior of cookies. The **SameSite** attribute can be set to one of the following values. Default value for Google Chrome is set to Lax.

- **None:** Indicates the browser to use a cookie in cross-site context only on secure connections.
- **Lax:** Indicates the browser to use a cookie for requests in the same-site context. In the cross-site context, only safe HTTP methods like GET request can use the cookie.
- **Strict:** Use the cookie only in the same-site context.

If there is no SameSite attribute in the cookie, the Google Chrome assumes the functionality of SameSite=Lax.

> **Note**
>
> For certain version of other browsers, the default value for the SameSite attribute might be set to **None**. In some browser versions, "SameSite = none" can be treated differently. For example, the following browsers reject a cookie with "SameSite = none":
>
> - Versions of Chrome from Chrome 51 to Chrome 66 (inclusive on both ends)
> - Versions of UC browser on Android earlier to version 12.13.2

Configure ADC generated cookies

To configure ADC generated cookie attributes, you must perform the following:

1. Create a load balancing virtual server
2. Set the ADC Cookie attributes for the load balancing virtual server, either through LB parameters or LB profile.
3. If you use an LB profile, set the LB profile to a load balancing virtual server.
4. If you choose to use the Computed ADC Cookie Attribute, configure the related rewrite policy.

> **Note**
>
> If an LB profile is bound to an LB virtual server, then the profile parameter configuration is considered instead of the global LB parameter configuration.

You can set the ADC generated cookie attributes by the following methods:

- Setting the ADC cookie attributes in load balancing parameters
- Setting the ADC cookie attributes in the load balancing profile

Setting the ADC cookie attributes in the load balancing parameters by using the CLI

To apply a policy uniformly to ADC generated cookies of all applications configured on the Citrix ADC appliance, you can set the ADC cookie attribute in the global LB parameters.

The **Literal ADC Cookie Attribute** setting allows you to unconditionally insert the cookie attributes to the ADC generated cookie.

At the command prompt, type:

```
set lb parameter -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

Example:

```
set lb parameter -LiteralADCCookieAttribute SameSite=None
<!--NeedCopy-->
```

The **Computed ADC Cookie Attribute** setting allows you to conditionally insert the cookie attributes, based on the client or server attributes, to the ADC generated cookie.

At the command prompt, type:

```
set lb parameter -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

Example:

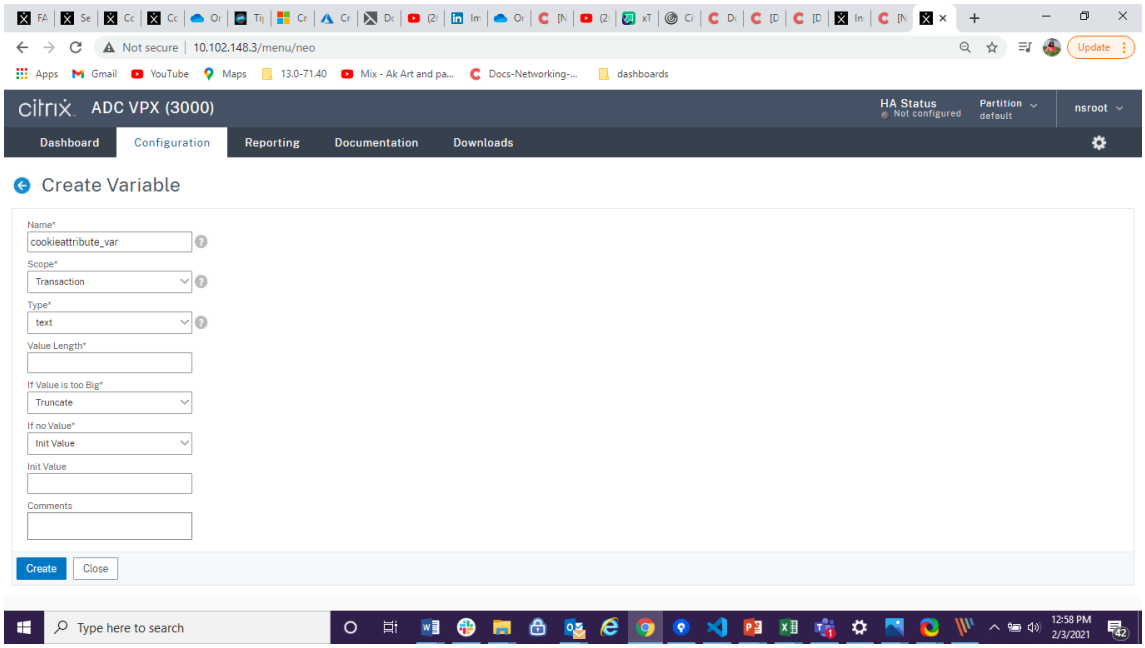
```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
```



```
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    "\"SameSite=None\""

add policy expression pol_iphone "(HTTP.REQ.HEADER(\"User-Agent\").
    CONTAINS(\"iP\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re
    /OS \\d+\\_/).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT \"false\").eq(\"true\"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER(\"User-Agent\").
    CONTAINS(\"Chrom\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT
    (re/Chrom.*\\d+./).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).
    BETWEEN(51,66).typecast_text_t ALT \"false\").eq(\"true\"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

bind rewrite global exception_samesite_attribute 90 110 -type
    RES_OVERRIDE
bind rewrite global append_samesite_attribute 100 110 -type
    RES_OVERRIDE
<!--NeedCopy-->
```

Configure variables by using the GUI

1. Navigate to **AppExpert > Variables**, and click **Add**.
2. In the **Create Variable** page, select **Scope** as **Transaction** and **Type** as **text** from the drop-down menu.

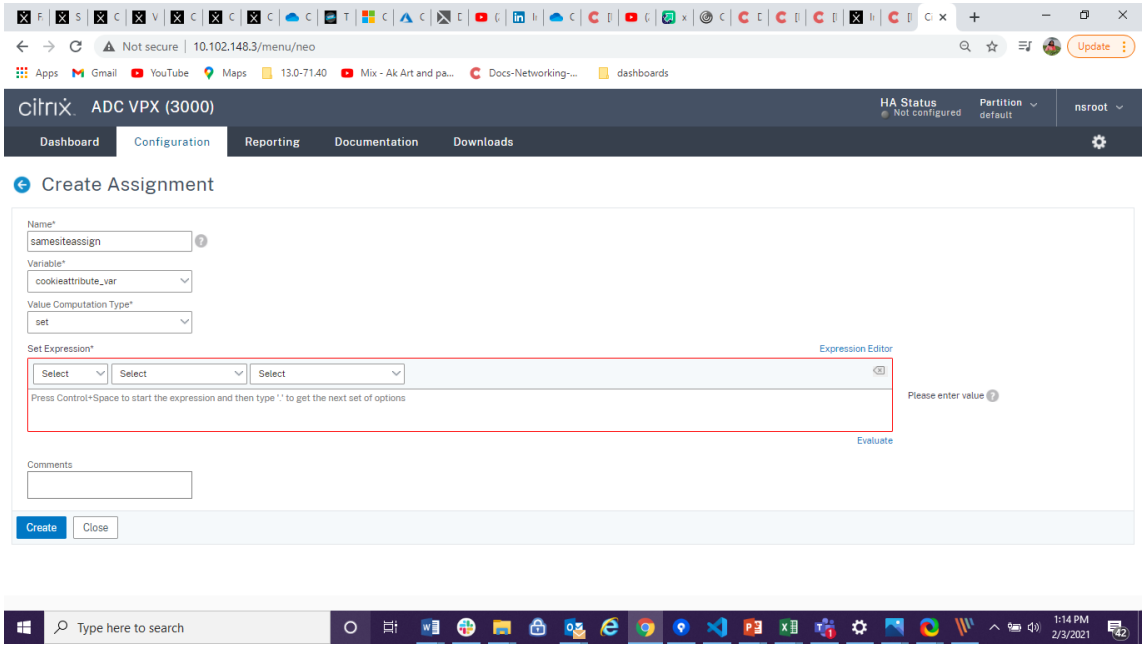


3. Enter other details, and click **Create**.

Create an assignment by using the GUI

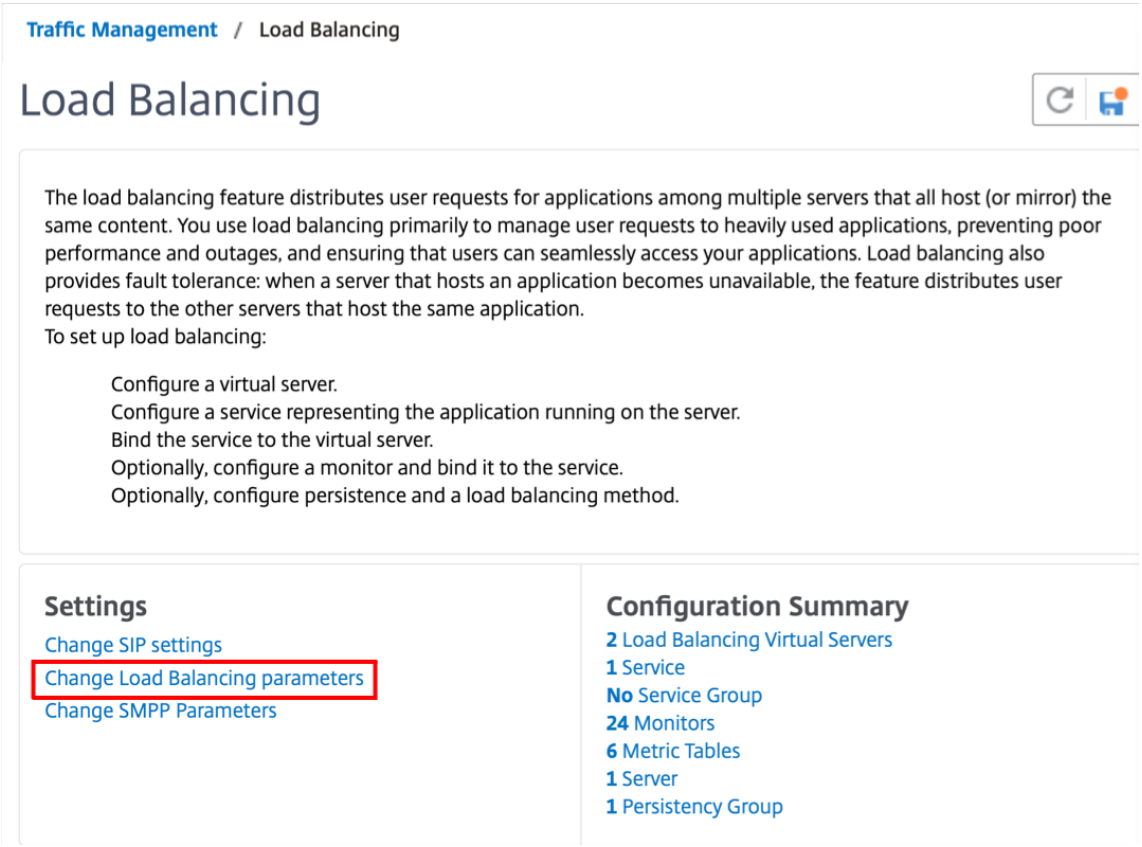
After configuring a variable, you can assign a value or specify the operation to be performed on the variable by creating an assignment.

1. Navigate to **AppExpert > Assignments**, and click **Add**.
2. In the **Create Assignment** page, enter the details, and click **Create**.



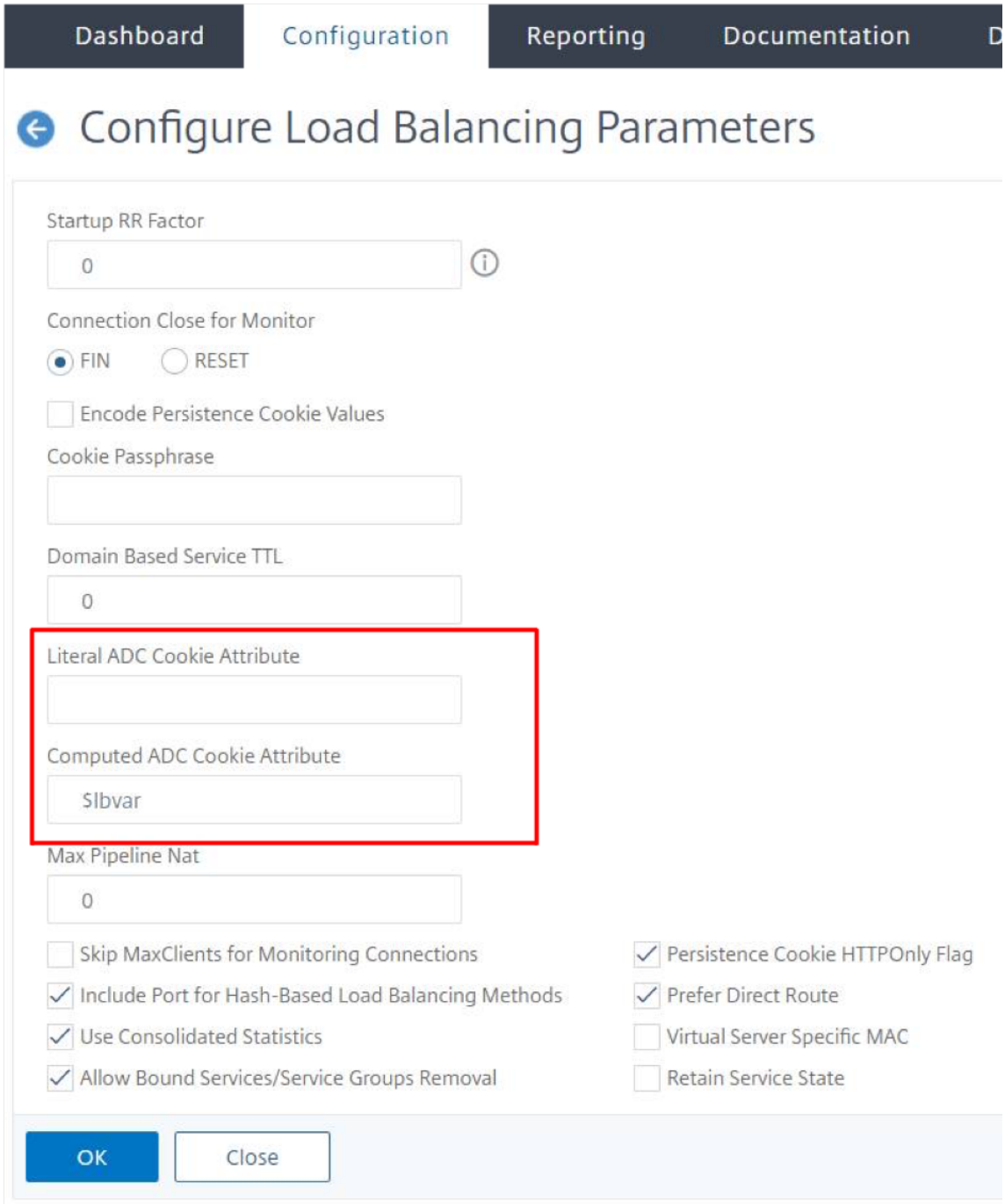
Setting the ADC cookie attributes in load balancing parameters by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Change Load Balancing parameters**.



2. In the **Configure Load Balancing Parameters** pane, enter appropriate values for either one of the fields based on your requirement:

   - **Literal ADC Cookie Attribute**
   - **Computed ADC Cookie Attribute**



3. Click **OK**.

Setting the ADC cookie attributes in the load balancing profile by using the CLI

To apply a policy for a specific application that is configured on the Citrix ADC appliance, you can set the cookie attribute parameters in the LB profile bound to the application-specific LB virtual server.

The **Literal ADC Cookie Attribute** setting in the LB profile allows you to unconditionally insert the

cookie attributes to the ADC generated cookie that is specific to a virtual server.

At the command prompt, type:

```
add lb profile <profile name> -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

Example:

```
add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
<!--NeedCopy-->
```

The **Computed ADC Cookie Attribute** setting in the LB profile allows you to conditionally insert the cookie attributes based on the client or server attributes, to the ADC generated cookie. Then, set this LB profile to an LB virtual server.

At the command prompt, type:

```
add lb profile <profile name> -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

Example:

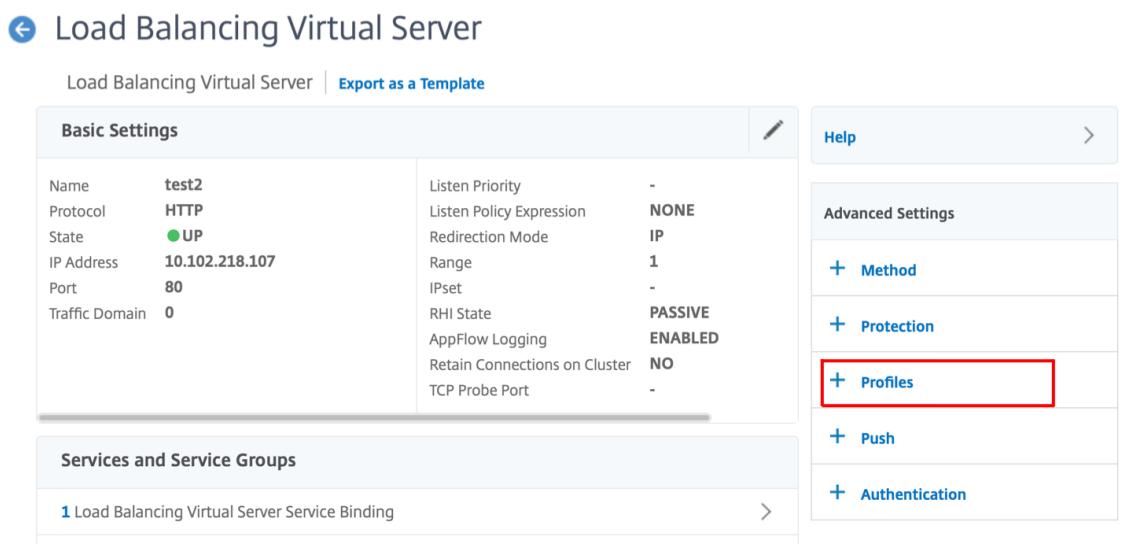
```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    "\"SameSite=None\""
add lb profile LB-Vserver-Profile-1 -ComputedADCCookieAttributE "
    $cookieattribute_var"

add policy expression pol_iphone "(HTTP.REQ.HEADER(\"User-Agent\").
    CONTAINS(\"iP\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re
    /OS \\d+\\_/).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT \"false\").eq(\"true\"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER(\"User-Agent\").
    CONTAINS(\"Chrom\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT
    (re/Chrom.*\\d+./).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).
    BETWEEN(51,66).typecast_text_t ALT \"false\").eq(\"true\"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
```



```
bind lb vserver LB-VServer-1 -policyName exception_samesite_attribute -
    priority 90 -gotoPriorityExpression 110 -type RESPONSE
bind lb vserver LB-VServer-1 -policyName append_samesite_attribute -
    priority 100 -gotoPriorityExpression 110 -type RESPONSE
<!--NeedCopy-->
```

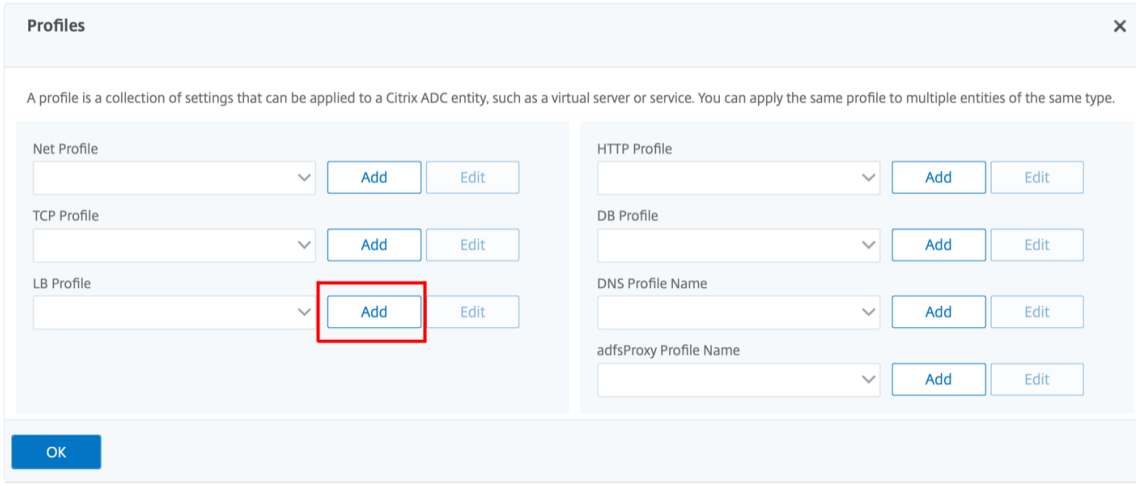
Setting the ADC Cookie attributes in the load balancing profile by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual servers**.
2. Select a virtual server and click **Edit**.
3. Under **Advanced Settings** section, click **Add profiles**.



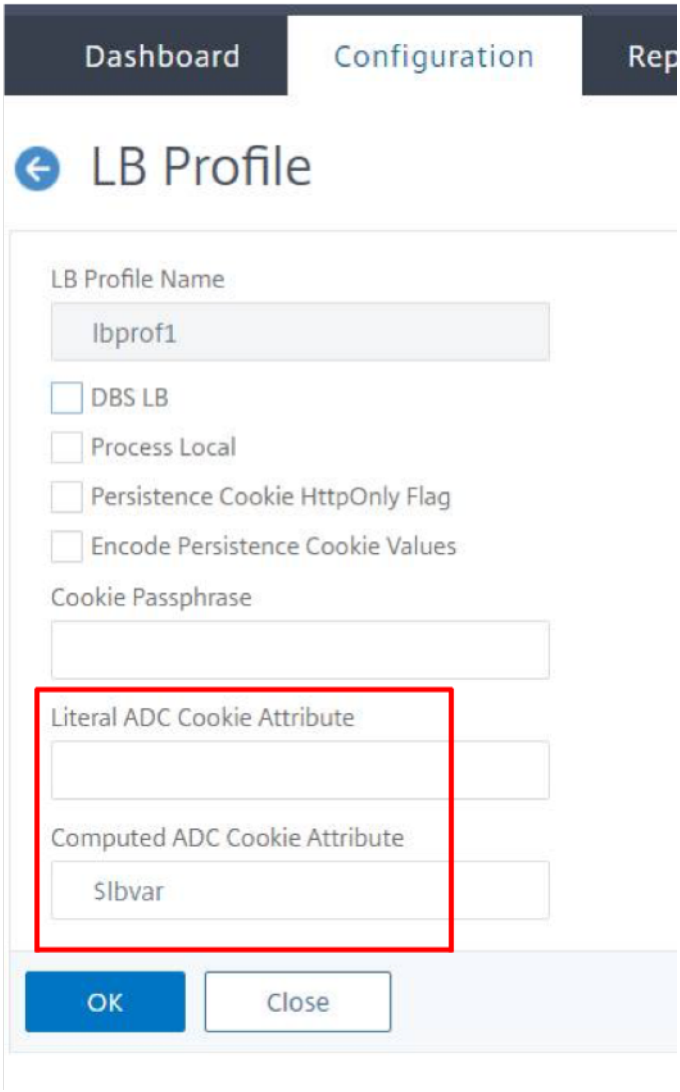
4. In the **Profiles** section, click **Add** to create an LB Profile.

   If you have already created a profile, choose it from the **LB Profile** drop-down menu.



5. In the **LB Profile** pane, enter appropriate values for either one of the fields based on your requirement:

- **Literal ADC Cookie Attribute**
- **Computed ADC Cookie Attribute**



1. Click **OK**.
2. Set the created LB profile to the LB virtual server created in **Step 1**.

Verify ns variable configuration

To verify that the ADC ns variable is configured appropriately in LB parameters or LB profile, use the show lb parameter or show lb profile commands.

The following table lists the various warning messages and its cause, when the ns variable is not cor-

rectly configured.

Warning message	Reasons
NS Variable is not configured. Configure it with type text() and scope transaction for variable	NS variable is not yet configured.
Scope of configured NS variable is not transaction.	Variable is configured but scope is not set to “transaction.”
Type of variable is not Text().	Variable is configured but type is not set to “Text”.
Configured value-max-size for the NS Variable is greater than 255.	The value configured for the NS variable is greater than 255 characters. **Note:** A maximum length of 255 characters can be appended to an ADC generated cookie. The characters that exceed the maximum length are truncated.

Sample output

In the following example, the warning message is displayed when the ns variable is not configured.

```
set lb parameter -ComputedADCCookieAttribute "$lbvar"

Warning: NS Variable is not configured. Please configure it with type
    text() and scope transaction
Done
<!--NeedCopy-->
```

The warning message is displayed in the following output of the `show lb parameter` command.

```
show lb parameter

Global LB parameters:
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
Use Port For Hash LB: YES
Prefer direct route: YES
Retain Service State: OFF
Start RR Factor: 0
Skip Maxclient for Monitoring: DISABLED
Monitor Connection Close: FIN
Use consolidated stats for LeastConnection: YES
```



```
13 Allow mac mode based vserver to pick thereturn traffic from services:
      DISABLED
14 Allow bound service removal: ENABLED
15 TTL for Domain Based Server: 0 secs
16
17 Citrix ADC Cookie Variable Name: $lbvar(NS Variable is not configured.
      Please configure it with type text() and scope transaction)
18
19 Done
20 <!--NeedCopy-->
```

Sample configuration for inserting cookie attributes in GSLB deployment

The following sample configuration applies to site persistence configured on GSLB services corresponding to an LB virtual server. To append some additional cookie attributes to the GSLB cookies, perform the following configuration.

- Set the ADC cookie attributes in the LB profile (LB-Vserver-Profile-1).
- Set the Literal ADC Cookie Attribute value, for example “SameSite=None”, in the LB profile.
- Set the LB profile to the load balancing virtual server (LB-VServer-1), which represents the GSLB service.

```
1  add gslb vserver GSLB-VServer-1 SSL -backupLBMethod ROUNDROBIN -
      tolerance 0 -appflowLog DISABLED
2  add gslb site site1 10.102.148.4 -publicIP 10.102.148.4
3  add gslb service site1_gsvc1 10.102.148.35 SSL 443 -publicIP
      10.102.148.35 -publicPort 443 -maxClient 0 -siteName site1 -
      sitePersistence HTTPRedirect -sitePrefix ss1 -cltTimeout 180 -
      svrTimeout 360 -downStateFlush ENABLED
4
5  bind gslb vserver GSLB-VServer-1 -serviceName site1_gsvc1
6  bind gslb vserver GSLB-VServer-1 -domainName www.gslb.com -TTL 5
7
8  add service service-1 10.102.84.140 SSL 443
9
10 add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
      =None
11 add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
      COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
12
13 bind lb vserver LB-VServer-1 service-1
14 <!--NeedCopy-->
```


> **Note**
>
> You can also conditionally insert the cookie attributes using the Computed ADC Cookie Attribute.

Sample configuration for inserting cookie attribute in content switching deployment

The following sample configuration applies when multiple applications are hosted behind a content switching virtual server. To apply the same policy to all the applications, bind the rewrite policies to content switching virtual server instead of LB virtual server, as follows:

- Set the ADC cookie attributes in the LB parameters.

  > **Note:**
  >
  > You can set the ADC cookie attributes in the LB profile as well.

- Configure the ns variable (cookieattribute_var) of Type set to Text and Scope set to Transaction.
- Set the Computed ADC Cookie Attribute in the global LB parameters using the ns variable.
- Set the rewrite policies (exception_samesite_attribute and append_samesite_attribute) to the content switching virtual servers for inserting the cookie attributes.

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    "\"SameSite=None\""

add policy expression pol_iphone "(HTTP.REQ.HEADER(\"User-Agent\").
    CONTAINS(\"iP\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re
    /OS \\d+\\_/).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT \"false\").eq(\"true\"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER(\"User-Agent\").
    CONTAINS(\"Chrom\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT
    (re/Chrom.*\\d+./).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).
    BETWEEN(51,66).typecast_text_t ALT \"false\").eq(\"true\"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.35 443
add lb vserver LB-VServer-2 SSL 10.102.148.36 443

add cs vserver CS-VServer-1 SSL 10.102.148.42 443 -persistenceType
    COOKIEINSERT

```



```
15 add cs action act1 -targetLBVserver v1
16 add cs action act2 -targetLBVserver v2
17 add cs policy CS-policy-1 -rule "HTTP.REQ.URL.CONTAINS(\"file1.html\")"
       -action act1
18 add cs policy CS-policy-2 -rule "HTTP.REQ.URL.CONTAINS(\"file2.html\")"
       -action act2
19
20 bind cs vserver CS-VServer-1 -policyName CS-policy-1 -priority 1
21 bind cs vserver CS-VServer-1 -policyName CS-policy-2 -priority 2
22
23 bind cs vserver -policyname exception_samesite_attribute 90 110 -type
      RES_OVERRIDE
24 bind cs vserver -policyname append_samesite_attribute 100 110 -type
      RES_OVERRIDE
25 <!--NeedCopy-->
```

Customize a load balancing configuration

September 14, 2021

After you configure a basic load balancing setup, you can make several modifications to it so that it distributes the load exactly as you need. The load balancing feature is complex. You can modify the basic elements by doing one or more of the following:

- Changing the load balancing algorithm
- Configuring load balancing groups and using them to create your load balancing configuration
- Configuring persistent client-server connections
- Configuring the redirection mode
- Assigning different weights to different services that have different capacities.

The default load balancing algorithm on the Citrix ADC appliance is the least connection method. In the least connection method, the appliance sends each incoming connection to the service that is currently handling the fewest connections. You can specify different load balancing algorithms, each of which is suited to different conditions.

To accommodate applications such as shopping carts, which require that all requests from the same user be directed to the same server, you can configure the appliance to maintain persistent connections between clients and servers. You can also specify persistence for a group of virtual servers. Persistence lets the appliance to direct individual client requests to the same service regardless of which virtual server in the group receives the client request.

You can enable and configure the redirection mode that the appliance uses when redirecting user

requests, choosing between IP-based and MAC-based forwarding. You can also assign weights to different services, specifying what percentage of incoming load must be directed to each service. Assigning weights enable you to include servers with different capacities in the same load balancing setup without;

- overloading the lower-capacity servers or
- allowing the higher-capacity servers to sit idle.

Customize the hash algorithm for persistence across virtual servers

September 14, 2021

The Citrix ADC appliance uses hash-based algorithms for maintaining persistence across virtual servers. By default, the hash-based load balancing method uses a hash value of the IP address and port number of the service. If a service is made available at different ports on the same server, the algorithm generates different hash values. Therefore, different load balancing virtual servers might send requests for the same application to different services, breaking the pseudo-persistence.

As an alternative to using the port number to generate the hash value, you can specify a unique hash identifier for each service. For a service, the same hash identifier value must be specified on all the virtual servers. If a physical server serves more than one type of application, each application type should have a unique hash identifier.

The algorithm for computing the hash value for a service works as follows:

- By default, a global setting specifies the use of port number in a hash calculation.
- If you configure a hash identifier for a service, it is used, and the port number is not, regardless of the global setting.
- If you do not configure a hash identifier, but change the default value of the global setting so that it does not specify use of the port number, the hash value is based only on the IP address of the service.
- If you do not configure a hash identifier or change the default value of the global setting to use the port number, the hash value is based on the IP address and the port number of the service.

You can also specify hash identifiers when using the CLI to bind services to a service group. In the configuration utility, you can open a service group and add hash identifiers on the Members tab.

To change the use-port-number global setting by using the CLI

At the command prompt, type:

set lb parameter -usePortForHashLb (YES NO)

Example:

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
Global LB parameters:
        Persistence Cookie HttpOnly Flag: DISABLED
        Use port for hash LB: NO
 Done
<!--NeedCopy-->
```

To change the use-port-number global setting by using the GUI

1. Navigate to Traffic Managment > Load Balancing > Configure Load Balancing parameters.
2. Select or clear Use Port for Hash Based LB Methods.

To create a new service and specify a hash identifier for a service by using the CLI

At the command prompt, type the following commands to set the hash ID and verify the setting:

add service < name > (< ip > < serverName >) < serviceType > < port > -hashId < positive_integer >

```
show service <name>
<!--NeedCopy-->
```

Example:

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
        flbkng (10.101.10.1:80) - HTTP
        State: DOWN
        Last state change was at Thu Nov  4 10:14:52 2010
        Time since last state change: 0 days, 00:00:15.990
        Server Name: 10.101.10.1
        Server ID : 0   Monitor Threshold : 0

```



```
        Down state flush: ENABLED
        Hash Id: 12345

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1

 Done
<!--NeedCopy-->
```

To specify a hash identifier for an existing service by using the CLI

Type the set service command, the name of the service, and **-hashID** followed by the ID value.

To specify a hash identifier while adding a service group member

To specify a hash identifier for each member to be added to the group and verify the setting, at the command prompt, type the following commands (Be sure to specify a unique hashID for each member.):

```
bind servicegroup <serviceGroupName> <memberName> <port> -hashId <
   positive_integer>

show servicegroup <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
        SRV - HTTP
        State: ENABLED  Monitor Threshold : 0
        ...

        1)         1.1.1.1:80   State: DOWN     Server Name: 1.1.1.1
              Server ID: 123  Weight: 1
        Hash Id: 32211

                Monitor Name: tcp-default       State: DOWN
        ...

        2)         2.2.2.2:80   State: DOWN     Server Name: 2.2.2.2
              Server ID: 123  Weight: 1
```



```
        Hash Id: 12345

                Monitor Name: tcp-default       State: DOWN
        ...
Done

<!--NeedCopy-->
```

To specify a hash identifier for a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. Create a new service, or open an existing service and specify the hash ID.

To specify a hash identifier for an already configured service group member by using the GUI

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a member and type a unique hash ID.

Configure the redirection mode

September 30, 2021

The redirection mode configures the method used by a virtual server to determine where to forward incoming traffic. The Citrix ADC appliance supports the following redirection modes. Before forwarding the request to a server, the redirection modes function as follows:

- IP-Based forwarding (the default): The destination IP address is changed to the server's IP address.
- MAC-Based forwarding: The destination MAC address is changed to the server's MAC address. However, the destination IP address is not changed. MAC-based redirection mode is used mostly in firewall load balancing deployments.
- IP TUNNEL Based: An IP-in-IP encapsulation is performed for the client IP packets. In the outer IP headers, the destination IP address is set to the IP address of the server and the source IP address is set to the subnet IP (SNIP). The client IP packets are not modified. This is applicable to both IPv4 and IPv6 packets.
- TOS ID Based: The virtual server's TOS ID is encoded in the TOS field of the IP header.

You can use either the IP TUNNEL or the TOS option to implement Direct Server Return (DSR). For more information see:

- Configure DSR mode when using TOS
- Configure load balancing in DSR mode for IPv6 networks by using the TOS field
- Configure load balancing in DSR mode by using IP Over IP

You can configure MAC-Based forwarding on networks that use DSR topology, link load balancing, or firewall load balancing. For more information on MAC-Based forwarding for load balancing, see Configure MBF for load balancing configuration.

To configure the redirection mode by using the CLI

At the command prompt, type:

```
set lb vserver <name> -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which the `-m MAC` option is enabled, you must bind a non-user monitor.

To configure the redirection mode by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server and select the redirection mode.

Configure per-VLAN wildcarded virtual servers

September 14, 2021

If you want to configure load balancing for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the CLI

At the command prompt, type the following commands to configure a wildcarded virtual server that listens to a specific VLAN and verify the configuration:


```
add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <
    expression> [-listenpriority <positive_integer>]

show vserver
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)
    " -listenpriority 10

show vserver Vserver-LB-vlan1
<!--NeedCopy-->
```

To configure a wildcarded virtual server that listens to a specific VLAN by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Create a new virtual server or open an existing virtual server.
3. Specify a listen policy priority and expression.

After you have created this virtual server, you bind it to one or more services as described in Setting Up Basic Load Balancing.

Assign weights to services

September 14, 2021

In a load balancing configuration, you assign weights to services to indicate the percentage of traffic that should be sent to each service. Services with higher weights can handle more requests; services with lower weights can handle fewer requests. Assigning weights to services allows the Citrix ADC appliance to determine how much traffic each load balanced server can handle, and therefore more effectively balance load.

Note: If you use a load balancing method that supports weighting of services (for example, the round robin method), you can assign a weight to the service.

The following table describes the load balancing methods that support weighting, and briefly describes the manner in which weighting affects how a service is selected for each one.

Load Balancing Methods	Service Selection with Weights
Round Robin	The virtual server prioritizes the queue of available services such that services with the highest weights come to the front of the queue more frequently than those with the lowest weights and receive proportionately more traffic. For a complete description, see The Round Robin Method.
Least Connection	The virtual server selects the service with the best combination of fewest active transactions and highest weight. For a complete description, see The Least Connection Method.
Least Response Time and Least Response Time Method using Monitors	The virtual server selects the service with the best combination of fewest active transactions and fastest average response time. For a complete description, see The Least Response Time Method.
Least Bandwidth	The virtual server selects the service with the best combination of least traffic and highest bandwidth. For a complete description, see The Least Bandwidth Method.
Least Packets	The virtual server selects the service with the best combination of fewest packets and highest weight. For a complete description, see The Least Packets Method.
Custom Load	The virtual server selects the service with the best combination of lowest load and highest weight. For a complete description, see The Custom Load Method.
Hashing methods and Token method	Weighting is not supported by these load balancing methods.

To configure a virtual server to assign weights to services by using the CLI

At the command prompt, type:

```
set lb vserver <name> -weight <Value> <ServiceName>
```



```
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
<!--NeedCopy-->
```

To configure a virtual server to assign weights to services by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open the virtual server, and then click in the **Services** section.
3. In the weight column for the service, assign a weight to the service.

Configure the MySQL and Microsoft SQL server version setting

September 14, 2021

You can specify the version of Microsoft® SQL Server® and the MySQL server for a load balancing virtual server that is of type MSSQL and MySQL respectively. The version setting is recommended if you expect some clients to not be running the same version as your MySQL or Microsoft SQL Server product. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version.

To set the Microsoft SQL server version parameter by using the CLI

At the command prompt, type the following commands to set the Microsoft SQL Server version parameter for a load balancing virtual server and verify the configuration:

```
set lb vserver <name> -mssqlServerVersion <mssqlServerVersion>

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
> set lb vserver myMSSQLvip -mssqlServerVersion 2008R2
 Done
> show lb vserver myMSSQLvip
    myMSSQLvip (190.0.2.12:1433) - MSSQL    Type: ADDRESS
 . . .
```



```
 . . .
    MSsql Server Version: 2008R2
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

To set the MySQL server version parameter by using the CLI

At the command prompt, type the following commands to set the MySQL Server version parameter for a load balancing virtual server and verify the configuration:

```
set lb vserver <name> -mysqlServerVersion <string>

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
> set lb vserver mysqlsvr -mysqlserverversion 5.5.30
 Done
> sh lb vserver mysqlsvr
     mysqlsvr (2.22.2.222:3306) - MYSQL      Type: ADDRESS
 . . .
 . . .
     Mysql Server Version: 5.5.30
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

To set the MySQL or Microsoft SQL server version parameter by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open a virtual server of type MySQL or MSSQL, and set the server version.

Multi-IP virtual servers

September 14, 2021

The Citrix ADC supports creating a single load balancing virtual server with multiple non-consecutive/consecutive IPv4 and IPv6 addresses of type VIP. Each VIP address bound to a virtual server is treated as an individual virtual server. These virtual servers have the same protocol and other virtual server level settings. A virtual server with multiple VIP addresses is also called multi-IP virtual server.

The following are some advantages of using multi-IP virtual servers:

- A multi-IP virtual server offloads the work of creating many virtual servers with the same settings and service bindings.
- Multi-IP virtual servers effectively reduces the possibility of reaching the maximum limit on virtual server entities.
- One multi-IP virtual server can be used for clients in different subnets to connect to the same set of servers.
- Only one multi-IP virtual server can be used for IPv6 and IPv4 clients to connect to the same set of servers.

Configure a multi-IP virtual server

Configuring a multi-IP virtual server consists of the following tasks:

- Create an IPset and bind multiple IP addresses to it.
- Bind the IPset to load balancing virtual servers.

Note the following points related to IPset configuration:

- An IPset can have:
  - non-consecutive/consecutive IPv4 addresses and IPv6 addresses
  - combinations of IPv4 and IPv6 addresses.
- All IPv4/IPv6 addresses to be associated with virtual servers using IPset must be of type VIP.
- A single IPset can be bound to multiple virtual servers.
- IPv4/IPv6 addresses can be bound/unbound to/from IPset irrespective any existing IPset bindings to virtual servers.
- You must unset the IPset binding to a virtual server before binding a new IPset to it.

To add an IPset and bind multiple VIP addresses to it by using the CLI

At the command prompt, type:


```
add ipset <name>

bind ipset <name> <IPaddress1 … >

bind ipset <name> <IPaddress2… >

show ipset <name>
<!--NeedCopy-->
```

To bind the IPset to a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -ipset <ipset name>

show lb vserver <name>
<!--NeedCopy-->
```

To add an IPset and bind multiple VIP addresses to it by using the GUI

Navigate to **System** > **Network** > **IPSets**, and create an IPset with multiple VIP addresses.

To bind the IPSet to a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server to which you want to bind the created IPset.
2. In **Basic Settings**, set the **IPset** parameter to the name of the created IPset.

```
> add ipset IPSET-1


Done

> bind ipset IPSET-1 9.9.9.10


Done

> bind ipset IPSET-1 1000::20


Done
```



```
15
16 > add lb vserver LBVS-1 HTTP 8.8.8.10 80  - ipset IPSET-1
17
18
19 Done
20
21 > add service SVC-1 3.3.3.10 HTTP 80
22
23
24 Done
25
26 > add service SVC-2 3.3.3.100 HTTP 80
27
28
29 Done
30
31 > bind lb vserver LBVS-1 SVC-1
32
33
34 Done
35
36 > bind lb vserver LBVS-1 SVC-2
37
38
39 Done
```

GSLB support for multi-IP virtual servers

In cloud deployments, floating IP addresses are not supported. These IP addresses are required for high availability deployments. With IPset support, you can associate a private IP address to each of the primary and secondary instances. One of the private IP addresses is added when creating the virtual server. The other IP address is bound to an IPset. This IPset is then associated with the virtual server. Typically a public IP is mapped to one of the private IPs based on which appliance is taking the traffic. During failover, this mapping changes dynamically to route the traffic to the new primary.

In GSLB deployments, the GSLB service represents the virtual server's IP address, port number, and service type. This IP address can be the IP address that is configured while adding the virtual server or it can be one of the IP addresses in the IPset. Irrespective of the IP address used in GSLB service, statistics and state are inherited from the same load balancing virtual server entity.

Parent child topology is also supported with IPset. The load balancing virtual servers on the child sites can have the IPset associated with it. Communication between the parent and the child sites is always using public IP address and the public port of the GSLB service

Also, with IPset support, you can have a single virtual server endpoint for both IPv4 and IPv6 traffic. Previously, you had to configure different virtual servers for IPv4 and IPv6 traffic. With IPset support, you can associate IPv4 and IPv6 IP addresses to the same IP set. You can add different GSLB services representing the IPv4 and IPv6 endpoints.

Note: Only one IP address is associated with a GSLB service. You cannot associate an IPset with a GSLB service. For details on configuring GSLB entities, see the topic Configure GSLB entities individually.

Limit the number of concurrent requests on a client connection

September 14, 2021

You can limit the number of concurrent requests on a single client connection. You can protect the servers from security vulnerabilities by limiting the number of concurrent requests. When the client connection reaches the specified maximum limit, the Citrix ADC appliance drops subsequent requests on the connection until the outstanding request count goes below the limit.

You can configure the maxPipelineNat parameter to limit the number of concurrent requests on a single client connection. This parameter is applicable only to the following service types and when "svrTimeout" is set to zero:

- ANY
- All UDP service types except DNS

The default value of maxPipelineNat parameter is 255. A value of zero (0) applies no limit to the number of concurrent requests. When no limit is set, the Citrix ADC appliance executes all requests.

> **Note**
>
> If you set MaxpipelineNAT to a higher value, then the probability of spoofing attack can be higher. Hence, it is recommended to set MaxpipelineNAT to a lower value.

To limit the number of concurrent connections for a client by using the CLI

At the command prompt, type:

```
set lb parameter -maxPipelineNat <positive_integer>
<!--NeedCopy-->
```

Example:

```
set lb parameter -maxPipelineNat 199
<!--NeedCopy-->
```


To limit the number of concurrent connections for a client by using the GUI

Navigate to **Traffic Management > Load Balancing > Configure Load Balancing Parameters**, specify a value for Max Pipeline NAT requests.

Configure diameter load balancing

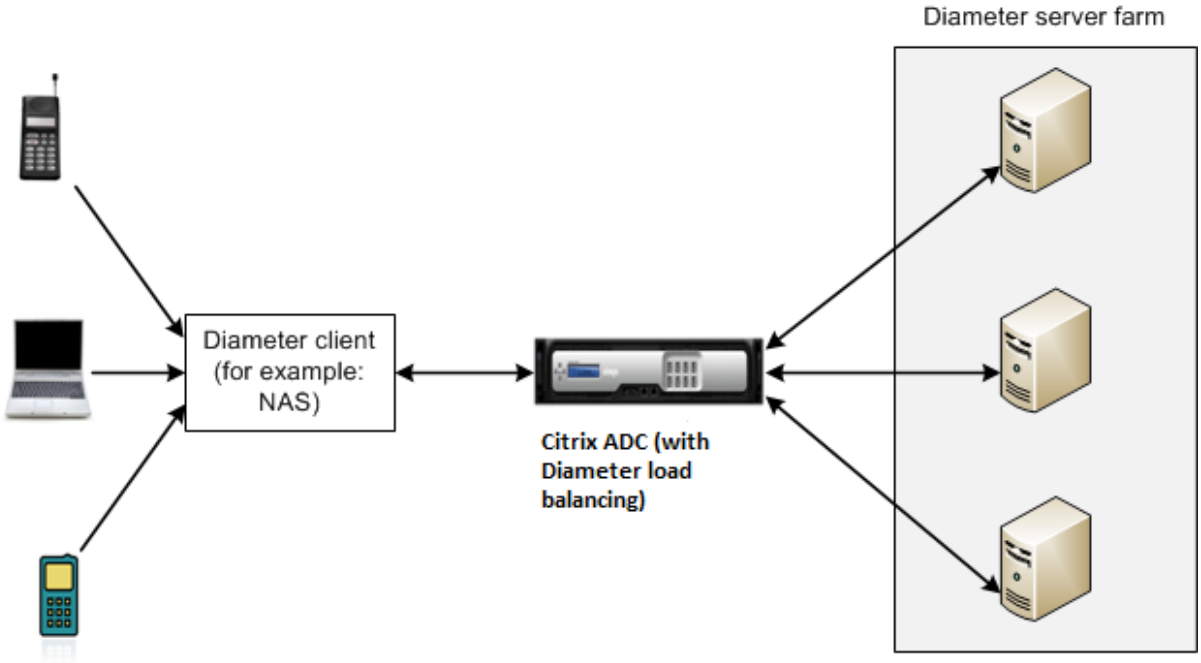
September 14, 2021

The Diameter protocol is a next generation Authentication, Authorization, and Accounting (AAA) signaling protocol used mainly on mobile devices such as laptops and mobile phones. It is a peer-to-peer protocol, as opposed to the traditional client-server model used by most other protocols. However, in most Diameter deployments, the clients originates the request and the server responds to the request.

When Diameter messages are exchanged, the Diameter server usually does much more processing than does the Diameter client. With the increase in control plane signaling volume, the Diameter server becomes a bottleneck. Therefore, Diameter messages must be load balanced to multiple servers. A virtual server performing load balancing of Diameter messages provides the following benefits:

- Lighter load on Diameter servers, which translates to faster response time to end users.
- Server health monitoring and better failover capabilities.
- Better scalability in terms of server addition without changing client configuration.
- High availability.
- SSL-Diameter offloading.

The following figure shows a Diameter system in a Citrix ADC deployment:



A Diameter system has the following components:

- **Diameter client.** Supports Diameter client applications in addition to the base protocol. Diameter clients are often implemented in devices at the edge of a network and provide access control services for that network. Typical examples of Diameter clients are a Network Access Server (NAS) and the Mobile IP Foreign Agent (FA).
- **Diameter agent.** Provides relay, proxy, redirect, or translation services. The Citrix ADC appliance (configured with a Diameter load balancing virtual server) plays the role of a Diameter agent.
- **Diameter server.** Handles the authentication, authorization, and accounting requests for a particular realm. A Diameter server must support Diameter server applications in addition to the base protocol.

In a typical Diameter topology, when an end-user device (such as a mobile phone) needs a service, it sends a request to a Diameter client. Each Diameter client establishes a single connection (TCP connection—SCTP is not yet supported) with a Diameter server as specified by the Diameter base-protocol RFC 6733. The connection is long-lived and all messages between the two Diameter nodes (client and server) are exchanged over this connection. The Citrix ADC uses message based load balancing.

Example:

A mobile service provider uses Diameter for its billing system. When a subscriber uses a prepaid number, the Diameter client repeatedly sends requests to the server to check the available balance. The Diameter protocol establishes a connection between the client and the server, and all requests are exchanged over that connection. Connection based load balancing would be pointless, because there

is only one connection. However, with the large number of messages on the connection, message based load balancing expedites the process of billing the prepaid mobile subscriber.

How diameter load balancing works

A Disconnect Peer Request (DPR) indicates the peer's intention of closing the connection, with the reason for closing the connection. The peer replies with a DPA (TCP always provides successful DPA).

- When the appliance receives a DPR from the client, it broadcasts the DPR to all servers and immediately replies with a DPA to the client. The servers reply with DPAs, but the appliance ignores them. The client sends a FIN, which the appliance broadcasts to all servers.
- When the appliance receives a DPR from the server, it replies with a DPA to that server alone, and does not remove the server from the reuse pool. When the server sends a FIN, the appliance replies with FIN/ACK and removes connections from the reuse pool.
- If the appliance receives a FIN from the client, it sends the client a FIN/ACK, broadcasts the FIN, and immediately removes the server connection from the reuse pool.
- If the appliance receives a FIN from the server, it sends a FIN/ACK and removes it from reuse pool. Any new message for this server is sent on a new connection.

Load balancing diameter traffic

When a client sends a request to the Citrix ADC appliance, the appliance parses the request and contextually load balances it to a Diameter server based on a persist AVP. The appliance has advertised the client identity to the server, so it does not add route entries, because the server is expecting messages directly from client.

Server initiated requests are not as frequent as client requests. Server initiated requests are similar to client initiated requests, except:

- Since messages are received from multiple servers, the appliance maintains the transaction state by adding a unique Hop by Hop (HbyH) number to each forwarded request message. When the message response arrives (with same HbyH number), the appliance translates this HbyH number to the HbyH number that was received on the server when the request arrived.
- The Citrix ADC appliance adds a route entry by putting its identity, because the client sees the appliance as a relay agent.

Note: If a Diameter message spans more than one packet, the appliance accumulates the packets in an incomplete header queue and forwards them to the server when the full message is accumulated. Similarly, if a single packet contains more than one Diameter message, the appliance splits the packet and forwards the messages to servers as determined by the load balancing virtual server.

Disconnect a session

A Disconnect Peer Request (DPR) indicates the peer's intention of closing the connection, with the reason for closing the connection. The peer replies with a DPA (TCP always provides successful DPA).

- When the Citrix ADC appliance receives a DPR from the client, it broadcasts the DPR to all servers and immediately replies with a DPA to the client. The servers reply with DPAs, but the appliance ignores them. The client sends a FIN, which the appliance broadcasts to all servers.
- When the appliance receives a DPR from the server, it replies with a DPA to that server alone, and does not remove the server from the reuse pool. When the server sends a FIN, the appliance replies with FIN/ACK and removes connections from the reuse pool.
- If the appliance receives a FIN from the client, it sends the client a FIN/ACK, broadcasts the FIN, and immediately removes the server connection from the reuse pool.
- If the appliance receives a FIN from the server, it sends a FIN/ACK and removes it from reuse pool. Any new message for this server is sent on a new connection.

Configure load balancing for diameter traffic

To configure the Citrix ADC appliance to load balance diameter traffic, you must first set the Diameter parameters on the appliance, then add the diameter monitor, add the diameter services, bind the services to the monitor, add the diameter load balancing virtual server, and bind the services to the virtual server.

To configure load balancing for diameter traffic by using the command line interface

Configure the diameter parameters.

```
set ns diameter -identity <string> -realm <string> -
   serverClosePropagation <YES|NO>
<!--NeedCopy-->
```

Example:

```
set ns diameter -identity mydomain.org -realm org -
   serverClosePropagation YES
<!--NeedCopy-->
```

Add a Diameter monitor.

```
add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm
    <string>
<!--NeedCopy-->
```

Example:


```
add lb monitor diameter_mon DIAMETER -originHost mydomain.org -
    originRealm org
<!--NeedCopy-->
```

Create the Diameter services.

```
add service <name> <IP> DIAMETER <port>
<!--NeedCopy-->
```

Example:

```
add service diameter_svc0 10.102.82.86 DIAMETER 3868

add service diameter_svc1 10.102.82.87 DIAMETER 3868

add service diameter_svc2 10.102.82.88 DIAMETER 3868

add service diameter_svc3 10.102.82.89 DIAMETER 3868
<!--NeedCopy-->
```

Bind the Diameter services to the Diameter monitor.

```
bind service <name>@ monitorName <monitorName>
<!--NeedCopy-->
```

Example:

```
bind service diameter_svc0 -monitorName diameter_mon

bind service diameter_svc1 -monitorName diameter_mon

bind service diameter_svc2 -monitorName diameter_mon

bind service diameter_svc3 -monitorName diameter_mon
<!--NeedCopy-->
```

Add a Diameter load balancing virtual server with Diameter persistence.

```
add lb vserver <name> DIAMETER <IPAddress> <port> -persistenceType
    DIAMETER -persistAVPno <positive_integer>
<!--NeedCopy-->
```

Example:


```
add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -
   persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

Bind the Diameter services to the Diameter load balancing virtual server.

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver diameter_vs diameter_svc0

bind lb vserver diameter_vs diameter_svc1

bind lb vserver diameter_vs diameter_svc2

bind lb vserver diameter_vs diameter_svc3
<!--NeedCopy-->
```

Save the configuration.

```
save ns config
<!--NeedCopy-->
```

Note: You can also configure load balancing of Diameter traffic over SSL by using the **SSL_DIAMETER** service type.

To configure load balancing for Diameter traffic by using the configuration utility

1. Navigate to **System** > **Settings** > **Change Diameter Parameters** and set the diameter parameters.
2. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and create a load balancing virtual server of type Diameter.
3. Create a service of type Diameter.
4. Create a monitor of type Diameter. In Special parameters, set the origin host and origin realm.
5. Bind the monitor to the service, and bind the service to the Diameter virtual server.
6. In Advanced Settings, click **Persistence**, specify the diameter, and enter a persistence AVP number.
7. Click **Save**, and click **Done**.

Configure FIX load balancing

September 14, 2021

Financial Information eXchange (FIX) protocol is an open message standard used in the financial industry for electronic exchange of information related to securities transaction between trading partners. FIX/SSL_FIX protocol is used extensively by buy-side and sell-side firms, trading platforms, and regulators for communicating trade information.

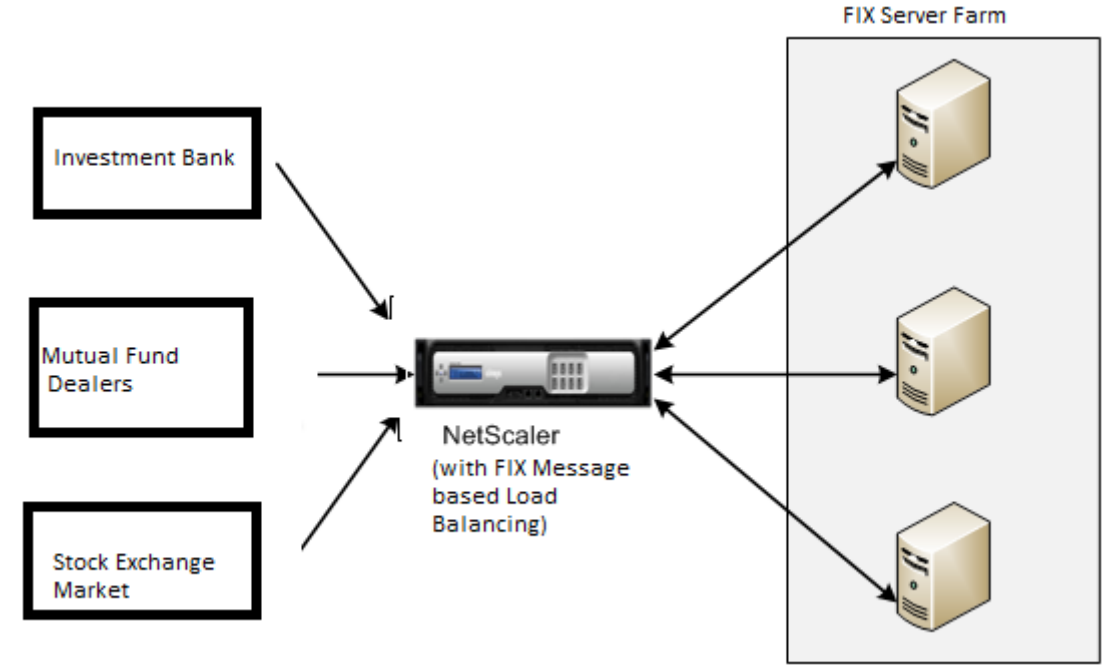
This feature enables you to configure a FIX or SSL_FIX load balancing virtual server to distribute incoming FIX messages and provide security in FIX messaging. Citrix ADC supports FIX message based load balancing (MBLB) for FIX 4.1, FIX 4.2, FIX 4.3, and FIX 4.4 versions.

FIX MBLB on a Citrix ADC appliance provides the following benefits:

1. Efficient management of FIX or SSL_FIX servers with superior HA and health monitoring.
2. SYN protection to all FIX or SSL_FIX servers.
3. FIX session persistence.

How FIX load balancing works

A FIX MBLB setup includes a FIX load-balancing virtual server and multiple load-balanced FIX servers. The FIX virtual server receives incoming client traffic, parses the incoming traffic into FIX messages, selects a FIX server for each FIX message, and forwards the message to the selected FIX server. The following conceptual drawing illustrates a typical FIX load balancing setup.



In a basic FIX MBLB setup, the FIX virtual server distributes FIX messages coming from clients to the load-balanced FIX servers using the round robin load-balancing method. With persistence of type FIXSESSION enabled, the FIX virtual server selects the same server for different FIX messages belonging to the same FIX session. The FIX session is determined based on the values of the **FIX** fields SenderCompID (tag 49) and TargetCompID (tag 56).

Configure and monitor load balancing for FIX traffic

Following are the configurations that you must do to load balance FIX message traffic:

1. Configuring FIX load balancing virtual server
2. Configuring SSL_FIX load balancing virtual server
3. Configuring FIX load balancing service
4. Configuring SSL_FIX load balancing service
5. Configuring FIXSESSION persistence
6. Setting persistence timeout
7. Displaying FIX/SSL_FIX stats
8. Monitoring FIX/SSL_FIX persistent sessions

To configure a FIX load balancing server by using the command line interface

At the command prompt, type:

```
add lb vserver <name> FIX <IP> <PORT>
<!--NeedCopy-->
```

Example

```
add lb vserver vs1 FIX 10.102.82.86 3868
<!--NeedCopy-->
```

To configure a SSL_FIX load balancing virtual server by using the command line interface

At the command prompt, type:

```
add lb vserver <name> SSL_FIX <IP> <PORT>
<!--NeedCopy-->
```

Example

```
add lb vserver vs1 SSL_FIX 10.102.82.86 3868
<!--NeedCopy-->
```


To configure a FIX service by using the command line interface

At the command prompt, type:

```
add service <name> <ip-addr> FIX <port>
<!--NeedCopy-->
```

Example

```
add service_svc1 10.102.82.86 FIX 3868
<!--NeedCopy-->
```

To configure a SSL_FIX service by using the command line interface

At the command prompt, type:

```
add service <name> <ip-addr> SSL_FIX <port>
<!--NeedCopy-->
```

Example

```
add service svc1 10.102.82.86 SSL_FIX 3868
<!--NeedCopy-->
```

To configure FIXSESSION persistence by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -persistenceType FIXSESSION
<!--NeedCopy-->
```

Example

```
set lb vserver vs1 -persistenceType FIXSESSION
<!--NeedCopy-->
```

To set persistence timeout by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -timeout <value>
<!--NeedCopy-->
```


Example

```
set lb vserver vs1  -timeout 2
<!--NeedCopy-->
```

To display FIX stats by using the command line interface

At the command prompt, type:

```
stat lb vserver <name>
<!--NeedCopy-->
```

Example

```
stat lb vserver_svc1
<!--NeedCopy-->
```

To bind FIX service to FIX virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <service name>
<!--NeedCopy-->
```

Example

```
bind lb vserver vs1 svc1
<!--NeedCopy-->
```

To display FIX persistent sessions by using the command line interface

At the command prompt, type:

```
show lb persistentSessions <name>
<!--NeedCopy-->
```

Example

```
show lb persistentSessions vs1
<!--NeedCopy-->
```


> **Note**
>
> Note: You can now configure the load balancing of FIX traffic over SSL by using the SSL_FIX service type. This service provides secured communication for FIX messages.

To configure FIX load balancing virtual server by using the GUI

1. Navigate to the **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page and click **Add** to create a FIX Load Balancing virtual server.
2. On the **Load Balancing Virtual Server** page, set the server parameters:
   a) Virtual Server Name
   b) Protocol type as “FIX”
   c) Server IP Address Type
   d) Server IP Address
   e) Server Port Number
3. Click **OK** and **Continue** to set other parameters.
4. In the **Services** section, select or add a new FIX load balancing virtual service, and bind it to the FIX server.
5. In the **Persistence** section, set the following parameters:
   a) Persistence type as ‘FIXSESSION’
   b) Time-out interval
6. Click **OK** and then **Done**.

To edit a FIX load balancing virtual server by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page, select a FIX server and click **Edit**.

To delete a FIX load balancing virtual server by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page, select a FIX server, and click **Delete**.

To configure FIX Load Balancing Virtual Service by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page and click **Add** to create a FIX Load Balancing virtual service.
2. On the **Services** page, set the following parameters. You can click the ‘More’ arrow to set other parameters such as Traffic Domain, Hash ID, Server ID, Cache Type, and Number of Active Connections.

a) Service Name – FIX Virtual Service Name
b) Choose Virtual Server type as (New or Existing)
c) Protocol – Protocol Type as 'FIX'
d) Server – Virtual Server IP address
e) Port – Server Port Number

3. Click **OK** and **Continue** to set other parameters such as Monitors, Threshold & Timeout, Profiles, and Policies.
4. Click **OK** and then **Done**.

To edit a FIX load balancing virtual service by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page, select a **FIX service** and click **Edit**.

To delete a FIX load balancing virtual service by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page, select a FIX service, and click **Delete**.

To display FIX load balancing server statistics

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page and then click **Statistics** to display the FIX server statistics.

To display Persistent sessions for a FIX server by using the GUI

Navigate to **Configuration** > **Traffic Management** page and, under **Monitor Sessions** click **Virtual Server Persistent Sessions**.

To clear Persistent sessions for a FIX server by using the GUI

1. Navigate to **Configuration** > **Traffic Management** page and, under **Monitor Sessions** click **Clear Persistent Sessions**.
2. On the **Clear Persistent Sessions** page, set the following parameters:
   a) Virtual Server – Choose a FIX virtual server
   b) Persistence Parameter – Choose a FIX persistence parameter
3. Click **OK**.

MQTT load balancing

September 14, 2021

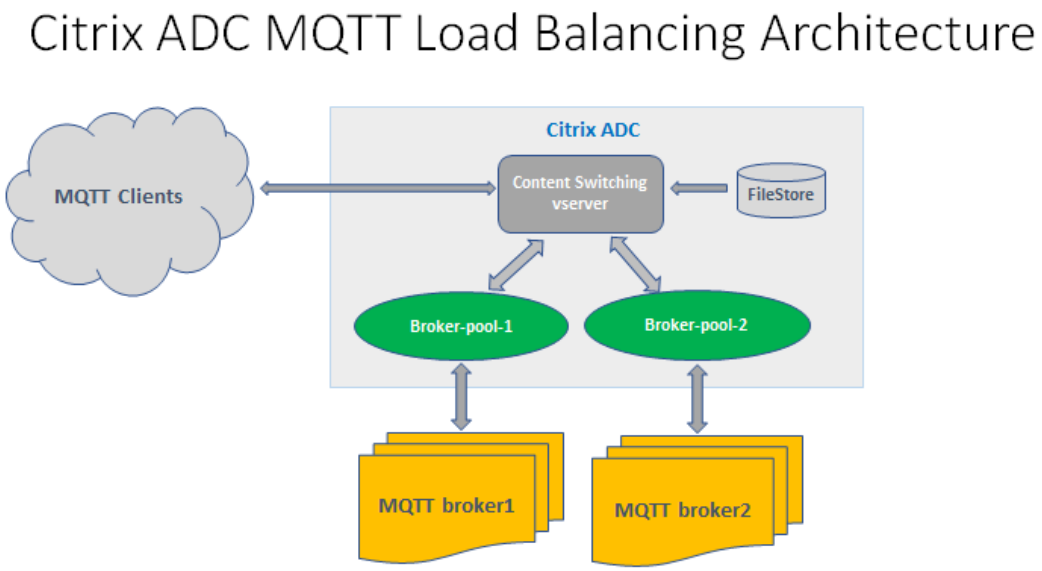
The Message Queuing Telemetry Transport (MQTT) is an OASIS standard messaging protocol for the Internet of Things (IoT). MQTT is a flexible and easy-to-use technology that provides effective communication within an IoT system. MQTT is a broker-based protocol and is widely used to facilitate the exchange of messages between clients and broker.

The following key benefits of MQTT make it a well-suited option for your IoT device:

- Reliability
- Fast response time
- Capability to support unlimited devices
- Publish/subscribe messaging that is perfect for many-to-many communication

IoT is the network of interconnected devices that are embedded with sensors, software, network connectivity, and necessary electronics. The embedded components enable IoT devices to collect and exchange data. The increase in use of IoT devices brings in multiple challenges for network infrastructure, with Scale being the prominent one. In a large scale deployment of IoT devices, the data generated by each IoT device needs to be analyzed swiftly. To achieve the scale requirement and efficient usage of resources, the load on the broker pool must be distributed evenly. With the support of the MQTT protocol, you can use the Citrix ADC appliance in IoT deployments to load balance the MQTT traffic.

The following figure depicts the MQTT architecture using a Citrix ADC appliance to load balance the MQTT traffic.



An IoT deployment with MQTT protocol has the following components:

- **MQTT broker.** A server that receives all messages from the clients and then routes the messages to the appropriate destination clients. The broker is responsible for receiving all messages, filtering the messages, determining who is subscribed to each message, and sending the message to these subscribed clients. The broker is the central hub through which every message must pass.
- **MQTT client.** Any device, from a micro controller up to a full-fledged server, which runs an MQTT library and connects to an MQTT broker over a network. Both publishers and subscribers are MQTT clients. The publisher and subscriber labels refer to whether the client is publishing messages or subscribed to receive messages.
- **MQTT load balancer.** The Citrix ADC appliance is configured with an MQTT load balancing virtual server to load balance MQTT traffic.

In a typical IoT deployment, the broker (cluster of servers) manages the group of IoT devices (IoT clients). The Citrix ADC appliance load balances the MQTT traffic to the brokers based on various parameters, such as Client ID, topic, and user name.

Configure load balancing for MQTT traffic

For the Citrix ADC appliance to load balance MQTT traffic, perform the following configuration tasks:

1. Configure MQTT/MQTT_TLS services or service groups.
2. Configure MQTT/MQTT_TLS load balancing virtual server.
3. Bind the MQTT/MQTT_TLS services to the MQTT/MQTT_TLS load balancing virtual server.

4. Configure MQTT/MQTT_TLS content switching virtual server.
5. Configure a content switching action that specifies the target load balancing virtual server
6. Configure a content switching policy.
7. Bind the content switching policy to a content switching virtual server that is already configured to redirect to the specific load balancing virtual server.
8. Save the configuration.

To configure load balancing for MQTT traffic by using the CLI

Configure MQTT/MQTT_TLS services or service groups.

```
add service <name> <IP> <protocol> <port>
add servicegroup <ServiceGroupName> <Protocol>
bind servicegroup <serviceGroupName> <IP> <port>
<!--NeedCopy-->
```

Example:

```
add service srvc1 10.106.163.3 MQTT 1883
add servicegroup srvcg1 MQTT
bind servicegroup srvcg1 10.106.163.3 1883
<!--NeedCopy-->
```

Configure MQTT/MQTT_TLS load balancing virtual server.

```
add lb vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver lb1 MQTT 10.106.163.9 1883
<!--NeedCopy-->
```

Bind the MQTT/MQTT_TLS services or service groups to the MQTT load balancing virtual server.

```
bind lb vserver <name> <serviceName>
bind lb vserver <name> <servicegroupName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver lb1 srvc1
bind lb vserver lb1 srvcg1
<!--NeedCopy-->
```


Configure MQTT/MQTT_TLS content switching virtual server.

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

Example:

```
add cs vserver cs1 MQTT 10.106.163.13 1883
<!--NeedCopy-->
```

Configure a content switching action that specifies the target load balancing virtual server.

```
add cs action <name> -targetLBVserver <string> [-comment <string>]
<!--NeedCopy-->
```

Example:

```
add cs action act1 -targetlbvserver lbv1
<!--NeedCopy-->
```

Configure a content switching policy.

```
add cs policy <policyName> [-url <string> | -rule <expression>] -
    action <actName>
<!--NeedCopy-->
```

Example:

```
add cs policy cspol1  -rule   “MQTT.COMMAND.EQ(CONNECT) && MQTT.CONNECT
    .FLAGS.QOS.eq(2)”  -action act1
<!--NeedCopy-->
```

Bind the content switching policy to a content switching virtual server that is already configured to redirect to the specific load balancing virtual server.

```
bind cs vserver <virtualServerName> -policyName <policyName> -priority
    <positiveInteger>
<!--NeedCopy-->
```

Example:

```
bind cs vserver cs1  -policyName cspol1 -priority 20
<!--NeedCopy-->
```

Save the configuration.


```
save ns config
<!--NeedCopy-->
```

To configure load balancing for MQTT traffic by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and create a load balancing virtual server of type **MQTT** or **MQTT_TLS**.
2. Create a service or service group of type MQTT.
3. Bind the service to the MQTT virtual server.
4. Click **Save**.

MQTT message length limit

The Citrix ADC appliance treats the messages with message length greater than 65536 bytes as jumbo packets, and discard them by default. The `dropmqttjumbomessage` lb parameter decides whether to process the jumbo packets or not. This parameter is by default set to **YES**, which implies that the jumbo MQTT packets are dropped by default. If this parameter is set to **NO**, the ADC appliance handles even the packets with message length greater than 65536 bytes.

To configure the ADC appliance to handle jumbo packets by using CLI:

```
Set lb parameter -dropMqttJumboMessage [YES | NO]
<!--NeedCopy-->
```

Example:

```
set lb parameter -dropMqttJumboMessage no
<!--NeedCopy-->
```

Protect a load balancing configuration against failure

September 14, 2021

When a load balancing virtual server fails, or when the virtual server is unable to handle excessive traffic, the load balancing setup can fail. You can protect your load balancing setup against failure by configuring;

- the Citrix ADC appliance to redirect excess traffic to an alternate URL,
- a backup load balancing virtual server, and
- a stateful connection failover.

Redirect client requests to an alternate URL

September 14, 2021

You can redirect requests to an alternate URL by using an HTTP 302 redirect if a load balancing virtual server of type HTTP or HTTPS goes DOWN or is disabled. The alternate URL can provide information about the status of the server. The configured redirect URL is specified in the location header of the HTTP response. The exact URL specified in the response depends on the following configuration options:

- If the configured redirect URL contains only the domain name, such as http://www.sample1.example.com, the redirect URL specified in the HTTP response appends the Uniform Resource Identifier (URI). It is specified in the HTTP request to the configured domain name. For example, if the request contains the GET http://www.sample2.example.com/images/site_nav.png header, then the location header in the redirect response specifies the location: http://www.sample1.example.com/images/site_nav.png header.

  > **Note**
  >
  > The domain names in the request and response can differ. In this topic, the two domains are referred to as sample1.example.com and sample2.example.com to explain the concept.

- If the configured redirect URL contains a complete path, then the redirect response specifies the complete configured URL, irrespective of the URI in the request. For example, the following are such URLs:
  - Requested URL - http://www.redirect.com/en/index.html
  - Redirect URL - http://www.redirect.com/en/site_down.html

The following table lists the preceding configuration options:

Configured Redirect URL	URL in HTTP Request	Header in HTTP Response
http://www.sample1.example.com	http://www.sample2.example.com/en/index.html	http://www.sample1.example.com/en/index.html
http://www.sample1.example.com/en/error.html	http://www.sample2.example.com/en/index.html	http://www.sample1.example.com/en/error.html

> **Note**
>
> - When configuring a redirect URL, the http://example.com URL is not the same as the http://example.com/ URL, because the latter contains the complete path to the Webroot path,

> /.
>
> - If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when both the primary and backup virtual servers are DOWN.

To configure a virtual server to redirect the client request to a URL by using the CLI

1. Create a load balancing virtual server.

   ```
   set lb vserver -redirect url
   ```

2. Verify that the redirect URL option is working as expected. Disable the virtual server.

   ```
   disable vserver <vserver_name>
   ```

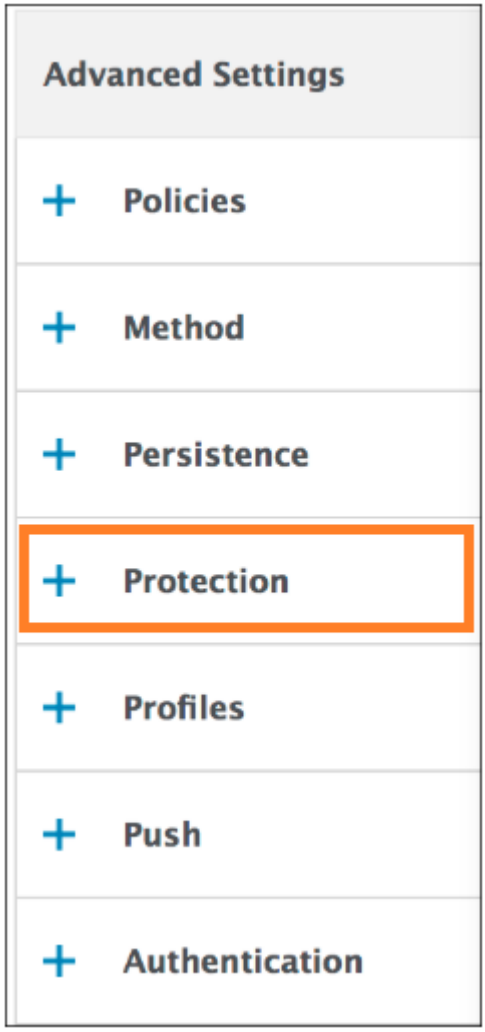
3. Access the website URL from a web browser to verify that the request is redirected as expected. You might have to clear the web browser cache and make a new connection before accessing the website.

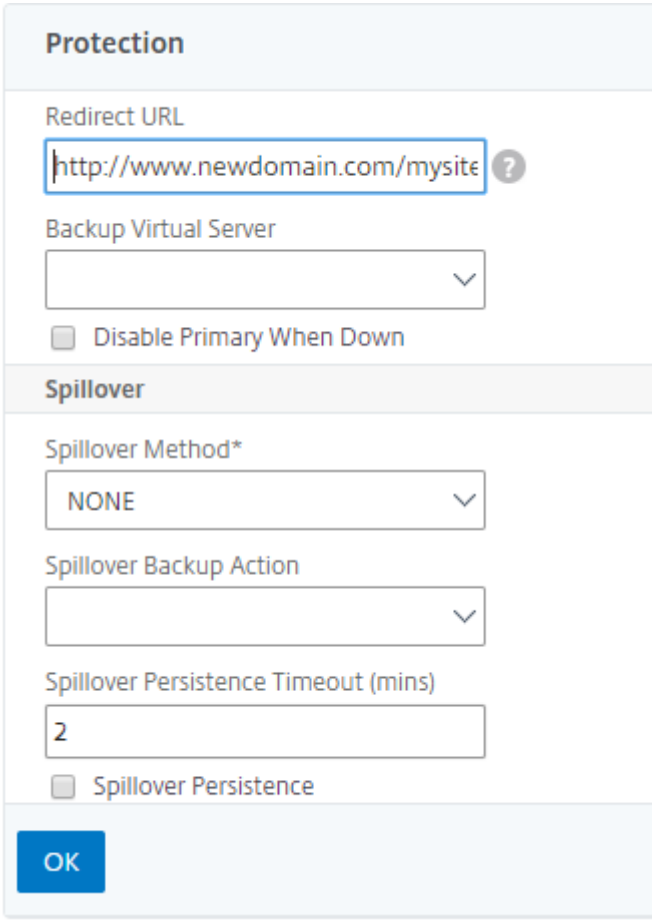
4. Enable virtual server.

   ```
   enable vserver <vserver_name>
   ```

To configure a virtual server to redirect the client request to a URL by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, to add a new virtual server, click **Add**.
3. To edit an existing virtual server, select the virtual server from the list and click **Edit**.
4. On the **Advanced Settings** tab, click **Protection**. In the **Redirect URL** field, type the redirect URL (for example, http://www.newdomain.com/mysite/maintenance).





5. Click **OK**.

Configure a backup load balancing virtual server

September 14, 2021

You can configure the Citrix ADC appliance to direct requests to a backup virtual server when the primary load balancing virtual server is DOWN or unavailable. The backup virtual server is a proxy and is transparent to the client. The appliance can also send a notification message to the client regarding the site outage.

> Note:
>
> The backup virtual server continues to handle the existing connections, even after the primary virtual server is deleted or disabled.

You can configure a backup load balancing virtual server when you create it, or you can change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an

existing backup virtual server, thus creating cascading backup virtual servers. The maximum depth of cascading backup virtual servers is 10.

If you have multiple virtual servers that connect to two servers, you have a choice for what happens if the primary virtual server goes DOWN and then comes back up. The default behavior is for the primary virtual server to resume its role as primary. However, you can configure the backup virtual server to remain in control when it takes over. For example, you can sync the updates on the backup virtual server to the primary virtual server and then manually force the original primary server to resume its role. In this case, you can designate the backup virtual server to remain in control when the primary virtual server goes DOWN and then comes back up.

You can configure a redirect URL on the primary load balancing virtual server as a fallback for when both the primary and the backup virtual servers are DOWN or have reached their threshold for handling requests. When services bound to virtual servers are OUT OF SERVICE, the appliance uses the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when the primary and backup virtual servers are down.

To set a backup virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-
   disablePrimaryOnDown]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -
   disablePrimaryOnDown
<!--NeedCopy-->
```

To set a backup virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and select a backup virtual server.
3. If you want the backup virtual server to remain in control until you manually enable the primary virtual server even if the primary virtual server comes back up, select **Disable Primary When Down**.

> **Note:** Starting from Citrix ADC version 12.1 build 51.xx, the GUI displays the effective state of that server indicating if the backup is active or not.
>
> The effective state of the current server can be one of the following:
>
> - **UP** – Indicates that the server is UP
> - **DOWN** – Indicates that the server is DOWN
> - **UP (Backup Active)** – Indicates that either the primary or the secondary virtual server is UP and traffic is directed to the backup virtual server.
> - **DOWN (Backup Active)** – Indicates that both the primary and the backup virtual servers are down and traffic is routed to the backup virtual server.
>
> When the **Disable Primary When Down** option is enabled on the primary virtual server and the primary server goes DOWN and is back UP again, the traffic is still served by the backup virtual server until the primary virtual server is re-enabled explicitly. You can use the command `enable lb vserver <vserver_name>` command to re-enable the primary virtual server.

Configure spillover

September 14, 2021

A spillover configuration on the appliance consists of a primary virtual server that is configured with a spillover method, a spillover threshold, and a backup virtual server. Backup virtual servers can also be configured for spillover, creating a chain of backup virtual servers.

The spillover method specifies the operational condition on which you want to base your spillover configuration (for example, the number of established connections, bandwidth, or combined health of the server farm). When a new connection arrives, the appliance verifies that the primary virtual server is up and compares the operational condition with the configured spillover threshold. If the threshold is reached, the spillover feature diverts new connections to the first available virtual server in the backup chain. The backup virtual server manages the connections it receives until the load on the primary falls below the threshold.

If you configure spillover persistence, the backup virtual server continues to process the connections it received, even after the load on the primary falls below the threshold. If you configure spillover persistence and a spillover persistence timeout, the backup virtual server processes connections for only the specified period after the load on the primary falls below the threshold.

Note: Usually, spillover is triggered if the value associated with the spillover method exceeds the threshold (for example, number of connections). However with the server-health spillover method, spillover is triggered if the health of the server farm falls below the threshold.

You can configure spillover in one of the following ways:

- Specify a predefined spillover method. Four predefined methods are available, and they fulfill common spillover requirements.
- Configure policy based spillover. In policy based spillover, you use a Citrix ADC rule to specify the conditions for spillover to occur. Citrix ADC rules give you the flexibility to configure spillover for various operational conditions.

Use policy based spillover if a predefined method does not satisfy your requirements. If you configure both for a primary virtual server, the policy based spillover configuration takes precedence over the predefined method.

First, you create the primary virtual server and the virtual servers that you need for the backup chain. You set up the backup chain by specifying one virtual server as the backup for the primary (that is, you create a secondary virtual server), a virtual server as the backup for the secondary (that is, you create a tertiary virtual server), and so on. Then, you configure spillover by either specifying a predefined spillover method or creating and binding spillover policies.

For instructions for assigning a virtual server as the backup for another virtual server, see Configuring a Backup Load Balancing Virtual Server.

Configure a predefined spillover method

Predefined spillover methods fulfill some of the more common spillover requirements. To use one of the predefined spillover methods, you configure spillover parameters on the primary virtual server. To create a chain of backup virtual servers, you also configure spillover parameters on backup virtual servers.

If the backup virtual servers reach their own threshold values, and the service type is TCP, the Citrix ADC appliance sends clients a TCP reset. For service types HTTP, SSL, and RTSP, it diverts new requests to the redirect URL configured for the primary virtual server. A redirect URL can be specified for only HTTP, SSL, and RTSP virtual servers. If a redirect URL is not configured, the Citrix ADC appliance sends clients a TCP reset (if the virtual server is of type TCP) or an HTTP 503 response (if the virtual server is of type HTTP or SSL).

Note: With RTSP virtual servers, the Citrix ADC appliance uses only data connections for spillover. If the backup RTSP virtual server is not available, the requests are redirected to an RTSP URL and an RTSP redirect message is sent to the client.

To configure a predefined spillover method for a virtual server by using the command line interface

At the command prompt, type:


```
set lb vserver <vServerName> -soMethod <spillOverType> -soThreshold <
    positiveInteger> -soPersistence ENABLED -soPersistenceTimeout <
    positiveInteger>
<!--NeedCopy-->
```

Example

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

To configure a predefined spillover method for a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and set the spillover parameters.

Configure policy based spillover

Spillover policies, which are based on rules (expressions), enable you to configure the appliance for a wider range of spillover scenarios. For example, you can configure spillover based on the virtual server's response time, or based on the number of connections in the virtual server's surge queue.

To configure policy based spillover, first create a spillover action. You then select the expression that you want to use in the spillover policy, configure the policy, and associate the action with it. Finally, you bind the spillover policy to a load balancing, content switching, or global server load balancing virtual server. You can bind multiple spillover policies to a virtual server, with priority numbers. The appliance evaluates the spillover policies in ascending order of priority numbers and performs the action associated with the last policy to evaluate to TRUE.

A virtual server can also have a backup action. The backup action is performed if the virtual server does not have one or more backup virtual servers, or if all backup virtual servers are DOWN, disabled, or have reached their own spillover limits.

When a spillover policy results in an UNDEF condition (an exception thrown when the result of policy evaluation is undefined), an UNDEF action is performed. The UNDEF action is always ACCEPT. You cannot specify an UNDEF action of your choice.

Configuring a Spillover Action

A spillover action is performed when the spillover policy with which it is associated evaluates to TRUE. Currently, SPILLOVER is the only supported spillover action.

To configure policy based spillover by using the command line interface

At the command prompt, type the following commands to configure a spillover policy and verify the configuration:

```
add spillover action <name> -action SPILLOVER

show spillover action <name>
<!--NeedCopy-->
```

Example

```
add spillover action mySoAction -action SPILLOVER
Done
<!--NeedCopy-->
```

```
show spillover action mySoAction
1) Name: mySoAction Action: SPILLOVER
Done
<!--NeedCopy-->
```

Selecting an Expression for the Spillover Policy

In the policy expression, you can use any virtual-server based expression that returns a Boolean value. For example, you can use one of the following expressions:

```
SYS.VSERVER("vserver").RESPTIME.GT(<int>)
SYS.VSERVER("vserver").STATE.EQ( " <string> " ), and
SYS.VSERVER("vserver").THROUGHPUT.LT    (<int>)
<!--NeedCopy-->
```

In addition to the existing functions such as RESPTIME, STATE, and THROUGHPUT, you can use the following virtual server based functions that have been introduced with this feature:

`Averagesurgecount`

Returns the average number of requests in the surge queues of active services. Returns 0 (zero) if there are no active services. Raises an UNDEF condition if used with a content switching or global server load balancing virtual server.

`Activeservices`

Returns the number of active services. Raises an UNDEF condition if used with a content switching or global server load balancing virtual server.

`Activetransactions`

Returns the value of the virtual-server-level counter for current active transactions.

is_dynamic_limit_reached

Returns a Boolean TRUE if the number of connections that the virtual server manages is equal to the dynamically calculated threshold. The dynamic threshold is the sum of the maximum client (Max Clients) settings of the bound services that are UP.

You can use a policy expression to implement any of the predefined spillover methods. The following table maps the predefined spillover methods to the expressions you can use to implement them:

Table 1. Converting predefined spillover methods to policy expressions

Predefined spillover method	Corresponding expression
CONNECTION	SYS.VSERVER("<vserver-name>").CONNECTIONS, used with the GT(int) arithmetic function.
BANDWIDTH	SYS.VSERVER("<vserver-name>").THROUGHPUT, used with the GT(int) arithmetic function.
HEALTH	SYS.VSERVER("<vserver-name>").HEALTH, used with the LT(int) arithmetic function.

Predefined spillover method	Corresponding expression
DYNAMICCONNECTION	SYS.VSERVER("<vserver-name>").IS_DYNAMIC_LIMIT_REACHED **Note**: If you implement policy based spillover by using the IS_DYNAMIC_LIMIT_REACHED function, you must also configure the predefined DYNAMICCONNECTION method for the virtual server, so that the statistics required for spillover to work are collected.

Configuring a Spillover Policy

A spillover policy uses a Boolean expression as a rule to specify the conditions that must be met for spillover to occur.

To configure a spillover policy by using the command line interface

At the command prompt, type the following commands to configure a spillover policy and verify the configuration:

```
add spillover policy <name> -rule <expression> -action <string> [-
    comment <string>]

show spillover policy <name>
<!--NeedCopy-->
```

Example

```
> add spillover policy mySoPolicy -rule SYS.VSERVER("v1").RESPTIME.GT
    (50) -action mySoAction -comment "Triggers spillover when the
    vserver's response time is greater than 50 ms."
Done

> show spillover policy mySoPolicy

1)  Name: mySoPolicy Rule: "SYS.VSERVER("v1").RESPTIME.GT(50)" Action:
    mySoAction Hits: 0 ActivePolicy: 0
Comment: "Triggers spillover when the vserver's response time is
    greater than 50 ms."
Done
```



```
>
<!--NeedCopy-->
```

Binding a Spillover Policy to a Virtual Server

You can bind a spillover policy to load balancing, content switching, or global server load balancing virtual servers). You can bind multiple policies to a virtual server, with Goto expressions controlling the flow of evaluation.

To bind a spillover policy to a virtual server by using the command line interface

At the command prompt, type the following commands to bind a spillover policy to a load balancing, content switching, or global server load balancing virtual server and verify the configuration:

```
bind (lb | cs | gslb) vserver <name> -policyName <string> -priority <
    positive_integer> [-gotoPriorityExpression <expression>]

show (lb | cs | gslb) vserver <name>
<!--NeedCopy-->
```

Example

```
> bind lb vserver vserver1 -policyName mySoPolicy -priority 5
Done
> show lb vserver vserver1
vserver1 (2.2.2.12:80) - HTTP Type: ADDRESS
. . .

1)  Spillover Policy Name: mySoPolicy Priority: 5
GotoPriority Expression: END
Flowtype: REQUEST
Done
>
<!--NeedCopy-->
```

Configuring a Backup Action for a Spillover Event

A backup action specifies what to do when the spillover threshold is reached but one or more backup virtual serves are either not configured or are down, disabled, or have reached their own thresholds.

Note: For the predefined spillover methods that are configured directly on the virtual server (as values of the Spillover Method parameter), the backup action is not configurable. By default, the appliance

sends clients a TCP reset (if the virtual server is of type TCP) or an HTTP 503 response (if the virtual server is of type HTTP or SSL).

The backup action is configured on the virtual server. You can configure the virtual server to accept requests (after the threshold specified by the policy is reached), redirect clients to a URL, or simply drop requests even before establishing TCP or SSL connections until the number of requests falls below the threshold. Therefore, lesser memory resources are utilized as the connections are reset even before allocating any data structures.

To configure a backup action for spillover by using the CLI

At the command prompt, type the following commands to configure a backup action and verify the configuration:

```
set lb vserver <name> -soBackupAction <soBackupAction>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver vs1 -soBackupAction REDIRECT -redirectURL `http://www.
   mysite.com/maintenance`
Done
> show lb vserver vs1
vs1 (10.102.29.76:80) - HTTP Type: ADDRESS
State: UP
. . .
Redirect URL: `http://www.mysite.com/maintenance`
. . .
Done
<!--NeedCopy-->
```

To configure a backup action for spillover by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and then specify a spillover backup action.

Connection failover

September 14, 2021

Connection failover helps prevent disruption of access to applications deployed in a distributed environment. In a Citrix ADC High Availability (HA) setup, *connection failover* (or *connection mirroring-CM*) refers to keeping active an established TCP or UDP connection when a failover occurs. The new primary Citrix ADC appliance has information about the connections established before the failover and continues to serve those connections. After failover, the client remains connected to the same physical server. The new primary appliance synchronizes the information with the new secondary appliance. If the L2Conn parameter is set, Layer 2 connection parameters are also synchronized with the secondary.

> Note:
>
> Consider a HA setup, where a client establishes a session with the primary node, which in turn establishes a session with the back end server. When a failover is triggered in this state, the packets received on a new primary from the existing client and server nodes are treated as stale packets, and the client and server connections are reset. Whereas if stateless connection failover is enabled (USIP is ON), after the failover, the connections are not reset when you receive packets from client or server nodes. Instead, the client and server connections are created dynamically.

You can set up connection failover in either stateless or stateful mode. In the stateless connection failover mode, the HA nodes do not exchange any information about the connections that are failed over. This method has no runtime overhead.

In the stateful connection failover mode, the primary appliance synchronizes the data of the failed-over connections with the new secondary appliance.

Connection failover is helpful if your deployment has long lasting connections. For example, if you are downloading a large file over FTP and a failover occurs during the download, the connection breaks and the download is aborted. However, if you configure the connection failover in stateful mode, the download continues even after the failover.

How connection failover works on Citrix ADC appliances

In a stateless connection failover, the new primary appliance tries to re-create the packet flow according to the information contained in the packets it receives.

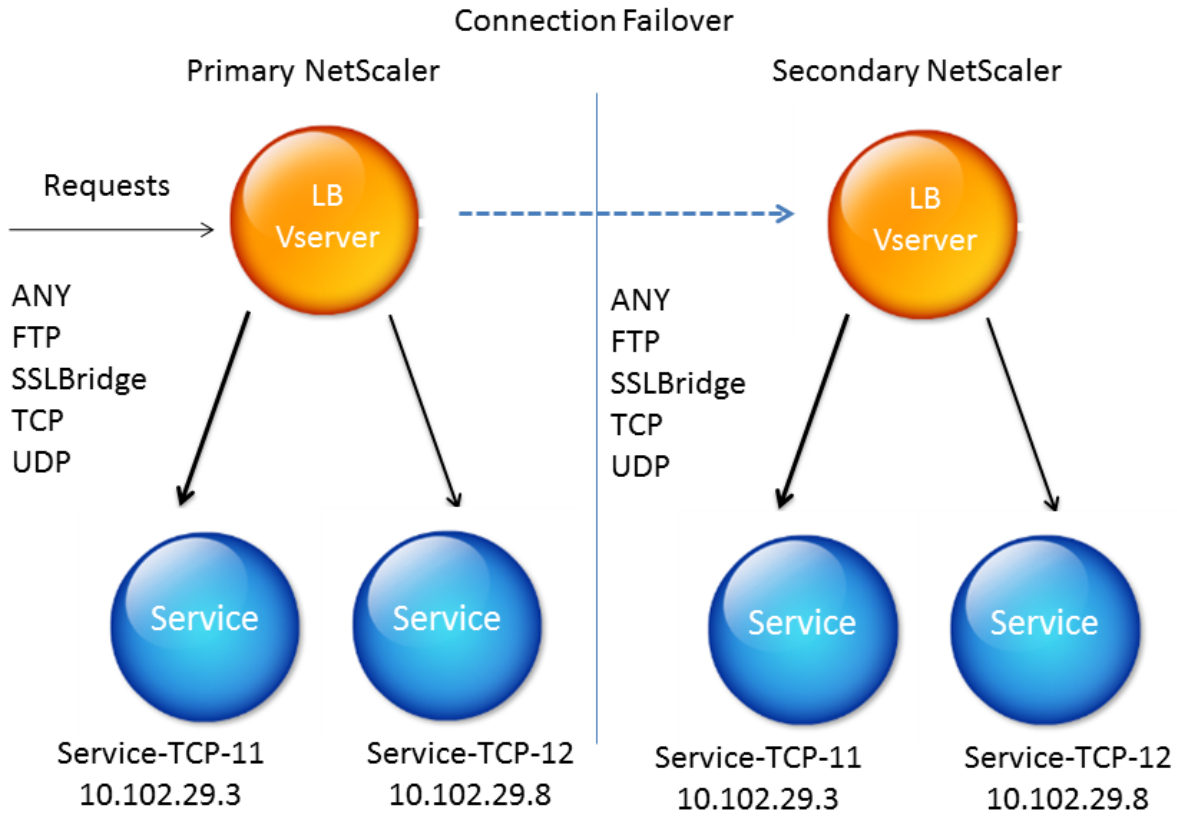
In stateful failover, to maintain current information about the mirrored connections, the primary appliance sends messages to the secondary appliance. The secondary appliance maintains the data related to the packets but uses it only in the event of a failover. If a failover occurs, the new primary (old secondary) appliance starts using the stored data about the mirrored connections and accepting traffic. During the transition period, the client and server might experience a brief disruption and retransmissions.

Note:

Verify that the primary appliance is able to authorize itself on the secondary appliance. To verify the correct configuration of the passwords, use the show `rpcnode` command from the command line or use the RPC option of the **Network** menu in the GUI.

A basic HA configuration with connection failover contains the entities shown in the following figure.

Figure 1. Connection Failover Entity Diagram



Note

Connection failover is not supported after either of the following events:

```
- An upgrade to a later release.
- An upgrade to a later build within the same release, if the new
    build uses a different HA version.
```

Supported setup

Connection failover can be configured only on load balancing virtual servers. It cannot be configured on content switching virtual servers. If you enable connection failover on load balancing virtual servers that are attached to a content switching virtual server, connection failover does not work be-

cause the load balancing virtual servers do not initially accept the traffic.

The following table describes the setup supported for connection failover.

Table 1. Connection Failover - Supported Setup

Setting	Stateless	Stateful
Service type	ANY.	ANY, UDP, TCP, FTP, SSL_BRIDGE.
Load balancing methods	All methods supported for the service type ANY. However, if Source IP persistence is not set, the SRCIPSRCPORTHASH method must be used.	All methods applicable to the supported service types.
Persistence types	SOURCEIP persistence.	All types applicable to the supported service types are supported.
USIP	Must be ON.	No restriction. It can be ON or OFF.
Service bindings	Service can be bound to only one virtual server.	Service can be bound to one or more virtual servers.
Internet Protocol (IP) versions	IPv4 and IPv6	IPV4 and IPv6
Redundancy support	Clustering and high availability	High availability

> Note:
>
> Stateful connection failover is supported only for connection-based switching services, for example, TCP. Because HTTP uses request-based switching, it does not support connection failover. In SSL, the existing connections are reset after the failover.

Features affected by connection failover

The following table lists the features affected if connection failover is configured.

Table 2. How Connection Failover Affects Citrix ADC Features

Feature	Impact of Connection Failover
SYN protection	For any connection, if a failover occurs after the appliance issues SYN-ACK but before it receives the final ACK, the connection is not supported by the connection failover. The client must reissue the request to establish the connection.
Surge protection	If the failover occurs before a connection with the server is established, the new primary appliance tries to establish the connection with the server. It also retransmits all the packets held during surge protection.
Access down	If enabled, the access-down functionality takes precedence over the connection failover.
Application firewall	The Application firewall feature is not supported.
INC	Independent network configuration is not supported in the high availability mode.
TCP buffering	TCP buffering is not compatible with connection mirroring.
Close on response	After failover, the NATPCBs might not be closed on response.

To configure connection failover by using GUI

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**. Open the virtual server, and in **Advanced Settings**, click **Protection**, and select **Connection Failover** as **Stateful**.

To configure connection failover by using CLI

At the command prompt:

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -connFailover stateful
```



```
Done
<!--NeedCopy-->
```

When connection failover is disabled on a virtual server, the resources allocated to the virtual server are freed.

To disable connection failover by using CLI

At the command prompt:

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -connFailover disable
Done
<!--NeedCopy-->
```

To disable connection failover by using GUI

Navigate to **Traffic Management**> **Load Balancing**> **Virtual Servers**. Open the virtual server, in **Protection**, select **Connection Failover** as Disabled.

Flush the surge queue

September 14, 2021

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. The Citrix ADC appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established and thus avoid overloads.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between the client and the server. However, if the appliance cannot establish a new connection with the server, it sends

the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection. The length of a surge queue decreases in any of the following conditions:

- A request in the queue gets sent to the server.
- A request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you might want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed. The other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN, or GSLB virtual servers or GSLB services.

Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the CLI

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while running the command, the surge queue of the specified entity is flushed. If more than one entity has the same name, the appliance flushes the surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while running the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a

<serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.

- If you run the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
<!--NeedCopy-->
```

Examples

```
flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
<!--NeedCopy-->
```

The previous command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10

```
flush ns surgeQ
<!--NeedCopy-->
```

The previous command flushes all the surge queues on the appliance.

To flush a surge queue by using the GUI

Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**, select a virtual server and, in the Action list, select **Flush Surge Queue**.

Manage a load balancing setup

September 14, 2021

An existing Load Balancing setup does not require a great deal of work to maintain as long as it is unchanged, but most do not remain unchanged for long. Increasing load requires new load-balanced servers and eventually new Citrix ADC appliances, which must be configured and added to the existing setup. Old servers wear out and must be replaced, requiring removal of some servers and addition of others. Upgrades to your networking equipment or changes to topology might also require modifications to your load balancing setup. Therefore, you need to perform operations on server objects,

services, and virtual servers. The Visualizer can display your configuration graphically, and you can perform operations on the entities in the display. You can also take advantage of other features that facilitate management of the traffic through your load balancing setup.

Manage server objects

September 14, 2021

During basic load balancing setup, when you create a service, a server object with the IP address of the service is created, if one does not exist. If you prefer for your service objects named with domain names rather than IP addresses, you might also have created one or more server objects manually. You can enable, disable, or remove any server object.

When you enable or disable a server object, you enable or disable all services associated with the server object. When you refresh the Citrix ADC appliance after disabling a server object, the state of its service appears as OUT OF SERVICE. If you specify a wait time when disabling a server object, the server object continues to handle established connections for the specified amount of time, but rejects new connections. If you remove a server object, the service to which it is bound is also deleted.

To enable a server by using the CLI

At the command prompt, type:

```
enable server <name>
<!--NeedCopy-->
```

Example:

```
enable server 10.102.29.5
<!--NeedCopy-->
```

To enable or disable a server object by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Servers**.
2. Select the server and, in the Action list, select **Enable** or **Disable**.

To disable a server object by using the CLI

At the command prompt, type:


```
disable server <name> <delay>
<!--NeedCopy-->
```

Example:

```
disable server 10.102.29.5 30
<!--NeedCopy-->
```

To remove a server object by using the CLI

At the command prompt, type:

```
rm server <name>
<!--NeedCopy-->
```

Example:

```
rm server 10.102.29.5
<!--NeedCopy-->
```

To remove a server object by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Servers**.
2. Select a server, and click **Remove**.

Manage services

September 14, 2021

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time during which the service continues to handle established connections, but rejects new ones, before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the service's state is OUT OF SERVICE.

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the Citrix ADC configuration.

To enable or disable a service by using the CLI

At the command prompt, type:

```
enable service <name>

disable service <name> <DelayInSeconds>
<!--NeedCopy-->
```

Examples:

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
<!--NeedCopy-->
```

To enable or disable a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Open a service and, in the **Action** list, select **Enable** or **Disable**.

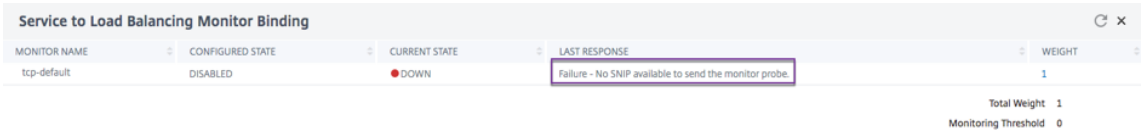
Identify the cause for the service state marked DOWN by using the GUI

Starting from Citrix ADC version 13.0 build 41.20, you can view the monitor probe information on the GUI for the services that are DOWN without navigating to the monitor binding interface. The value in the **Server State** column of the Services page is clickable. You can click **DOWN** to identify the root cause because of which the service is marked DOWN.

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Click **DOWN** in the **Server State** column corresponding to the service that is DOWN.

The Service to Load Balancing Monitor Binding page appears.

The **Last Response** column displays the reason because of which the service is marked DOWN.



Manage a load balancing virtual server

September 14, 2021

Virtual servers are enabled by default when you create them. You can disable and enable virtual servers manually. If you disable a virtual server, the virtual service's state appears as OUT OF SERVICE. When this happens, the virtual server terminates all connections, either immediately or after allowing existing connections to complete, depending on the setting of the downStateFLush parameter. If downStateFLush is ENABLED (default), all the connections are flushed. If DISABLED, the virtual server continues to serve requests on existing connections.

You remove a virtual server only when you no longer require the virtual server. Before you remove it, you must unbind all services from it.

To enable or disable a virtual server by using the CLI

At the command prompt, type:

```
enable lb vserver <name>
<!--NeedCopy-->
```

```
disable lb vserver <name>
<!--NeedCopy-->
```

Examples:

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
<!--NeedCopy-->
```

To enable or disable a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Select a virtual server, and in the **Action** list, select **Enable** or **Disable**.

To unbind a service from a virtual server by using the CLI

At the command prompt, type:

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```


Example:

```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

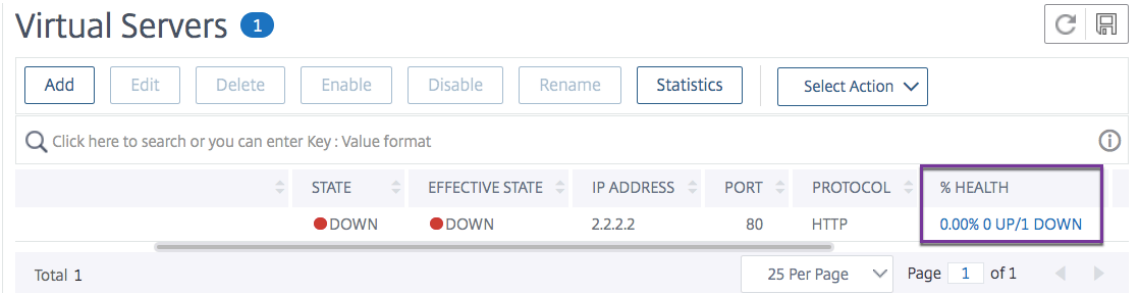
To unbind a service from a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and click in the **Services** section.
3. Select a service and click **Unbind**.

Identify the cause for the virtual server state marked DOWN by using the GUI

Starting from Citrix ADC version 13.0 build 41.20, you can view the monitor probe information on the GUI for the virtual servers that are DOWN without navigating to the monitor binding interface. The value in the **% HEALTH** column of the Virtual Server page is clickable. You can click the value in the **% HEALTH** column to identify the root cause because of which the virtual server is marked DOWN.

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Click the value in the **% HEALTH** column corresponding to the virtual server that is down.

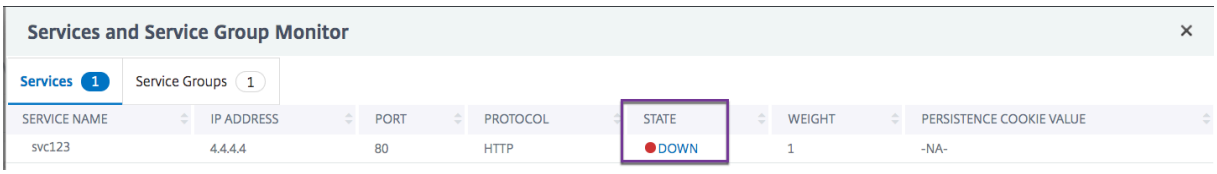


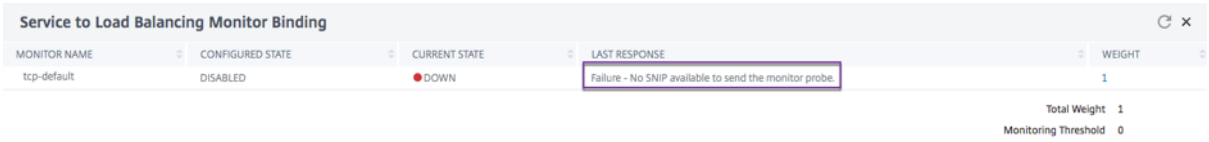
The Service and Service Group Monitor page appears. The services and service groups bound to this virtual server are displayed in the respective tabs.

If you are using services bound to load balancing virtual, perform the following:

In the **Services** tab, click **DOWN** corresponding to the service that is down.

The **Last Response** column in the Service to Load Balancing Monitor Binding page displays the reason because of which the virtual server is marked down.

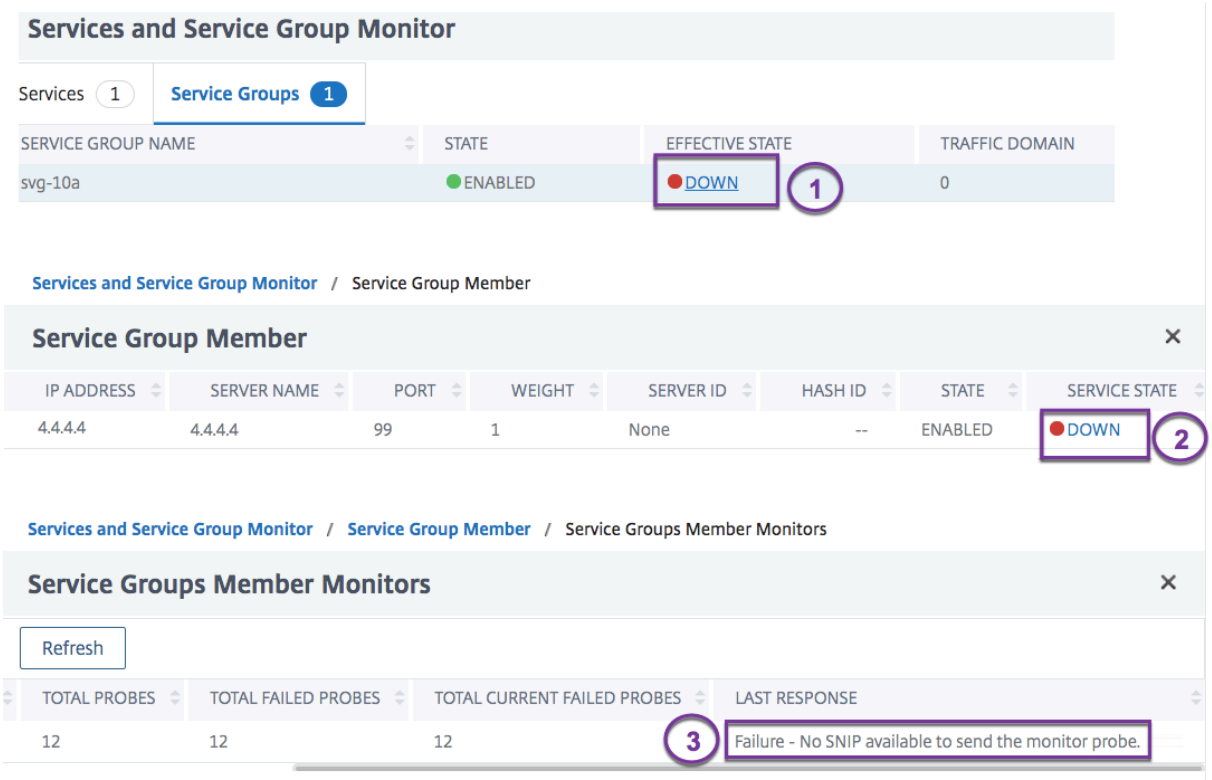




If you are using service groups bound to load balancing virtual, perform the following:

In the **Service Groups** tab, Click **DOWN** in the Services and Service Group Monitor page and then click **DOWN** in the Service Group Member page.

The **Last Response** column in the Service Groups Member Monitors page displays the reason because of which the virtual server is marked down.

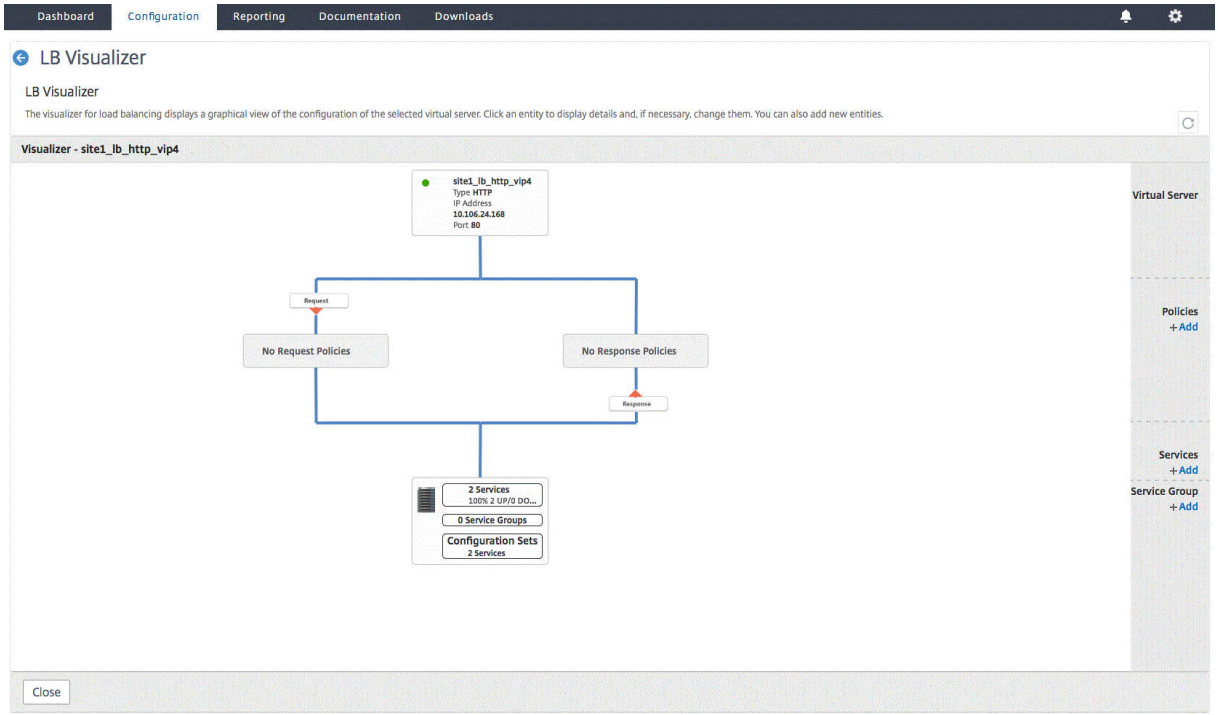


Load balancing visualizer

September 14, 2021

The Load Balancing Visualizer is a tool that you can use to view and modify the load balancing configuration in a graphical format. Following is an example of the Visualizer display.

Figure 1. Load Balancing Visualizer Display



You can use the visualizer to view the following:

- The services and service groups that are bound to a virtual server.
- The monitors that are bound to each service.
- The policies that are bound to the virtual server.
- The policy labels, if configured.
- Configuration details of any displayed element.

You can also use the Visualizer to add and bind new objects, modify existing ones, and enable or disable objects. Most configuration elements displayed in the Visualizer appear under the same names as in other parts of the configuration utility. However, unlike the rest of the configuration utility, the Visualizer groups services that have the same configuration details and monitor bindings into an entity called a service container.

A service container is set of similar services and service groups that are bound to a single load balancing virtual server. The services in the container have the same properties, except for the name, IP address, and port, and their monitor bindings must have the same weight and binding state. When you bind a new service to a virtual server, it is placed into an existing container if its configuration and monitor bindings match those of other services. Otherwise, it is placed in its own container.

The following procedures provide only the basic steps for using the Visualizer. Because the Visualizer duplicates functionality in other areas of the Load Balancing feature, other methods of viewing or configuring all the settings that can be configured in the Visualizer are provided throughout the Load Balancing documentation.

Note: The Visualizer requires a graphic interface, so it is available only through the configuration util-

ity.

To view load balancing virtual server properties by using the Visualizer

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.

To view configuration details for services, service groups, and monitors by using the Visualizer

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click **Visualizer**.
3. In the Load Balancing Visualizer dialog box, double-click the entity to view the configuration details of the entity that is bound to this virtual server, you can do the following:

To view configuration details for policies and policy labels by using the Visualizer in the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, double-click the policies entity to view the policies that are bound to this virtual server.

To modify a resource in a load balancing configuration by using the Visualizer

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, on the Visualizer image, double-click the resource that you want to modify.

To add a load balancing configuration by using the Visualizer

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, select the virtual server that you want to configure, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, click **+** to add the resource.

Manage client traffic

September 14, 2021

Managing client connections properly helps to ensure that your applications remain available to users even when your Citrix ADC appliance is experiencing high loads. Various load balancing features and other features available on the appliance can be integrated into a load balancing setup to process load more efficiently, divert it when necessary, and prioritize the tasks that the appliance must perform:

- **Sessionless load balancing.** You can configure sessionless load balancing virtual servers and perform load balancing without creating sessions in configurations that use DSR or intrusion detection systems (IDS).
- **Integrated caching.** You can redirect HTTP requests to a cache.
- **Priority queuing.** You can direct requests based on priority, by integrating your configuration with the Priority Queuing feature.
- **SureConnect.** You can use load balancing with the Sure Connect feature to redirect important requests to a custom webpage, insulating them from delays due to network congestion.
- **Delayed cleanup.** You can configure delayed cleanup of virtual server connections to prevent the cleanup process from using CPU cycles during periods when the Citrix ADC appliance is experiencing high loads.
- **Rewrite.** You can use the Rewrite feature to modify port and protocol when performing HTTP redirection, or insert the virtual server IP address and port into a custom Request header.
- **RTSP NAT.**
- **Rate-based monitoring.** You can enable rate-based monitoring to divert excess traffic.
- **Layer 2 Parameters.** You can configure a virtual server to use the L2 parameters to identify a connection.
- **ICMP Response**. You can configure the appliance to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the `ICMP VSERVER RESPONSE`. The following settings can be made on a virtual server:
  - When you set `ICMP VSERVER RESPONSE` to PASSIVE on all virtual servers, the appliance always responds.
  - When you set `ICMP VSERVER RESPONSE` to ACTIVE on all virtual servers, the appliance responds even if one virtual server is UP.
  - When you set `ICMP VSERVER RESPONSE` to ACTIVE on some and PASSIVE on others, the appliance responds even if one virtual server set to ACTIVE is UP.

Configure sessionless load balancing virtual servers

September 14, 2021

When the Citrix ADC appliance performs load balancing, it creates and maintains sessions between clients and servers. The maintenance of session information places a significant load on the appliance

resources, and sessions might not be needed in scenarios such as a direct server return (DSR) setup and the load balancing of intrusion detection systems (IDS). To avoid creating sessions when they are not necessary, you can configure a virtual server on the appliance for sessionless load balancing. In sessionless load balancing, the appliance carries out load balancing on a per-packet basis.

Sessionless load balancing can operate in MAC-based forwarding mode or IP-based forwarding mode.

For MAC-based forwarding, the IP address of the sessionless virtual server must be specified on all the physical servers to which the traffic is forwarded.

For IP-based forwarding in sessionless load balancing, the IP address and port of the virtual server need not be specified on the physical servers, because this information is included in the forwarded packets. When forwarding a packet from the client to the physical server, the appliance leaves client details such as IP address and port unchanged and adds the IP address and port of the destination.

Supported setup

Citrix ADC sessionless load balancing supports the following service types and load balancing methods:

Service Types

- ANY for MAC-based redirection
- ANY, DNS, and UDP for IP-based redirection

Load Balancing Methods

- Round Robin
- Least Bandwidth
- LRTM (Least response time method)
- Source IP Hash
- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port Hash
- Custom Load

Limitations

Sessionless load balancing has the following limitations:

- The appliance must be deployed in two-arm mode.
- A service must be bound to only one virtual server.

- Sessionless load balancing is not supported for service groups.
- Sessionless load balancing is not supported for domain based services (DBS services).
- Sessionless load balancing in the IP mode is not supported for a virtual server that is configured as a backup to a primary virtual server.
- You cannot enable spillover mode.
- For all the services bound to a sessionless load balancing virtual server, the Use Source IP (USIP) option must be enabled.
- For a wildcard virtual server or service, the destination IP address is not changed.

Note:

- While configuring a virtual server for sessionless load balancing, explicitly specify a supported load balancing method. The default method, Least Connection, cannot be used for sessionless load balancing.
- To configure sessionless load balancing in MAC-based redirection mode on a virtual server, the MAC-based forwarding option must be enabled on the Citrix ADC appliance.

To add a sessionless virtual server by using the CLI

At the command prompt, type the following commands to add a sessionless virtual server and verify the configuration:

```
add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -m <
   redirectionMode> -sessionless <(ENABLED|DISABLED)> -lbMethod <
   load_balancing_method>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
 add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -
    lbMethod roundrobin -m ip
      Done
 show lb vserver sesslessv1
      sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
      State: DOWN
      ...
      Effective State: DOWN
      Client Idle Timeout: 120 sec
      Down state flush: ENABLED
      ...
      Persistence: NONE
      Sessionless LB: ENABLED
```



```
13        Connection Failover: DISABLED
14        L2Conn: OFF
15        1) Policy : cmp_text Priority:8680 Inherited
16        2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
17  <!--NeedCopy-->
```

To configure sessionless load balancing on an existing virtual server

At the command prompt, type:

```
1  set lb vserver <name>@ -m <redirectionMode> -sessionless <(ENABLED|
       DISABLED)> -lbMethod <load_balancing_method>
2  <!--NeedCopy-->
```

Example

```
1  set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
2    Done
3  <!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which the `-m MAC` option is enabled, you must bind a non-user monitor.

To configure a sessionless virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open the virtual server, and in Advanced Settings, click Traffic Settings, and then select Sessionless Load Balancing.

Redirect HTTP requests to a cache

September 14, 2021

The Citrix ADC cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the impact of responding to HTTP requests and improve your website performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the Citrix ADC appliance sends cacheable HTTP requests to the cache, and non-cacheable HTTP requests to the origin Web server.

To configure cache redirection on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -cacheable <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -cacheable yes
<!--NeedCopy-->
```

To configure cache redirection on a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Cacheable.

Direct requests according to priority

September 16, 2021

> **Warning:**
>
> Priority Queuing (PQ) deprecated from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use the AppQoE feature. However, Priority Queuing (PQ) is removed and no longer available on the Citrix ADC appliance release 13.1 onwards. For more information, see AppQoE topic.

The Citrix ADC appliance supports prioritization of client requests with its priority queuing feature. This feature allows you to designate certain requests, such as those from important clients, as priority requests and sends them to the “front of the line,” so that the appliance responds to them first. This allows you to provide uninterrupted service to those clients through demand surges or DDoS attacks on your website.

To configure priority queuing on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -pq <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -pq yes
<!--NeedCopy-->
```

To configure priority queuing on a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Priority Queuing.

Note: Configure priority queuing globally for it to function correctly.

Direct requests to a custom webpage

September 16, 2021

> **Warning:**
>
> SureConnect is deprecated from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use the AppQoE feature. However, SureConnect is removed and no longer available on the Citrix ADC appliance release 13.1 onwards. For more information, see AppQoE topic.

The Citrix ADC appliance provides the SureConnect option to ensure that web applications respond despite delays caused by limited server capacity or processing speed. SureConnect does this by displaying an alternative webpage of your choice when the server that hosts the primary webpage is either unavailable or responding slowly.

To configure SureConnect on a virtual server, you must first configure the alternative content. For information about configuring a SureConnect website, see SureConnect. After you configure the website, enable SureConnect on the load balancing virtual server to put your SureConnect custom webpage in use.

Note: For SureConnect to function correctly, you must configure it globally.

To enable SureConnect on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -sc <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -sc yes
<!--NeedCopy-->
```

To enable SureConnect on a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select SureConnect.

Enable cleanup of virtual server connections

September 14, 2021

Under certain conditions, you can configure the downStateFlush setting to immediately terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes and health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN.

A virtual server is marked DOWN only when all services bound to it are marked DOWN. When a virtual server goes DOWN, it terminates all connections, either immediately or after allowing existing connections to complete.

Do not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with one service bound to it, Service-TCP-1. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-TCP-1	State of connections
E	E	Both client and server connections are terminated.
E	D	For some service types, such as TCP, for which the Citrix ADC appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only client connections are terminated.
D	E	For some service types, such as TCP, for which the Citrix ADC appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown of a service, see Graceful Shutdown of Services.

To configure the down state flush setting on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -downStateFlush <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
<!--NeedCopy-->
```

To configure the down state flush setting on a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Down State Flush.

Rewrite ports and protocols for HTTP redirection

September 14, 2021

Virtual servers and the services that are bound to them might use different ports. When a service responds to an HTTP connection with a redirect, you might need to configure the Citrix ADC appliance to modify the port and the protocol to make sure that the redirection goes through successfully. You do this by enabling and configuring the redirectPortRewrite setting.

This setting affects only HTTP and HTTPS traffic. If this setting is enabled on a virtual server, the virtual server rewrites the port on redirects, replacing the port used by the service with the port used by the virtual server.

If the virtual server or service is of type SSL, you must enable SSL redirect on the virtual server or service. If both the virtual server and service are of type SSL, enable SSL redirect on the virtual server.

The redirectPortRewrite setting can be used in the following scenarios:

- The virtual server is of type HTTP and the services are of type SSL.
- The virtual server is of type SSL and the services are of type HTTP.

- The virtual server is of type HTTP and the services are of type HTTP.
- The virtual server is of type SSL and the services are of type SSL.

Scenario 1: The virtual server is of type HTTP and services are of type SSL. SSL redirect, and optionally port rewrite, is enabled on the service. If port rewrite is enabled, the port of HTTPS URLs is rewritten. HTTP URLs from the server are sent as is to the client.

Only SSL redirect is enabled. The virtual server can be configured on any port. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL redirect and port rewrite are enabled. The virtual server is configured on port 80. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

SSL redirect and port rewrite are enabled. Virtual server is configured on port 8080. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	http://domain.com:8080/
https://domain.com:444/	http://domain.com:8080/

Scenario 2: The virtual server is of type SSL and services are of type HTTP. If port rewrite is enabled,

only the port of HTTP URLs is rewritten. HTTPS URLs from the server are sent as is to the client.

SSL redirect is enabled on the virtual server. The virtual server can be configured on any port. See the following table.

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 443. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL redirect and port rewrite are enabled. The virtual server is configured on port 444. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	https://domain.com:444/
http://domain.com:8080/	https://domain.com:444/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

Scenario 3: The virtual server and service are of type HTTP. Port rewrite must be enabled on the virtual server. Only the port of HTTP URLs is rewritten. HTTPS URLs from the server are sent as is to the client.

The virtual server is configured on port 80. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

The virtual server is configured on port 8080. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com:8080/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

Scenario 4: The virtual server and service are of type SSL. If port rewrite is enabled, only the port of HTTPS URLs is rewritten. HTTP URLs from the server are sent as is to the client.

SSL redirect is enabled on the virtual server. The virtual server can be configured on any port. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 443. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/

Redirect URL from the Server	Redirect URL sent to the Client
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 444. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com:444/
https://domain.com:445/	https://domain.com:444/

To configure HTTP redirection on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -redirectPortRewrite (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

To configure HTTP redirection on a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open the virtual server, and in the Advanced Settings pane, click Traffic Settings, and then select Rewrite.

To configure SSL Redirect on an SSL virtual server or service by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)

```



```
set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled

set ssl service service-SSL-1 -sslRedirect enabled
<!--NeedCopy-->
```

To configure SSL redirection and SSL port rewrite on an SSL virtual server or service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click SSL Parameters, and select SSL Redirect.

Insert IP address and port of a virtual server in the request header

September 14, 2021

If you have multiple virtual servers that communicate with different applications on the same service, you must do the following:

Configure the Citrix ADC appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, then you must configure the required header tag on both virtual servers.

Note: This option is not supported for wild card virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the CLI

At the command prompt, type:

```
set lb vserver <name> -insertVserverIPPort <insertVserverIPPort> [<
    vipHeader>]
```



```
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
<!--NeedCopy-->
```

To insert the IP address and port of the virtual server in the client requests by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open the virtual server, and in the Advanced Settings pane, click **Traffic Settings**, and then select Virtual Server IP Port Insertion and specify a virtual server IP port header.

Use a specified source IP for back-end communication

October 27, 2021

For communication with the physical servers or other peer devices, the Citrix ADC appliance uses an IP address owned by it as the source IP address. The Citrix ADC appliance maintains a pool of its IP addresses, and dynamically selects an IP address while connecting with a server. Depending on the subnet in which the physical server is placed, the appliance decides which IP address to use. This address pool is used for sending traffic and monitor probes.

In many situations, you might want the appliance to use a specific IP address or any IP address from a specific set of IP addresses for back-end communications. The following are a few examples:

- A server can distinguish monitor probes from traffic if the source IP address used for monitor probes belongs to a specific set.
- To improve server security, a server might be configured to respond to requests from a specific set of IP addresses or, sometimes, from a single specific IP address. In such a case, the appliance can use only the IP addresses accepted by the server as the source IP address.
- The appliance can manage its internal connections efficiently if it can distribute its IP addresses into IP sets and use an address from a set only for connecting to a specific service.

To configure the appliance to use a specified source IP address, create net profiles (network profiles) and configure the appliance entities to use the profile. A net profile can be bound to load balancing or content switching virtual servers, Citrix Gateway VPN virtual servers, services, service groups, or monitors. A net profile has Citrix ADC owned IP addresses (SNIPs and VIPs) that can be used as the source IP address. It can be a single IP address or a set of IP addresses, referred to as an IP set. If a

net profile has an IP set, the appliance dynamically selects an IP address from the IP set at the time of connection. If a profile has a single IP address, the same IP address is used as the source IP.

If a net profile is bound to a load balancing or content switching virtual server, the profile is used for sending traffic to all the services bound to it. If a net profile is bound to a service group, the appliance uses the profile for all the members of the service group. If a net profile is bound to a monitor, the appliance uses the profile for all the probes sent from the monitor.

> **Note:**
>
> - When a Citrix ADC appliance uses a VIP address to communicate with a server, it uses session entries to identify whether the traffic destined to the VIP address is a response from a server or a request from a client.
> - You can bind a net profile to Citrix Gateway VPN virtual servers. However, you need to note some points when binding a net profile. For more information, see Points to note when binding a net profile to VPN virtual server.
> - The net profile IPs bound to a service or service group are not only used for sending traffic towards the corresponding back-end servers, but also for the DNS requests that are triggered by any unresolved back-end FQDN.

Usage of a net profile for sending traffic

If the Use Source IP Address (USIP) option is enabled, the appliance uses the IP address of the client and ignores all the net profiles. If the USIP option is not enabled, the appliance selects the source IP in the following manner:

- If there is no net profile on the virtual server or the service/service group, the appliance uses the default method.
- If there is a net profile only on the service/service group, the appliance uses that net profile.
- If there is a net profile only on the virtual server, the appliance uses the net profile.
- If there is a net profile both on the virtual server and service/service group, the appliance uses the net profile bound to the service/service group.

Usage of a net profile for sending monitor probes:

For monitor probes, the appliance selects the source IP in the following manner:

- If there is a net profile bound to the monitor, the appliance uses the net profile of the monitor. It ignores the net profiles bound to the virtual server or service/service group.
- If there is no net profile bound to the monitor,
  - If there is a net profile on the service/service group, the appliance uses the net profile of the service/service group.
  - If there is no net profile even on the service/service group, the appliance uses the default method of selecting a source IP.

Note: If there is no net profile bound to a service, the appliance looks for a net profile on the service group if the service is bound to a service group.

To use a specified source IP address for communication, go through the following steps:

1. Create IP sets from the pool of SNIPs and VIPs owned by the Citrix ADC appliance. An IP set can consist of both SNIP and VIP addresses. For instructions, see Creating IP Sets.
2. Create net profiles. For instructions, see Creating a Net Profile.
3. Bind the net profiles to the appliance entities. For instructions, see Binding a Net Profile to a Citrix ADC Entity.

> **Note:**
>
> - A net profile can have only the IP addresses specified as SNIP and VIP on the Citrix ADC appliance.
> - Source IP persistence is not honored for Citrix ADC initiated packets.

Manage net profiles

A net profile (or network profile) contains an IP address or an IP set. During communication with physical servers or peers, the Citrix ADC appliance uses the addresses specified in the profile as the source IP address.

- For instructions on creating a network profile, see Creating a Network Profile.
- For instructions on binding a network profile to a Citrix ADC entity, see Binding a Net Profile to a Citrix ADC Entity.

Create an IP set

An IP set is a set of IP addresses, which are configured on the Citrix ADC appliance as Subnet IP addresses (SNIPs) or Virtual IP addresses (VIPs). An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it. To create an IP set, add an IP set, and bind Citrix ADC owned IP addresses to it. SNIP addresses and VIP addresses can be present in the same IP set.

To create an IP set by using the CLI

At the command prompt, type the following commands:

```
add ipset <name>

bind ipset <name> <IPAddress>
<!--NeedCopy-->
```


Or

```
bind ipset <name> <IPAddress>

show ipset [<name>]
<!--NeedCopy-->
```

The preceding command shows the names of all the IP sets on the appliance if you do not pass any name. It shows the IP addresses bound to the specified IP set if you pass a name.

Examples

```
1.
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done

2.
 > add ipset testnwipset
 Done
> bind ipset testnwipset 21.21.21.[21-25]
 IPAddress "21.21.21.21" bound
 IPAddress "21.21.21.22" bound
 IPAddress "21.21.21.23" bound
 IPAddress "21.21.21.24" bound
 IPAddress "21.21.21.25" bound
 Done

3.
 > bind ipset skpipset 11.11.11.101
 ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address
   owned by the Citrix ADC appliance]
 > add ns ip 11.11.11.101 255.255.255.0 -type SNIP
 ip "11.11.11.101" added
 Done
 > bind ipset skpipset 11.11.11.101
 IPAddress "11.11.11.101" bound
 Done
4.
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
```



```
3) Name: ipset-3
4) Name: skpnewipset
 Done

5.
> sh ipset skpnewipset
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
 Done
<!--NeedCopy-->
```

To create an IP set by using the GUI

Navigate to **System > Network > IP Sets**, and create an IP set.

Create a net profile

A net profile (network profile) consists of one or more SNIP or VIP addresses of the Citrix ADC appliance.

To create a net profile by using the CLI

At the command prompt, type:

```
add netprofile <name> [-srcIp <srcIpVal>]
<!--NeedCopy-->
```

If the srcIpVal is not provided in this command, it can be provided later by using the `set netprofile` command.

Examples

```
add netprofile skpnetprofile1 -srcIp 21.21.20.1
Done

add netprofile baksnp -srcIp bakipset
Done

set netprofile yahnp -srcIp 12.12.23.1
```



```
Done

set netprofile citkbnp -srcIp citkbipset
Done
<!--NeedCopy-->
```

Bind a net profile to a Citrix ADC entity

A net profile can be bound to a load balancing virtual server, service, service group, or a monitor.

Note: You can bind a net profile at the time of creating a Citrix ADC entity or bind it to an existing entity.

To bind a net profile to a server by using the command line interface

You can bind a net profile to load balancing virtual servers and content switching virtual servers. Specify the appropriate virtual server.

At the command prompt, type:

```
set lb vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Or

```
set cs vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Examples

```
set lb vserver skpnwvs1 -netProfile gntnp
 Done
set cs vserver mmdcsv -netProfile mmdnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Profiles**, and set a net profile.

To bind a net profile to a service by using the CLI

At the command prompt, type:

```
set service <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Example

```
set service brnssvc1 -netProfile brnsnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, click **Profiles**, and set a net profile.

To bind a net profile to a service group by using the CLI

At the command prompt, type:

```
set servicegroup <serviceGroupName> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Example

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a service group by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Service Groups**, and open a service group.
2. In Advanced Settings, click **Profiles**, and set a net profile.

To bind a net profile to a monitor by using the CLI

At the command prompt, type:

```
set monitor <monitor_name> -netProfile <net_profile_name>
```


Example

```
set monitor brnsecvmon1 -netProfile brnsmonnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a monitor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Open a monitor, and set the net profile.

Set a time-out value for idle client connections

September 14, 2021

You can configure a virtual server to terminate any idle client connections after a configured time-out period (in seconds) elapses. When you configure this setting, the Citrix ADC appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection. By default, the client idle time-out value is set to 180 seconds.

To set a time-out value for idle client connections by using the CLI

At the command prompt, type:

```
set lb vserver <name> -cltTimeout <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -cltTimeout 100
<!--NeedCopy-->
```

To set a time-out value for idle client connections by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In **Advanced Settings,** click **Traffic Settings**, and set the client idle time-out value in seconds.

Manage RTSP connections

September 14, 2021

The Citrix ADC appliance can use either of two topologies—NAT-on mode or NAT-off mode—to load balance RTSP servers. In NAT-on mode, Network Address Translation (NAT) is enabled and configured on the appliance. RTSP requests and responses both pass through the appliance. You must therefore configure the appliance to perform network address translation (NAT) to identify the data connection.

For more information about enabling and configuring NAT, see IP Addressing.

In NAT-off mode, NAT is not enabled and configured. The appliance receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. The load balanced RTSP servers send their responses directly to the client, bypassing the appliance. You must therefore configure the appliance to use Direct Server Return (DSR) mode, and assign publicly accessible FQDNs in DNS to your load balanced RTSP servers.

For more information about enabling and configuring DSR mode, see Configuring Load Balancing in Direct Server Return Mode. For more information about configuring DNS, see Domain Name System. In either case, when you configure RTSP load balancing, you must also configure rtspNat to match the topology of your load balancing setup.

To configure RTSP NAT by using the CLI

At the command prompt, type:

```
set lb vserver <name>  -RTSPNAT <ValueOfRTSPNAT>
<!--NeedCopy-->
```

Example:

```
set lb vserver vserver-LB-1  -RTSPNAT ON
<!--NeedCopy-->
```

To configure RTSP NAT by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server of type RTSP.
2. In Advanced Settings, click **Traffic Settings**, and select **RTSP Natting**.

Manage client traffic based on traffic rate

September 14, 2021

You can monitor the rate of traffic that flows through load balancing virtual servers and control the behavior of the Citrix ADC appliance based on the traffic rate. For example:

- Throttle the traffic flow if it is too high.
- Cache information based on the traffic rate.
- If the traffic rate is too high, redirect excess traffic to a different load balancing virtual server.
- Apply rate-based monitoring to HTTP and Domain Name System (DNS) requests.

For more information on rate-based policies, see Rate Limiting.

Identify a connection with layer 2 parameters

September 14, 2021

Generally, to identify a connection, the Citrix ADC appliance uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.

Enabling the L2Conn parameter for a load balancing virtual server allows multiple TCP and non-TCP connections with the same 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) to co-exist on the Citrix ADC appliance. The appliance uses both the 4-tuple and the Layer 2 parameters to identify TCP and non-TCP connections.

You can enable the L2Conn option in the following scenarios:

- Multiple VLANs are configured on the Citrix ADC appliance, and a firewall is set up for each VLAN.
- You want the traffic originating from the servers in one VLAN and bound for a virtual server in another VLAN to pass through the firewalls configured for both VLANs.

Therefore, when an nCore Citrix ADC appliance on which the l2Conn parameter is set for one or more load balancing virtual servers is downgraded to a Classic build or to an nCore build that does not support the l2Conn parameter, the load balancing configurations that use the l2Conn parameter become ineffective.

To configure the L2 connection option by using the CLI

At the command prompt, type:


```
add lb vserver <name> <serviceType> <IPAddress>@ <port> -l2Conn ON
<!--NeedCopy-->
```

Example

```
 add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
<!--NeedCopy-->
```

To configure the L2 connection option by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Layer 2 Parameters.

Configure the Prefer Direct Route option

September 14, 2021

On a wildcard load balancing virtual server if you explicitly configure a route to a destination, by default, the Citrix ADC appliance forwards traffic according to the configured route. If you want the appliance to not look up for the configured route, you can set the Prefer Direct Route option to NO.

If a device is directly connected to a Citrix ADC appliance, the appliance directly forwards traffic to the device. For example, if the destination of a packet is a firewall, the packet need not be routed through another firewall. However, sometimes, you might want the traffic to go through the firewall even if the device is directly connected to it. In such cases, you can set the Prefer Direct Route Option to NO.

Note: The preferDirectRoute setting is applicable to all the wildcard virtual servers on the Citrix ADC appliance.

To set the Prefer Direct Route option by using the CLI

At the command prompt, type:

```
set lb parameter -preferDirectRoute (YES | NO)
<!--NeedCopy-->
```

Example:

```
set lb parameter -preferDirectRoute YES
<!--NeedCopy-->
```


To set the Prefer Direct Route option by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Configure Load Balancing parameters**.
2. Select Prefer Direct Route.

Use a source port from a specified port range for back-end communication

September 14, 2021

By default, for configurations with USIP option disabled or with USIP and use proxy port options enabled, the Citrix ADC appliance communicates to the servers from a random source port (greater than 1024).

The appliance supports using a source port from a specified port range for communicating to the servers. One of the use cases of this feature is for servers that are configured to identify received traffic belonging to a specific set based on the source port for logging and monitoring purposes. For example, identifying internal and external traffic for logging purpose.

Configuring the Citrix ADC appliance to use a source port from a port range for communicating to the servers consists of the following tasks:

- **Create a net profile and set the source port range parameter**. A source port range parameter specifies one or more port ranges. The appliance randomly selects one of the free ports from the specified port ranges and used it as the source port for each connection to servers.
- **Bind the net profile to load balancing virtual servers, services, or service groups**: A net profile with source port range setting can be bound to a virtual server, service, or a service group of a load balancing configuration. For a connection to a virtual server, the appliance randomly selects one of the free ports from the specified port ranges of a net profile and use this port as the source port for connecting to one of the bound servers.

To specify a source port range or ranges by using the CLI

At the command prompt, type:

```
bind netProfile <name> (-srcPortRange <int[-int]> ...)

show netprofile <name>
<!--NeedCopy-->
```


To specify a source port range or ranges by using the GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the **Source Port Range** parameter while adding or modifying NetProfiles.

Sample Configuration

In the following sample configuration, net profile PARTIAL-NAT-1 has partial NAT settings and is bound to load balancing virtual server LBVS-1, which is of type ANY. For packets received on LBVS-1 from 192.0.0.0/8, the Citrix ADC appliance translates the last octet of the packet's source IP address to 100. For example, a packet with source IP address 192.0.2.30 received on LBVS-1, the Citrix ADC appliance translates the source IP address to 100.0.2.30 before sending it one of the bound servers.

```
1  ```
2  > add netprofile CUSTOM-SRCPORT-NP-1
3  Done
4  > bind netprofile CUSTOM-SRCPRT-NP-1  - srcportrange 2000-3000
5
6  Done
7  > bind netprofile CUSTOM-SRCPRT-NP-1  - srcportrange 5000-6000
8
9  Done
10 > add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
11
12 Done
13 <!--NeedCopy--> ```
```

Configure source IP persistency for back-end communication

September 14, 2021

By default, for a load balancing configuration with the USIP option disabled and a net profile bound to a virtual server or services or service groups, the Citrix ADC appliance uses the round robin algorithm to select an IP address from the net profile for communicating with the servers. Because of this selection method, the IP address selected can be different for different sessions of a specific client.

Some situations require that the Citrix ADC appliance route all of a specific client's traffic from the same IP address when sending the traffic to servers. The servers can then, for example, identify traffic belonging to a specific set for logging and monitoring purposes.

The source IP persistency option of a net profile enables the Citrix ADC appliance to use the same address, specified in the net profile, to communicate with servers about all sessions initiated from a

specific client to a virtual server.

To enable source IP persistency in a net profile by using the CLI

To enable source IP persistency while adding a net profile, at the command prompt, type:

```
add netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable source IP persistency in an existing net profile, at the command prompt, type:

```
set netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable source IP persistency in a net profile by using the GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Select **Source IP Persistency** while adding or modifying a net profile.

Example

In the following sample configuration, net profile NETPROFILE-IPPRSTNCY-1 has the source IP persistency option enabled and is bound to load balancing virtual server LBVS-1.

The Citrix ADC appliance always uses the same IP address (in this example, 192.0.2.11) to communicate with servers bound to LBVS-1, for all sessions initiated from a specific client to the virtual server.

```
```
> add ipset IPSET-1

Done
> bind ipset IPSET-1 192.0.2.[11-15]
 IPAddress "192.0.2.11" bound
 IPAddress "192.0.2.12" bound
 IPAddress "192.0.2.13" bound
 IPAddress "192.0.2.14" bound
 IPAddress "192.0.2.15" bound
 Done
```

 
```

```
> add netprofile NETPROFILE-IPPRSTNCY-1 -srcIp IPSET-1 -
   srcippersistency ENABLED

Done
> set lb vserver LBVS-1 -netprofile NETPROFILE-IPPRSTNCY-1

 Done
<!--NeedCopy--> ```
```

Use IPv6 link local addresses on the server side of a load balancing setup

September 14, 2021

IPv6 link local address is supported for services, service groups, and servers of a load balancing configuration. You can specify a link local IPv6 address along with the associated VLAN ID in services, service groups, and servers configurations. The Citrix ADC appliance uses the link local SNIP6 address from the same VLAN as specified in the services, service groups, and servers configurations to communicate with them.

A link local IPv6 address and the associated VLAN ID are specified in the following format in services, service groups, and servers configurations: `<IPv6_Addrs>%<vlan_id>`

For example, `fe80:123:4567::a%2048:`, `fe80:123:4567::a` is the link local address AND 2048 is the VLAN ID.

```
> add service SERVICE-1 fe80:123:4567::a%2048 HTTP 80

Done
> bind servicegroup SERVICE-GROUP-1 fe80::1%24 80

Done
> add server SERVER-1 fe80:b:c:d::e:f:a/64%1028

Done
```

Advanced load balancing settings

September 14, 2021

In addition to configuring virtual servers, you can configure advanced settings for services.

To configure advanced load balancing settings, see the following sections:

Gradually step up the load on a new service with virtual server–level slow start

September 14, 2021

You can configure the Citrix ADC appliance to gradually increase the load on a service (the number of requests that the service receives per second) immediately after the service is either added to a load balancing configuration or has a state change from DOWN to UP (in this document, the term "new service" is used for both situations). You can either increase the load manually with load values and intervals of your choice (manual slow start) or configure the appliance to increase the load at a specified interval (automated slow start) until the service is receiving as many requests as the other

services in the configuration. During the ramp-up period for the new service, the appliance uses the configured load balancing method.

This functionality is not available globally. It has to be configured for each virtual server. The functionality is available only for virtual servers that use one of the following load balancing methods:

- Round robin
- Least connection
- Least response time
- Least bandwidth
- Least packets
- LRTM (Least Response Time Method)
- Custom load

For this functionality, you need to set the following parameters:

- The new service request rate, which is the amount by which to increase the number or percentage of requests sent to a new service each time the rate is incremented. That is, you specify the size of the increment in terms of either the number of requests per second or the percentage of the load being borne, at the time, by the existing services. If this value is set to 0 (zero), slow start is not performed on new services.

  Note: In an automated slow start mode, the final increment is smaller than the specified value if the specified value would place a heavier load on the new service than on the other services.

- The increment interval, in seconds. If this value is set to 0 (zero), the load is not incremented automatically. You have to increment it manually.

With an automated slow start, a service is taken out of the slow start phase when one of the following conditions applies:

- The actual request rate is less than the new service request rate.
- The service does not receive traffic for three successive increment intervals.
- The request rate has been incremented 200 times.
- The percentage of traffic that the new service must receive is greater than or equal to 100.

With manual slow start, the service remains in the slow start phase until you take it out of that phase.

Manual slow start

If you want to manually increase the load on a new service, do not specify an increment interval for the load balancing virtual server. Specify only the new service request rate and the units. With no interval specified, the appliance does not increment the load periodically. It maintains the load on the new service at the value specified by the combination of the new service request rate and units until you manually modify either parameter. For example, if you set the new service request rate and

unit parameters to 25 and “per second,” respectively, the appliance maintains the load on the new service at 25 requests per second until you change either parameter. When you want the new service to exit the slow start mode and receive as many requests as the existing services, set the new service request rate parameter to 0.

As an example, assume that you are using a virtual server to load balance 2 services, Service1 and Service2, in round robin mode. Further assume that the virtual server is receiving 240 requests per second, and that it is distributing the load evenly across the services. When a new service, Service3, is added to the configuration, you might want to increase the load on it manually through values of 10, 20, and 40 requests per second before sending it its full share of the load. The following table shows the values to which you set the three parameters.

Table 1. Parameter Values

Parameter	Value
Interval in seconds	0
New service request rate	10, 20, 40, and 0, at intervals that you choose
Units for the new service request rate	Requests per second

When you set the new service request rate parameter to 0, Service3 is no longer considered a new service, and receives its full share of the load.

Assume that you add another service, Service4, during the ramp-up period for Service3. In this example, Service4 is added when the new service request rate parameter is set to 40. Therefore, Service4 begins receiving 40 requests per second.

The following table shows the load distribution on the services during the period described in this example.

Table 2. Load Distribution on Services when Manually Stepping Up the Load

	new service request rate = 10 req/sec (Service3added)	new service request rate = 20 req/sec	new service request rate = 40 req/sec (Service4added)	new service request rate = 0 req/sec (new services exit slow start mode)
Service1	115	110	80	60
Service2	115	110	80	60
Service3	10	20	40	60
Service4	-	-	40	60

	new service request rate = 10 req/sec (Service3added)	new service request rate = 20 req/sec	new service request rate = 40 req/sec (Service4added)	new service request rate = 0 req/sec (new services exit slow start mode)
Total req/sec (load on the virtual server)	240	240	240	240

Automated slow start

If you want the appliance to increase the load on a new service automatically at specified intervals until the service can be considered capable of handling its full share of the load, set the new service request rate parameter, the units parameter, and the increment interval. When all the parameters are set to values other than 0, the appliance increments the load on a new service by the value of the new service request rate, at the specified interval, until the service is receiving its full share of the load.

As an example, assume that four services, Service1, Service2, Service3, and Service4, are bound to a load balancing virtual server, vserver1. Further assume that vserver1 receives 100 requests per second, and that it distributes the load evenly across the services (25 requests per second per service). When you add a fifth service, Service5, to the configuration, you might want the appliance to send the new service 4 requests per second for the first 10 seconds, 8 requests per second for the next 10 seconds, and so on, until it is receiving 20 requests per second. For this requirement, the following table shows the values to which you set the three parameters:

Table 3. Parameter Values

Parameter	**Value**
Interval in seconds	10
Increment value	4
Units for the new service request rate	Requests per second

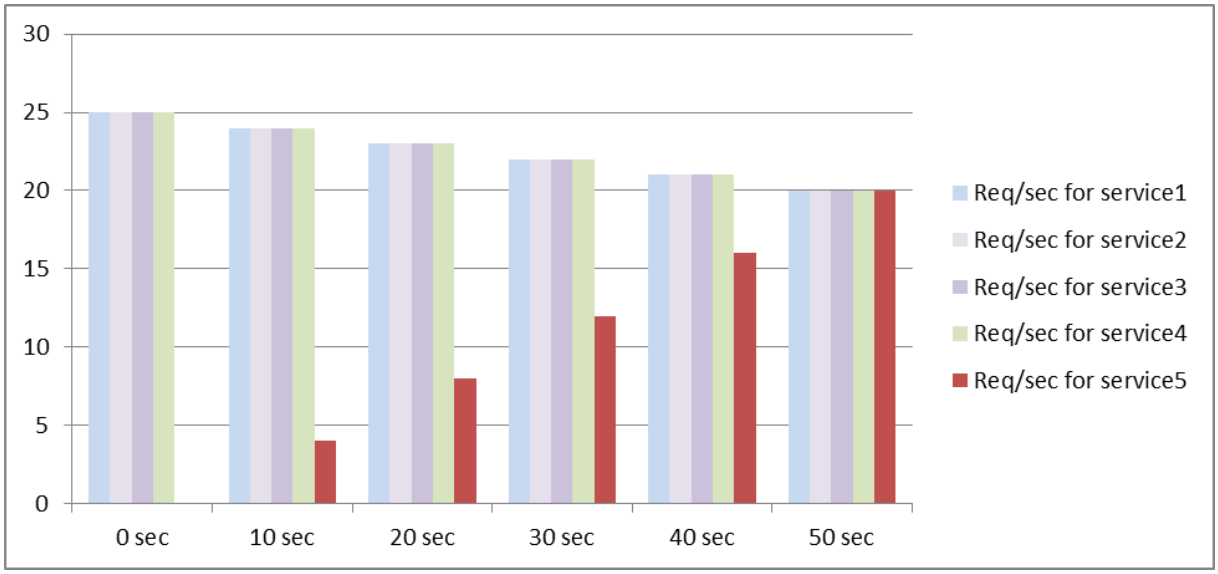
With this configuration, the new service begins receiving as many requests as the existing services 50 seconds after it is added or its state has changed from DOWN to UP. During each interval in this period, the appliance distributes to the existing servers the excess of requests that would have been sent to the new service in the absence of stepwise increments. For example, in the absence of stepwise increments, each service, including Service5, would have received 20 requests each per second. With stepwise increments, during the first 10 seconds, when Service5 receives only 4 requests per second,

the appliance distributes the excess of 16 requests per second to the existing services, resulting in the distribution pattern shown in the following table and figure over the 50-second period. After the 50-second period, Service5 is no longer considered a new service, and it receives its normal share of traffic.

Table 4. Load Distribution Pattern on All Services for the 50-second Period Immediately after Service5 is Added

	0 sec	10 sec	20 sec	30 sec	40 sec	50 sec
Req/sec forService1	25	24	23	22	21	20
Req/sec forService2	25	24	23	22	21	20
Req/sec forService3	25	24	23	22	21	20
Req/sec forService4	25	24	23	22	21	20
Req/sec forService5	0	4	8	12	16	20
Total req/sec (load on the virtual server)	100	100	100	100	100	100

Figure 1. A Graph of the Load Distribution Pattern on All Services for the 50-second Period Immediately after Service5 is Added



An alternative requirement might be for the appliance to send Service5 25% of the load on the existing services in the first 5 seconds, 50% in the next 5 seconds, and so on, until it is receiving 20 requests per second. For this requirement, the following table shows the values to which you set the three parameters.

Table 5. Parameter Values

Parameter	Value
Interval in seconds	5
Increment value	25
Units for the new service request rate	Percent

With this configuration, the service begins receiving as many requests as the existing services 20 seconds after it is added or its state has changed from DOWN to UP. The traffic distribution during the ramp-up period for the new service is identical to the one described earlier, where the unit for the step increments was “requests per second.”

Set the slow start parameters

You set the slow start parameters by using either the `set lb vserver` or the `add lb vserver` command. The following command is for setting slow start parameters when adding a virtual server.

To configure stepwise load increments for a new service by using the command line interface

At the command prompt, type the following commands to configure stepwise increments in the load for a service and verify the configuration:

```
add lb vserver <name> <serviceType> <IPAddress> <port> [-
    newServiceRequest <positive_integer>] [<newServiceRequestUnit>] [-
    newServiceRequestIncrementInterval <positive_integer>]

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -
    newServiceRequestIncrementInterval 10
Done

show lb vserver BR_LB
BR_LB (192.0.2.33:80) - HTTP Type: ADDRESS
State: UP
...
...
New Service Startup Request Rate: 5 PER_SECOND, Increment Interval: 10
...
...
Done
<!--NeedCopy-->
```

To configure stepwise load increments for a new service by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select Method, and set the following slow start parameters:
   - New Service Startup Request Rate.
   - New Service Request Unit.
   - Increment Interval.

The no-monitor option for services

September 14, 2021

If you use an external system to perform health checks on the services and do not want the Citrix ADC appliance to monitor the health of a service, you can set the no-monitor option for the service. If you do so, the appliance does not send probes to check the health of the service but shows the service as UP. Even if the service goes DOWN, the appliance continues to send traffic from the client to the service as specified by the load balancing method.

The monitor can be in the ENABLED or DISABLED state when you set the no-monitor option, and when you remove the no-monitor option, the earlier state of the monitor is resumed.

You can set the no-monitor option for a service when creating the service. You can also set the no-monitor option on an existing service.

The following are the consequences of setting the no-monitor option:

- If a service for which you enabled the no-monitor option goes down, the appliance continues to show the service as UP and continues to forward traffic to the service. A persistent connection to the service can worsen the situation. In that case, or if many services shown as UP are actually DOWN, the system may fail. To avoid such a situation, when the external mechanism that monitors the services reports a service as DOWN, remove the service from the Citrix ADC configuration.
- If you configure the no-monitor option on a service, you cannot configure load balancing in the Direct Server Return (DSR) mode. For an existing service, if you set the no-monitor option, you cannot configure the DSR mode for the service.

To set the no-monitor option for a new service by using the CLI

At the command prompt, type the following commands to create a service with the health monitor option, and verify the configuration:

```
add service <serviceName> <IP | serverName> <serviceType> <port> -
   healthMonitor (YES|NO)
<!--NeedCopy-->
```

Example:

```
add service nomonsrvc 10.102.21.21 http 80 -healthMonitor no
Done

show service nomonsrvc
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
```



```
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done
<!--NeedCopy-->
```

To set the no-monitor option for an existing service by using the CLI

At the command prompt, type the following command to set the health monitor option:

```
set service <name> -healthMonitor (YES|NO)
<!--NeedCopy-->
```

Example:

```
By default, the state of a service and the state of the corresponding
   monitor are UP.
 >show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP


1) Monitor Name: http-ecv
 State: UP Weight: 1
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done

When the no-monitor option is set on a service, the state of the
   monitor changes to UNKNOWN.
set service LB-SVC1 -healthMonitor NO
Done

show service LB-SVC1
```



```
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010.
Time since last state change: 5 days, 18:55:48.710
Health monitoring: OFF

1) Monitor Name: http-ecv
   State: UNKNOWN Weight: 1
         Probes: 100028 Failed [Total: 0 Current: 0]
   Last response: Probe skipped - Health monitoring is turned off.
     Response Time: 0.0 millisec
  Done
When the no-monitor option is removed, the earlier state of the monitor
    is resumed.
> set service LB-SVC1 -healthMonitor YES
Done
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010
Time since last state change: 5 days, 18:57:47.880
1) Monitor Name: http-ecv
   State: UP Weight: 1
   Probes: 100029 Failed [Total: 0 Current: 0]
   Last response: Success - Pattern found in response.
   Response Time: 5.690 millisec
   Done
<!--NeedCopy-->
```

To set the no-monitor option for a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and clear Health Monitoring.

Protect applications on protected servers against traffic surges

September 14, 2021

The Citrix ADC appliance provides the surge protection option to maintain the capacity of a server or cache. The appliance regulates the flow of client requests to servers and controls the number of clients that can simultaneously access the servers. The appliance blocks any surges passed to the server, thus preventing overloading of the server.

For surge protection to function correctly, you must enable it globally. For more information about surge protection, see Surge Protection.

To set surge protection on the service by using the CLI

At the command prompt, type:

```
set service <name> -sp <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -sp ON
<!--NeedCopy-->
```

To set surge protection on the service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a source.
2. In Advanced Settings, select **Traffic Settings**, and select **Surge Protection**.

Enable cleanup of virtual server and service connections

September 14, 2021

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes or health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN. If a service or virtual server is marked DOWN, the server and client side connections must be flushed. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

Under certain conditions, you can configure the **downStateFlush** setting to immediately terminate existing connections when a service or a virtual server is marked DOWN. Do not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with one service bound to it, Service-1. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-1	State of Connections
E	E	Both client and server connections are terminated.
E	D	For some service types, such as TCP, for which the Citrix ADC appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only client connections are terminated.
D	E	For some service types, such as TCP, for which the Citrix ADC appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown of a service, see Graceful Shutdown of Services.

To set down state flush on the service by using the CLI

At the command prompt, type:

```
set service <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -downStateFlush enabled
<!--NeedCopy-->
```

To set down state flush on the service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **Down State Flush**.

To set down state flush on the virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

Example:

```
set lb vserver vsvr1 -downStateFlush enabled
<!--NeedCopy-->
```

To set down state flush on the virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **Traffic Settings**, and select **Down State Flush**.

Graceful shutdown of services

September 14, 2021

During scheduled network outages such as system upgrades or hardware maintenance, you may have to close or disable some services. You can later enable the service by using the "enable service <name> command.

To avoid disrupting established sessions, you can place a service in the Transition Out of Service (TROFS) state by doing one of the following:

- Adding a TROFS code or string to the monitor—Configure the server to send a specific code or string in response to a monitor probe.
- Explicitly disable the service and:
  - Set a delay (in seconds).
  - Enable graceful shutdown.

Adding a TROFS Code or String

If you bind only one monitor to a service, and the monitor is TROFS-enabled, it can place the service in the TROFS state on the basis of the server's response to a monitor probe. This response is compared with the value in the trofsCode parameter for an HTTP monitor or the trofsString parameter for an HTTP-ECV or TCP-ECV monitor. If the code matches, the service is placed in the TROFS state. In this state, it continues to honor the persistent connections.

If multiple monitors are bound to a service, the effective state of the service is calculated on the basis of the state of all the monitors that are bound to the service. Upon receiving a TROFS response, the state of the TROFS-enabled monitor is considered as UP for the purpose of this calculation. For more information about how a Citrix ADC appliance designates a service as UP, see Setting a Threshold Value for the Monitors Bound to a Service.

Important:

- You can bind multiple monitors to a service, but must not TROFS-enable more than one of them.
- You can convert a TROFS-enabled monitor to a monitor that is not TROFS-enabled, but not vice versa.

To configure a TROFS code or string in a monitor by using the command line interface

At the command prompt, type one of the following commands:

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>

add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>
```



```
4
5 add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
6 <!--NeedCopy-->
```

To modify the TROFS code or string by using the command line interface

At the command prompt, type one of the following commands:

```
1 set lb monitor <trofs monitorname> HTTP -trofscode <newcode>
2
3 set lb monitor <trofs monitorname> HTTP-ECV -trofsstring <new string>
4
5 set lb monitor <trofs monitorname> TCP-ECV -trofsstring <new string>
6 <!--NeedCopy-->
```

Note: You can use the set command only if a TROFS-enabled monitor was added earlier. You cannot use this command to set the TROFS code or string for a monitor that is not TROFS-enabled.

To configure a TROFS code or string in a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. On the Monitors pane, click Add, and do one of the following:
   - Select Type as HTTP, and specify a TROFS Code.
   - Select Type as HTTP-ECV or TCP-ECV, and specify a TROFS String.

Disabling a Service

Often, however, you cannot estimate the amount of time needed for all the connections to a service to complete the existing transactions. If a transaction is unfinished when the wait time expires, shutting down the service may result in data loss. In this case, you can specify graceful shutdown for the service, so that the service is disabled only when all the current active client connections are closed by either the server or the client. See the following table for behavior if you specify a wait time in addition to graceful shutdown.

Persistence is maintained according to the specified method even if you enable graceful shutdown. The system continues to serve all the persistent clients, including new connections from the clients, unless the service is marked DOWN during the graceful shutdown state as a result of the checks made by a monitor.

The following table describes the graceful shutdown options.

State	Results
Graceful shutdown is enabled and a wait time is specified.	Service is shut down after the last of the current active client connections is served, even if the wait time has not expired. The appliance checks the status of the connections once every second. If the wait time expires, any open sessions are closed.
Graceful shutdown is disabled and a wait time is specified.	Service is shut down only after the wait time expires, even if all established connections are served before expiration.
Graceful shutdown is enabled and no wait time is specified.	Service is shut down only after the last of the previously established connections is served, regardless of the time taken to serve the last connection.
Graceful shutdown is disabled and no wait time is specified.	No graceful shutdown. Service is shut down immediately after the disable option is chosen or the disable command is issued. (The default wait time is zero seconds.)

To terminate existing connections when a service or a virtual server is marked DOWN, you can use the Down State Flush option. For more information, see Enabling Cleanup of Virtual Server Connections.

To configure graceful shutdown for a service by using the command line interface

At the command prompt, type the following commands to shut down a service gracefully and verify the configuration:

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

Example:

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
```



```
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
1) Monitor Name: tcp-default
State: UP              Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN           Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
<!--NeedCopy-->
```

To configure graceful shutdown for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and from the Action list, click Disable. Enter a wait time, and select Graceful.

Enable or disable persistence session on TROFS services

September 14, 2021

You can set the trofsPersistence flag to specify whether a service in the transition out of service (TROFS) state must maintain persistent sessions. When a monitor is TROFS enabled, it can place a service in the TROFS state on the basis of the server's response to a monitor probe. This response

is compared with the value in the trofsCode parameter for an HTTP monitor or the trofsString parameter for an HTTP-ECV or TCP-ECV monitor. If the code matches, the service is placed in the TROFS state. In this state, it continues to honor the active client connections. In some cases, the honored active sessions might have to include persistent sessions. But in other cases, especially those involving long-lived persistence sessions or persistence methods such as custom server ID, honoring the persistent sessions can prevent the service from transitioning to the out-of-service state.

If you set the trofsPersistence flag to ENABLED, persistent sessions are honored. If you set it to DISABLED, they are not.

To set the trofsPersistence flag by using the command line interface

At the command prompt, type one of the following commands to set the `trofsPersistence` flag for a new virtual server or an existing virtual server, or to return the setting to its default value:

```
add lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

set lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

unset lb vserver <name> [-trofsPersistence]
<!--NeedCopy-->
```

Argument

trofsPersistence. Honor current active client connections and new requests on persistence sessions when the service is in TROFS state.

Possible values: ENABLED, DISABLED. Default: ENABLED.

Examples:

```
add lb vserver v1 http 10.102.217.42 80 -persistencetype SOURCEIP -
    trofsPersistence ENABLED

set lb vserver v1 -trofsPersistence DISABLED

unset lb vserver v1 -trofsPersistence
<!--NeedCopy-->
```


Direct requests to a custom webpage

September 14, 2021

> **Warning**
>
> SureConnect (SC) is deprecated from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use the AppQoE feature. For more information, see AppQoE.

For SureConnect to function correctly, you must set it globally. The Citrix ADC provides the SureConnect option to ensure the response from an application.
For more information about the SureConnect option, see Sure Connect.

To set SureConnect on the service by using the CLI

At the command prompt, type:

```
set service <name> -sc <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -sc ON
<!--NeedCopy-->
```

To set SureConnect on the service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select Traffic Settings, and select **Sure Connect**.

Enable access to services when down

September 14, 2021

You can enable access to a service when it is disabled or in a DOWN state by configuring the Citrix ADC appliance to use Layer 2 mode to bridge the packets sent to the service. Normally, when requests are forwarded to services that are DOWN, the request packets are dropped. When you enable the **Access Down** setting, however, these request packets are sent directly to the load balanced servers.

For more information about Layer 2 and Layer 3 modes, see IP Addressing.

For the appliance to bridge packets sent to the DOWN services, enable Layer 2 mode with the access-Down parameter.

To enable access down on a service by using the CLI

At the command prompt, type:

```
set service <name> -accessDown <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -accessDown YES
<!--NeedCopy-->
```

To enable access down on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **Access Down**.

Enable TCP buffering of responses

September 14, 2021

The Citrix ADC appliance provides a TCP buffering option that buffers only responses from the load balanced server. This enables the appliance to deliver server responses to the client at the maximum speed that the client can accept them. The appliance allocates from 0 through 4095 MB (MB) of memory for TCP buffering, and from 4 through 20480 kilobytes (KB) of memory per connection.

Note: TCP buffering set at the service level takes precedence over the global setting.
For more information about configuring TCP buffering globally, see TCP Buffering.

To enable TCP Buffering on a service by using the CLI

At the command prompt, type:

```
set service <name> -TCPB <Value>
<!--NeedCopy-->
```

Example:


```
set service Service-HTTP-1 -TCPB YES
<!--NeedCopy-->
```

To enable TCP Buffering on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **TCP Buffering**.

Enable compression

September 14, 2021

The Citrix ADC appliance provides a compression option to transparently compress HTML and text files by using a set of built-in compression policies. Compression reduces bandwidth requirements and can significantly improve server responsiveness in bandwidth-constrained setups. The compression policies are associated with services bound to the virtual server. The policies determine whether a response can be compressed and send compressible content to the appliance, which compresses it and sends it to the client.

Note: For compression to function correctly, you must enable it globally. For more information about configuring compression globally, see Compression.

To enable compression on a service by using the CLI

At the command prompt, type:

```
set service <name> -CMP <YES | NO>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -CMP YES
<!--NeedCopy-->
```

To enable compression on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **Compression**.

Enable external TCP health check for UDP virtual servers

September 14, 2021

In public clouds, you can use the Citrix ADC appliance as a second-tier load balancer when the native load balancer is used as a first tier. The native load balancer can be an application load balancer (ALB) or a network load balancer (NLB). Most of the public clouds do not support UDP health probes in their native load balancers. To monitor the health of the UDP application, public clouds recommend adding a TCP-based endpoint to your service. The endpoint reflects the health of the UDP application.

The Citrix ADC appliance supports the external TCP-based health check for a UDP virtual server. This feature introduces a TCP listener on the VIP of the virtual server and the configured port. The TCP listener reflects the status of the virtual server.

To enable an external TCP health check for UDP virtual servers by CLI

At the command prompt, type the following command to enable an external TCP health check with the tcpProbePort option:

```
1 add lb vserver <name> <serviceType> <IPAddress> <port> -tcpProbePort <
     tcpProbePort>
2 <!--NeedCopy-->
```

Example:

```
1 add lb vserver Vserver-UDP-1 UDP 10.102.29.60 80 tcpProbePort 5000
2 <!--NeedCopy-->
```

To enable an external TCP health check for UDP virtual servers by GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and then create a virtual server.
2. Click **Add** to create a virtual server.
3. In the **Basic Settings** pane, add the port number in the **TCP Probe Port** field.
4. Click **OK**.

Maintain client connection for multiple client requests

September 14, 2021

You can set the client keep-alive parameter to configure an HTTP or SSL service to keep a client connection to a website open across multiple client requests. If client keep-alive is enabled, even when the load balanced Web server closes a connection, the Citrix ADC appliance keeps the connection between the client and itself open. This setting allows services to serve multiple client requests on a single client connection.

If you do not enable this setting, the client opens a new connection for every request that it sends to the website. The client keep-alive setting saves the packet round trip time required to establish and close connections. This setting also reduces the time to complete each transaction. Client keep-alive can be enabled only on HTTP or SSL service types.

Client keep-alive set at the service level takes precedence over the global client keep-alive setting. For more information about client keep-alive, see Client Keep-Alive.

To enable the client keep-alive on a service by using the CLI

At the command prompt, type:

```
set service <name> -CKA <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -CKA YES
<!--NeedCopy-->
```

To enable the client keep-alive on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **Client Keep-Alive**.

Insert the IP address of the client in the request header

September 14, 2021

A Citrix ADC uses the subnet IP (SNIP) address to connect to the server. The server need not be aware of the client.

However, in some situations, the server needs to be aware of the client it has to serve. When you enable the client IP setting, the appliance inserts the client's IPv4 or IPv6 address while forwarding the requests to the server. The server inserts this client IP in the header of the responses. The server is thus aware of the client.

> **Note**: To insert multiple headers, you need to perform one of the following:
>
> - Add rewrite policies to check CLIENT.IS_SSL and insert appropriate header.
> - Bind the appropriate rewrite policy for each virtual server based on the type.

To insert the client IP address in the client request by using the CLI

At the command prompt, type:

```
set service <name> -CIP <Value> <cipHeader>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -CIP enabled X-Forwarded-For
<!--NeedCopy-->
```

To insert the client IP address in the client request by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and edit a service.
2. In the **Service Settings** pane, click the **edit icon**.
3. In the **Load Balancing Service** pane, select the **Insert Client IP Address** check box.

Retrieve location details from user IP address using geolocation database

September 14, 2021

> **Note** This feature is available from Citrix ADC release 12.1 build 50.x and later.

The Citrix ADC appliance can get user location details like continent, county, and city. For any public IP address from a geo location database. It is performed using the advanced policy infrastructure. The retrieved location details are then used in a rewrite action or a responder action for performing the following use cases.

- Insert an HTTP header with user location details (such as country, city information) when sending the client request to the back-end server.
- Add country name in the HTML page response for an invalid user.

The appliance can also log the location details using the audit logging mechanism.

Getting user location details using geolocation functions

The components interact as per the following:

1. User sends a client request from a particular geographic location.
2. The Citrix ADC appliance looks for the user IP address from the client request and retrieves the geo location details. The details include continent, country, region, city, ISP, organization, or custom details from a geolocation database.
3. Once the location details are retrieved, the appliance uses either a responder policy or a rewrite policy to evaluate the request.
4. In a rewrite policy, the appliance adds a header with the geo location details and sends it to the back-end server. For example, insert a custom HTTP header with country information.
5. In a responder policy, the appliance evaluates the HTTP request and based on policy evaluation, allows access to the users or redirects the user to an error page. It states the region from where they are accessing the application does not have access.

Setting up geolocation database

As a pre-requisite, you must have a geolocation database to run on the Citrix ADC appliance. The geolocation database files are available with Citrix ADC firmware. To download the database files from a vendor, convert it into Citrix ADC format and import it into your appliance.
For more information about the geolocation database, see Add a location file to create static proximity database topic.

Geolocation functions

The following table gives a list of geolocation functions that retrieves location details of any public IP address. These functions can be used in rewrite or responder policies.

Geolocation function	Example
CLIENT.IP.SRC.LOCATION	Asia.In.Karnataka.Bangalore
CLIENT.IP.SRC.LOCATION.GET (1).LOCATION_LONG	India
CLIENT.IP.SRC.LOCATION(3)	Asia.In.Karnataka
CLIENT.IP.SRC.LAT_LONG	12,77
CLIENT.IPV6.SRC.LOCATION	North America.US.California.Santa Clara.Verizon.Citrix
CLIENT.IPV6.SRC.LOCATION(3)	North America.US.California
CLIENT.IPV6.SRC.LOCATION.GET(1).LOCATION_L	United States

Geolocation function	Example
CLIENT.IPV6.SRC.LOCATION.GET(3)	California
CLIENT.IPV6.SRC.LAT_LONG	36, -119

Configuring geolocation functions

To configure geolocation functions using advanced policy infrastructure, you must enable the load balancing, rewrite, and responder features and then complete the following use cases.

Enable load balancing, responder, rewrite features

If you want the Citrix ADC appliance to authorize user access from a particular geo location, you must enable the load balancing, rewrite, and responder features.

```
enable ns feature loadbalancing rewrite responder
<!--NeedCopy-->
```

Use case 1: Configuring geolocation function for redirecting invalid users outside geo location

When a user from India is requesting access to a webpage, block the request and respond with an HTML page with country name.

The following steps help you to complete the configuration of this use case.

- Add responder action
- Add responder policy
- Bind responder policy to load balancing server

For more information about the GUI procedures for rewrite action and rewrite policy configuration, see Responder topic

Add responder action

Add a responder action to respond with HTML page with country name.
At the command prompt, type:

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>][-reasonPhrase <
    string>]
<!--NeedCopy-->
```


Example:

```
add responder action responder_act respondwith "HTTP.REQ.VERSION + \"
    304 Requested Page not allowed in your country - \" + CLIENT.IP.SRC.
    LOCATION.GET (1).LOCATION_LONG + \"\r\n\""
<!--NeedCopy-->
```

Add audit log message action

You can configure audit message actions to log messages at various log levels, either in syslog format only or in both syslog and newnslog formats. Audit-message actions use expressions to specify the format of the audit messages.
To create an audit message action by using the command line interface

At the command prompt, type:

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-logtoNewnslog
 (YES|NO)] [-bypassSafetyCheck (YES|NO)]
```

Example:

```
add audit messageaction msg1 DEBUG "\"Request Location: \"+CLIENT.IP.
    SRC.LOCATION"
<!--NeedCopy-->
```

Add responder policy

Add a responder policy to identify requests coming from India and associate the responder action to this policy.
At the command prompt, type:

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <
    string>] [-logAction <string>] [-appflowAction <string>]
<!--NeedCopy-->
```

Example:

```
add responder policy responder_pol CLIENT.IP.SRC.MATCHES_LOCATION("Asia
    .India.*.*.*.*") responder_act -logaction msg1
<!--NeedCopy-->
```

Bind responder policy to load balancing server

Bind the responder policy to a load balancing virtual server of type HTTP/SSL.
At the command prompt, type:


```
bind lb vserver <vserver name> -policyName < policy_name >  -priority
    <> -type <L7InlineREQUEST | L4Inline-REQUEST>
<!--NeedCopy-->
```

Example:

```
bind lb vserver http_vserver -policyName responder_pol -priority 100 -
    type REQUEST
<!--NeedCopy-->
```

Use case 2: Configuring geolocation function for inserting new HTTP header with location details for back-end to respond

Consider a scenario, where a Citrix ADC appliance must insert the user location in the HTTP header of a request sent to the application server so that the server can use the information for some business logic.
The following steps help you to complete the configuration of this use case.

- Add rewrite action
- Add rewrite policy
- Bind rewrite policy to load balancing

For more information about the GUI procedures for rewrite action and rewrite policy configuration, see Responder topic.

Add rewrite action

Add a rewrite action to insert a custom HTTP header with user geolocation details in the request and send it back-end servers.
At the command prompt, type:

```
add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
    pattern <expression> | -search <expression>] [-refineSearch <string
    >][-comment <string>]
<!--NeedCopy-->
```

Example:

```
add rewrite action rewrite_act insert_http_header "User_location"
    CLIENT.IP.SRC.LOCATION
<!--NeedCopy-->
```


Add rewrite policy

Add a rewrite policy to evaluate if the rewrite action must be run. In this case, all requests going to the application server must have a custom HTTP header, so the rule can be “true.”
At the command prompt, type:

```
add rewrite policy <name> <rule> <action> [<undefAction>] [-comment <
   string>] [-logAction <string>]
<!--NeedCopy-->
```

Example:

```
add rewrite policy rewrite_pol true rewrite_act -logaction log_act
<!--NeedCopy-->
```

Bind rewrite policy to load balancing

Bind the rewrite policy to the required load balancing virtual server of type HTTP/SSL.
At the command prompt, type:

```
bind lb vserver <vserver name> -policyName < policy_name >  -priority
   <> -type <L7InlineREQUEST | L4Inline-REQUEST>
<!--NeedCopy-->
```

Example:

```
bind lb vserver http_vserver -policyName rewrite_pol -priority 100 -
   type REQUEST
<!--NeedCopy-->
```

Syslog support for logging geolocation details (optional)

If you prefer to log the user’s geolocation details, you must specify the SYSLOG action to be performed when a request matches the policy. The appliance stores the details as a log message in the ns.log file. For more information about SYSLOG and NSLOG auditing, see Audit logging topic.

Output for user geolocation details

The following output is logged in the appliance using the SYSLOG or `newnslog` action if you try to access an application from the Bangalore location and if the appliance uses the geolocation function, “CLIENT.IP.SRC.LOCATION’.


```
Asia.India.Karnataka.Banglore
<!--NeedCopy-->
```

Sample output log:

```
07/23/2018:19:03:54 GMT  Debug 0-PPE-0 : default REWRITE Message 22 0 :
     "Request Location: asia.in.karnataka.bangalore.*.*"
07/23/2018:19:23:55 GMT  Debug 0-PPE-0 : default RESPONDER Message 32 0
Done
<!--NeedCopy-->
```

Use the source IP address of the client when connecting to the server

September 14, 2021

You can configure the Citrix ADC appliance to forward packets from the client to the server without changing the source IP address. This is useful when you cannot insert the client IP address into a header, such as when working with non-HTTP services.

For more information about configuring USIP globally, see Enabling Use Source IP Mode.

To enable USIP mode for a service by using the CLI

At the command prompt, type:

```
set service <name> -usip (YES | NO)
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -usip YES
<!--NeedCopy-->
```

To enable USIP mode for a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and open a service.
2. In Advanced Settings, in the Service Settings section, select **Use Source IP Address**.

Use client source IP address for back end communication in a v4-v6 load balancing configuration

September 14, 2021

In a v4-to-v6 load balancing configuration, for services with USIP disabled, the Citrix ADC appliance communicates to the related servers from one of the configured IPv6 SNIP (SNIP6) addresses.

For services with USIP enabled, you must set the global USIP NAT prefix parameter for making the related servers aware of the client's IP address of the request packets. USIP NAT prefix is a global IPv6 prefix of length 32/40/48/56/64/96 bits configured on the Citrix ADC appliance.

For a load balancing service that has USIP enabled, the appliance translates the IPv4 request packet to an IPv6 packet and sets the source IP address of the translated IPv6 packet to a concatenation of:

- the USIP NAT prefix of length of 32/40/48/56/64/96 bits.
- zeros padded if the USIP NAT prefix length is less than 96 bits. Number of bits padded with zeros = 96-USIP NAT prefix length. For example, if the USIP NAT prefix length is 64, then the number of bits padded with zeros = 96-64 = 32.
- the IPv4 source address [32 bits] that was received in the request packet. In other words, the last 32 bits of the source IPv6 address is set to the IPv4 address of the client.

On receiving an IPv6 response packet from the server, the Citrix ADC appliance translates the IPv6 packet to an IPv4 packet and sets the destination IP address of the translated IPv4 packet to the last 32 bits of the destination IP address of the IPv6 packet.

> **Note**: This feature is not supported for Citrix Gateway configuration and, content switching and cache redirection load balancing configurations.

Configuration Steps

Configuring USIP for a v4-to-v6 load balancing configuration consists of the following tasks:

- **Add global USIP NAT prefix**. It is a global IPv6 prefix of length 32/40/48/56/64/96 bits to be configured on the appliance.
- **Enable global USIP mode**. For more information, see Enable Use Source IP Mode.
- **Enable USIP mode for load balancing services**. For more information, see Use source IP address of the client when connecting to the server.

To add a global USIP NAT prefix by using the CLI:

- `set ipv6 -usipnatprefix <prefix/prefix_length>`
- `show ipv6`

To add a global USIP NAT prefix by using the GUI:

1. Navigate to **System** > **Network**, and click **Change IPv6 Settings**.
2. On the **Configure Configuration for IPV6** screen, set the **USIP NAT Prefix** parameter.

Sample configuration

```
> set ipv6 -usipnatprefix 2001:DB8:90::/64
Done

> enable ns mode USIP
Done

> add lb vserver LBVS-1 HTTP 203.0.113.90 80
Done

> add service SVC-1 2001:DB8:5001::30 HTTP 80 -usip yes
Done

> add service SVC-2 2001:DB8:5001::60 HTTP 80 -usip yes
Done

> bind lb vserver LBVS-1 SVC-1
Done

> bind lb vserver LBVS-1 SVC-2
Done

<!--NeedCopy-->
```

Configure the source port for server-side connections

September 14, 2021

When the Citrix ADC appliance connects to a physical server, it can use the source port from the client's request, or it can use a proxy port as the source port for the connection. You can set the Use Proxy Port parameter to YES to handle situations such as the following scenario:

- The Citrix ADC appliance is configured with two load balancing virtual servers, LBVS1 and LBVS2.
- Both the virtual servers are bound to the same service, S-ANY.
- Use (the client's) source IP address (USIP) is enabled on the service.
- Client C1 sends two requests, Req1 and Req2, for the same service.
- LBVS1 receives Req1 and LBVS2 receives Req2.

- LBVS1 and LBVS2 forward the request to S-ANY, and when S-ANY sends the response, LBVS1 and LBVS2 forward the response to the client.
- Consider two cases:
  - Use the client port. When the appliance uses the client port, both the virtual servers use the client's IP address (because USIP is ON) and the client's port when connecting to the server. Therefore, when the service sends the response, the appliance cannot determine which virtual server must receive the response.
  - Use proxy port. When the appliance uses a proxy port, the virtual servers use the client's IP address (because USIP is ON), but different ports when connecting to the server. Therefore, when the service sends the response, the port number identifies the virtual server that must receive the response.

However, if you require a fully transparent configuration, such as a fully transparent cache redirection configuration, you must disable the Use Proxy port Setting so that the Citrix ADC appliance can use the source port from the client's request.

The Use Proxy Port option becomes relevant if the use source IP (USIP) option is enabled. For TCP-based service types, such as TCP, HTTP, and SSL, the option is enabled by default. For UDP-based service types, such as UDP and DNS, including ANY, the option is disabled by default. For more information about the USIP option, see "Enabling Use Source IP Mode."

You can configure the **Use Proxy Port** setting either globally or on a given service.

Configure the use proxy port setting on a service

You configure the **Use ProxyPort** setting on the service if you want to override the global setting.

To configure the Use Proxy Port setting on a service by using the CLI

At the command prompt, type:

```
set service <name> -useProxyPort (YES | NO)
<!--NeedCopy-->
```

Example:

```
set service svc1 -useproxyport YES
Done

show service svc1
svc1 (10.102.29.30:80) - HTTP
State: UP
. . .
Use Source IP: YES Use Proxy Port: YES
```



```
. . .
Done
<!--NeedCopy-->
```

To configure the Use Proxy Port setting on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select Traffic Settings, and select **Use Proxy Port**.

Configure the use proxy port setting globally

You configure the **Use Proxy Port** setting globally if you want to apply the setting to all the services on the Citrix ADC appliance. The service-specific **Use Proxy Port** settings overrides the global setting.

To configure the Use Proxy Port setting globally by using the CLI

At the command prompt, type the following commands to configure the **Use Proxy Port** setting globally and verify the configuration:

```
set ns param -useproxyport ( ENABLED | DISABLED )`
show ns param`
<!--NeedCopy-->
```

Example:

```
set ns param -useproxyport ENABLED

Done

show ns param
Global configuration settings:
. . .
Use Proxy Port: ENABLED
Done
<!--NeedCopy-->
```

To configure the Use Proxy Port setting globally by using the GUI

Navigate to **System > Settings > Change global system** settings, and select or clear Use Proxy Port.

Set a limit on the number of client connections

September 14, 2021

You can specify a maximum number of client connections that each load balanced server can handle. The Citrix ADC appliance then opens client connections to a server only until this limit is reached. When the load balanced server reaches its limit, monitor probes are skipped, and the server is not used for load balancing until it has finished processing existing connections and frees up capacity.

For more information on the **Maximum Client** setting, see Load Balancing Domain-Name Based Services.

Note: Connections that are in the process of closing are not considered for this limit.

To set a limit to the number of client connections by using the CLI

At the command prompt, type:

```
set service <name> -maxclient <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -maxClient 1000
<!--NeedCopy-->
```

To set a limit to the number of client connections by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Thresholds & Timeouts**, and select **Maximum Clients**.

Set a limit on the number of requests per connection to the server

September 14, 2021

The Citrix ADC appliance can be configured to reuse connections to improve performance. In some scenarios, however, load balanced Web servers might have issues when connections are reused for too many requests. For HTTP or SSL services, use the max request option to limit the number of requests sent through a single connection to a load balanced Web server.

Note: You can configure the max request option for HTTP or SSL services only.

To limit the number of client requests per connection by using the CLI

At the command prompt, type:

```
set service <ServiceName> -maxReq <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -maxReq 100
<!--NeedCopy-->
```

To limit the number of client requests per connection by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Thresholds & Timeouts**, and select **Maximum Requests**.

Set a threshold value for the monitors bound to a service

September 14, 2021

The Citrix ADC appliance designates a service as UP only when the sum of the weights of all monitors bound to it and that are UP is equal to or greater than the threshold value configured on the service. The weight for a monitor specifies how much that monitor contributes to designating the service to which it is bound as UP.

By default, the monitor threshold is set to 0 and monitor weights are set to 1. All monitors have equal weightage then and a service can go DOWN when any one of the monitors goes DOWN.

For example, assume that three monitors, named Monitor-HTTP-1, Monitor-HTTP-2, and Monitor-HTTP-3 respectively, are bound to Service-HTTP-1, and that the threshold configured on the service is three. Suppose the following weights are assigned to each monitor:

- The weight of Monitor-HTTP-1 is 1.
- The weight of Monitor-HTTP-2 is 3.
- The weight of Monitor-HTTP-3 is 1.

The service is marked UP only when one of the following is true:

- Monitor-HTTP-2 is UP.
- Monitor-HTTP-2 and Monitor-HTTP-1 or Monitor-HTTP-3 are UP
- All three monitors are UP.

To set the monitor threshold value on a service by using the CLI

At the command prompt, type:

```
set service <name> -monThreshold <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -monThreshold 100
<!--NeedCopy-->
```

To set the monitor threshold value on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Thresholds & Timeouts**, and select **Monitor Threshold**.

Set a timeout value for idle client connections

September 14, 2021

You can configure the service with a time-out value to terminate any idle client connections when the configured time elapses. If the client is idle during the configured time, the Citrix ADC appliance closes the client connection.

To set a timeout value for idle client connections by using the CLI

At the command prompt, type:

```
set service <name> -cltTimeout <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -cltTimeout 100
<!--NeedCopy-->
```

To set a timeout value for idle client connections by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Thresholds & Timeouts**, and select **Client Idle Time-out**.

Set a timeout value for idle server connections

September 14, 2021

You can configure a service with a timeout value to terminate any idle server connections when the configured time (in seconds) elapses. If the server is idle for the configured amount of time, the Citrix ADC appliance closes the server connection.

To set a timeout value for idle server connections by using the CLI

At the command prompt, type:

```
set service <name> -svrTimeout <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -svrTimeout 100
<!--NeedCopy-->
```

To set a timeout value for idle server connections by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Thresholds & Timeouts**, and select **Server Idle Time-out**.

Set a limit on the bandwidth usage by clients

September 14, 2021

Sometimes, servers might have limited bandwidth to handle client requests and might become overloaded. To prevent overloading a server, you can specify a maximum limit on the bandwidth, in Kbps, processed by the server. The Citrix ADC appliance forwards requests to a load balanced server only until this limit is reached.

To set a maximum bandwidth limit on a service by using the CLI

At the command prompt, type:

```
set service <name> -maxBandwidth <Value>
<!--NeedCopy-->
```


Example:

```
set service Service-HTTP-1 -maxBandwidth 100
<!--NeedCopy-->
```

To set a maximum bandwidth limit on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, select **Thresholds & Timeouts**, and select **Maximum Bandwidth**.

Redirect client requests to a cache

September 14, 2021

You can configure a service to redirect client requests to a cache, and forward the non-cacheable requests to a service chosen by the configured load balancing method.

To set cache redirection on a service by using the CLI

At the command prompt, type:

```
set service <name> -cacheable <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -cacheable YES
<!--NeedCopy-->
```

To set cache redirection on a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Open a service, and set the Cache Type.

Retain the VLAN identifier for VLAN transparency

September 14, 2021

You can configure a load balancing virtual server to retain the client's VLAN identifier in packets that are to be forwarded to servers. The virtual server must be a wildcard virtual server of type ANY, and must be functioning in MAC mode.

To configure a load balancing virtual server to retain the client VLAN ID by using the CLI

At the command prompt, type the following command to configure a load balancing virtual server to retain the client VLAN ID and verify the configuration:

```
set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED

show lb vserver <name>
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which the `-m MAC` option is enabled, you must bind a non-user monitor.

To configure a load balancing virtual server to retain the client VLAN ID by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **Traffic Settings**, and select **Retain VLAN ID**.

Configure automatic state transition based on percentage health of bound services

September 14, 2021

You can configure a load balancing virtual server to automatically transition from the UP state to the DOWN state if the percentage of active services falls below a configured threshold. For example, if you bind 10 services to a load balancing virtual server and configure a threshold of 50% for that virtual server, it transitions from UP to DOWN if six or more services are DOWN. When the percentage health rises above the threshold value, the virtual server returns to the UP state.

You can also enable an SNMP alarm called ENTITY-STATE if you want the Citrix ADC appliance to notify you when the percentage health of bound services causes a virtual server to change state.

To configure percentage based automatic state transition by using the CLI

At the command prompt, type the following commands to configure an automatic state transition for a virtual server and verify the configuration:

```
set lb vserver <name> -healthThreshold <positive_integer>

show lb vserver <name>
<!--NeedCopy-->
```

To configure percentage based automatic state transition by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **Traffic Settings**, and set a **Health Threshold**.

To enable the ENTITY-STATE alarm by using the CLI

At the command prompt, type the following commands to enable the ENTITY-STATE SNMP alarm and verify the configuration:

```
enable snmp alarm ENTITY-STATE

show snmp alarm
<!--NeedCopy-->
```

To enable the ENTITY-STATE alarm by using the GUI

1. Navigate to **System > SNMP > Alarms**.
2. Select **ENTITY-STATE** and, in the Action list, select **Enable**.

Built-in monitors

September 14, 2021

The Citrix ADC appliance contains various built-in monitors that you can use to monitor your services. These built-in monitors handle most of the common protocols. They provide options to modify some parameters, such as interval, response time-out to meet your requirements. However, you cannot modify the monitor name and protocol. For more information, see Modifying Monitors. You can also bind a built-in monitor to a service and unbind it from the service.

> **Note**
>
> You can create a custom monitor based on a built-in monitor. To learn how to create custom monitors, see Configuring Monitors in a Load Balancing Setup.

TCP-based application monitoring

September 14, 2021

The Citrix ADC appliance has two built-in monitors that monitor TCP-based applications: `tcp-default` and `ping-default`. When you create a service, the appropriate default monitor is bound to it automatically, so that the service can be used immediately if it is UP. The tcp-default monitor is bound to all TCP services. The ping-default monitor is bound to all non-TCP services.

You can't delete or modify default monitors. When you bind any other monitor to a TCP service, the default monitor is unbound from the service. The following table lists the monitor types, and the parameters and monitoring processes associated with each type.

Monitor type	Specific parameters	Process
tcp	Not applicable	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination, and then closes the connection. If the appliance observes TCP traffic to the destination, it does not send TCP monitoring requests. This occurs if LRTM is disabled. By default, LRTM is disabled on this monitor.

Monitor type	Specific parameters	Process
http	httprequest [“HEAD /”] - HTTP request that is sent to the service. respcode [200] - A set of HTTP response codes are expected from the service.	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. After the connection is established, the appliance sends HTTP requests, and then compares the response code with the configured set of response codes.
tcp-ecv	send [””] - is the data that is sent to the service. The maximum permissible length of the string is 512 bytes. recv [””] - expected response from the service. The maximum permissible length of the string is 128 bytes. The last character is NULL termination.	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send specific data to the service and expects a specific response through the receive parameter. Different servers send different sizes of segments. However, the pattern must be within 16 TCP segments.

Monitor type	Specific parameters	Process
http-ecv	send [””] - HTTP data that is sent to the service; recv [””] - the expected HTTP response data from the service	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send the HTTP data to the service and expects the HTTP response that the receive parameter specifies. (HTTP body part without including HTTP headers). Empty response data matches any response. Expected data might be anywhere in the first 24 K bytes of the HTTP body of the response.
ping	Not Applicable	The Citrix ADC appliance sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.

To configure built-in monitors for TCP-based applications, see Configuring Monitors in a Load Balancing Setup.

To configure TCP-based monitors by using CLI

Type the following command:

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

Example for TCP monitor type:


```
add lb monitor Exch2010-RPC-AddressBook TCP -LRTM ENABLED -interval 10
    -resptimeout 5 -destPort 59601
<!--NeedCopy-->
```

Example for HTTP monitor type:

```
add lb monitor Mon_S4B_FE_2 HTTP -respCode 200 -httpRequest "GET /
    Autodiscover/XFrame/XFrame.html" -LRTM ENABLED -retries 10 -secure
    YES
<!--NeedCopy-->
```

Example for HTTP-ECV monitor type:

```
add lb monitor STM_EXC2016_SSLBridge_MON HTTP-ECV -send "GET /owa/
    healthcheck.htm" -recv "200 OK" -LRTM ENABLED -destPort 443 -secure
    YES
<!--NeedCopy-->
```

Example for PING monitor type:

```
add lb monitor lbmon-localhost-ping PING -LRTM DISABLED -destIP
    127.0.0.1
<!--NeedCopy-->
```

SSL service monitoring

September 14, 2021

The Citrix ADC appliance has built-in secure monitors, TCPS, and HTTPS. You can use the secure monitors to monitor HTTP and non-HTTP traffic. To configure a secure HTTP monitor, select the monitor type as HTTP, and then set the secure flag. To configure a secure TCP monitor, select the monitor type as TCP, and then set the secure flag. The secure monitors work as follows:

- **Secure TCP monitoring.** The Citrix ADC appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. After the handshake is over, the appliance closes the connection.
- **Secure HTTP monitoring.** The Citrix ADC appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. When the SSL connection is established, the appliance sends HTTP requests over the encrypted channel and checks the response codes.

The following table describes the available built-in monitors for monitoring SSL services.

Monitor type	Probe	Success criteria (Direct condition)
TCP	TCP connection; SSL handshake	Successful TCP connection established and successful SSL handshake.
HTTP	TCP connection; SSL handshake; Encrypted HTTP request	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP response code in server HTTP response is encrypted.
TCP-ECV	TCP connection. SSL handshake (Data sent to a server is encrypted.)	Successful TCP connection is established, successful SSL handshake is performed, and expected TCP data is received from the server.
HTTP-ECV	TCP connection; SSL handshake (Encrypted HTTP request)	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP data is received from the server.

Sample configuration for HTTPS-ECV health check monitor

HTTP services have predefined monitors capable of Extended Content Verification (ECV).
These monitors are used when a validation is required beyond a successful TCP connection. These monitors validate the service as UP, when all the following criteria are met:

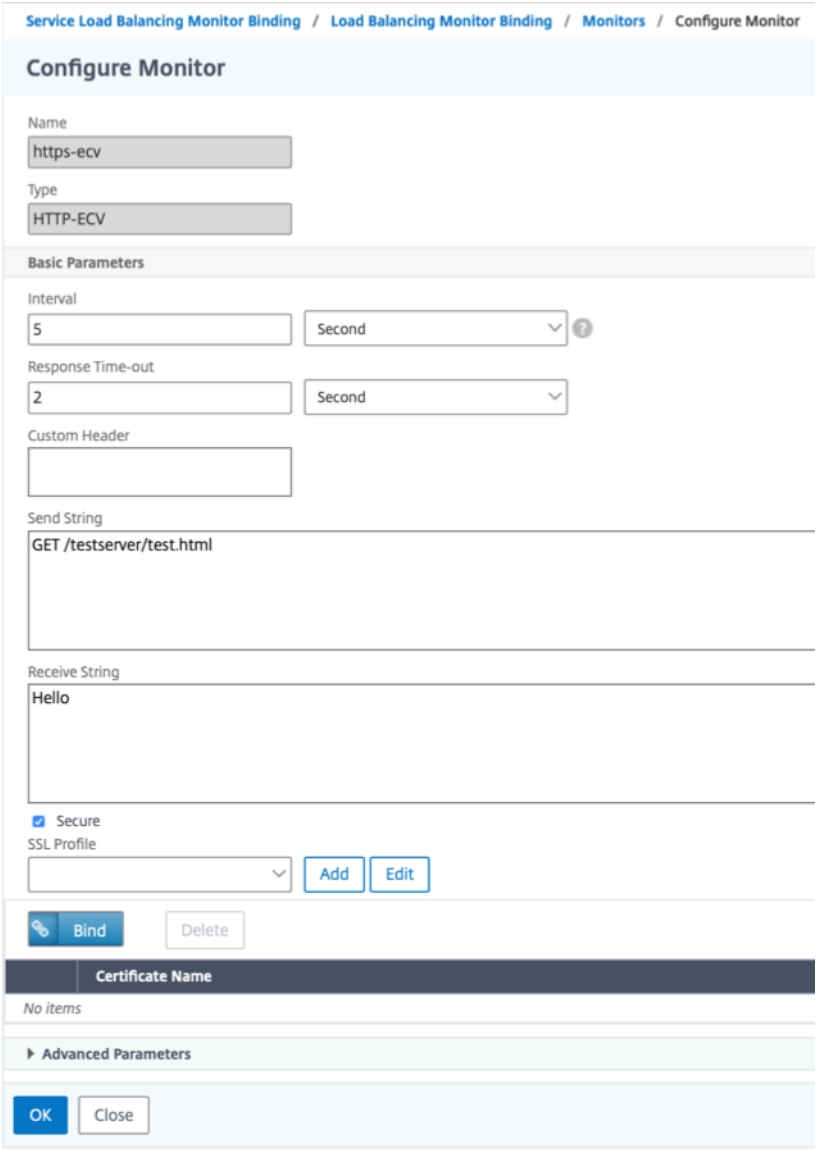
- A successful TCP connection.
- A particular type of request must be generated.
- A specific message is expected in reply from the **Receive String**.

For these monitors, a request string is configured along with a reply string. If the reply string received by the Citrix ADC monitor matches the configured string, then the service is marked UP.

Bind a monitor to a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, create a service, and specify the protocol as **SSL**. Click **OK**.

2. Click in the **Service to Load Balancing Monitor Binding** pane, and click **Add Binding**.
3. Choose the monitor type as **HTTPS-ECV** and click **Edit**.
4. In the **Configure Monitor** pane under **Basic Parameters** tab, enter values for the following parameters:
   - **Send String** – The string that the monitor must send to the service.
   - **Receive String** – The string that the monitor must receive to mark the service as UP.



5. Click **OK** to complete the monitor configuration.
6. Click **Select**.
7. Click **Bind** to bind the **HTTPS-ECV** monitor to the service.
8. Click **Close**.

Bind a monitor to a service by using the CLI

At the command prompt, type:

```
bind service <servicename> -monitorName https-ecv
<!--NeedCopy-->
```

Example:

```
bind services1 -monitorName https-ecv
<!--NeedCopy-->
```

HTTP/2 service monitoring

September 14, 2021

Citrix ADC appliance supports HTTP/2 monitors for monitoring the health status of HTTP/2 services.

HTTP/2 monitor can be configured in two different ways. Depending on the traffic type, you can configure an HTTP/2 monitor.

- **HTTP/2 Direct.** You can configure HTTP/2 Direct for monitoring non-secure HTTP/2 services.
- **HTTP/2 SSL.** You can configure HTTP/2 SSL for monitoring secure traffic over SSL. Enable the secure flag parameter in the HTTP/2 to monitor the SSL traffic.

The http2direct and http2ssl are the two different built-in monitors that are supported for the HTTP/2 protocol.

The following table lists the configuration types, and monitoring processes associated with each type.

Configuration type	Probe	Success criteria
HTTP/2 Direct	TCP Connection; HTTP2 Connection preface & Settings Negotiation; HTTP2 Request	HTTP/2 response status code must match the configured response code.
HTTP/2 SSL	TCP Connection; SSL Handshake; HTTP2 Connection preface & Settings Negotiation; HTTP2 Request	The server must always select ALPN with the HTTP/2 protocol and the HTTP/2 response status code must match the configured response code.

Bind the HTTP/2 monitor to a service by using the CLI

At the command prompt, type:

- `bind service <servicename> -monitorName <name>`
- `bind service <servicename> -monitorName <name>`

Example:

- `bind service s1 -monitorName http2direct`
- `bind service s2 -monitorName http2ssl`

Proxy protocol service monitoring

September 14, 2021

Citrix ADC appliance with a proxy protocol supports monitor check. The monitor check ensures that the back end server also supports the proxy protocol. The Citrix ADC appliance has four built-in monitor types for HTTP or TCP related services: HTTP, HTTPS, HTTP-ECV, and TCP-ECV.

The following table lists the monitor types, and the parameters and monitoring processes associated with each type.

Configuration type	Probe	Success criteria
HTTP	`httprequest` ["HEAD /"] - HTTP request that is sent to the service. `respcode` [200] - A set of HTTP response codes are expected from the service.	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. After the connection is established, the appliance sends HTTP requests, and then compares the response code with the configured set of response codes.

Configuration type	Probe	Success criteria
HTTPS	`httprequest` [“HEAD /”] - HTTPS request that is sent to the service. `respcode` [200] - A set of HTTPS response codes are expected from the service.	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. After the connection is established, the appliance sends HTTPS requests, and then compares the response code with the configured set of response codes.
HTTP-ECV	send [””] - HTTP data that is sent to the service. Received [””] - the expected HTTP response data from the service	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send the HTTP data to the service and expects the HTTP response that the receive parameter specifies. (HTTP body part without including HTTP headers). Empty response data matches any response. Expected data might be anywhere in the first 24 K bytes of the HTTP body of the response.

Configuration type	Probe	Success criteria
TCP-ECV	send [””] - is the data that is sent to the service. The maximum permissible length of the string is 512 K bytes. received [””] - the expected response from the service. The maximum permissible length of the string is 128 K bytes.	The Citrix ADC appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send specific data to the service and expects a specific response through the receive parameter. Different servers send different sizes of segments. However, the pattern must be within 16 TCP segments.

You can configure the proxy protocol monitor using `netprofile`.

Configure proxy protocol monitor by using the CLI

At the command prompt, type:

1. Add net profile with proxy protocol enabled

```
add netprofile <name> -proxyProtocol ( ENABLED | DISABLED )
```

Example:

```
1 add netprofile profile1 -proxyProtocol ENABLED
```

1. Bind the net profile to a service.

```
set service <name> -netprofile <netprofile-name>
```

Example:

```
1 set service S1 -netprofile profile1
```

> **Note**
>
> You can run the preceding command if you want net profile to bound to a service.

1. Bind the net profile to a monitor.

```
set lb monitor <monitor-name> <type> -netprofile <netprofile-name>
```

Example:

```
1 set lb monitor http1 HTTPS  - netprofile profile1
```

> **Note**
>
> - You can run the preceding command if you want the net profile to bound to a monitor.
> - You can select a monitor type of your choice. It can be HTTP, HTTPS, TCP-ECV, or HTTP-ECV.
>
> **Important**
>
> - In a general case, the net profile (proxy protocol enabled) bound to a service is considered.
> - If the net profile is bound to both monitor and service, the net profile bound to monitor is considered. The net profile bound to service is ignored.

FTP service monitoring

September 14, 2021

To monitor FTP services, the Citrix ADC appliance opens two connections to the FTP server. It first connects to the control port, which is used to transfer commands between a client and an FTP server. After it receives the expected response, it connects to the data port, which is used to transfer files between a client and an FTP server. Only when the FTP server responds as expected, on both the connections, it is marked UP.

Note: Monitor probes originate from the NSIP address.

The Citrix ADC appliance has two built-in monitors for FTP services: the FTP monitor and the FTP-EXTENDED monitor. The FTP-EXTENDED monitor is a scriptable monitor. It uses the nsftp.pl script. The FTP-EXTENDED monitor script is enhanced to send secure probes to FTP services. You can create a monitor of type FTP-EXTENDED. The nsftp.pl script is automatically taken from the default directory.

To send secure FTP probes to FTP services by using the CLI

At the command prompt, type:

```
1 add lb monitor <monitorName> <type> -username <string> -password <
     string> -filename <filename>
2 <!--NeedCopy-->
```

Example


```
add monitor mon1 FTP-EXTENDED -username root -password freebsd -
    filename fsdf
<!--NeedCopy-->
```

To send secure FTP probes to FTP services by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**.
2. Specify the monitor type as **FTP-EXTENDED**, and set the parameters.
3. In **Special Parameters**, specify a **File Name**, **User Name**, and **Password**.

To configure built-in monitors to check the state of FTP services, see Configuring Monitors in a Load Balancing Setup.

Secure monitoring of servers by using SFTP

September 14, 2021

A user script 'nssftp.pl' is added to support SSH File Transfer Protocol (SFTP) monitoring. It is available in the current list of in-built Citrix ADC user monitors and is located in the /netscaler/monitors directory. The SFTP monitor uses the specified user name and password to check if the file is present on the server.

To configure secure monitoring using SFTP by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
    string> -secure ( YES | NO )
<!--NeedCopy-->
```

Example:

```
add monitor SFTP_MON USER -scriptname nssftp.pl -scriptargs "file=
    example.txt;user=sam;password=sam_passwd"
<!--NeedCopy-->
```

To configure secure monitoring using SFTP by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors** and in **Type** specify **USER**.

2. In **Special Parameters**, in **Script Name**, select nssftp.pl.
3. Specify the **Script Arguments**.

Set SSL parameters on a secure monitor

September 14, 2021

> **Important**
>
> This feature is supported only on the new Default profiles. For more information about these profiles, see Enhanced SSL Profiles Infrastructure Overview.

A monitor inherits either the global settings or the settings of the service to which it is bound. If a monitor is bound to a non-SSL or non-SSL_TCP service, such as SSL_BRIDGE, you cannot configure it with SSL settings such as the protocol version or the ciphers to be used. Therefore, if your deployment requires SSL-based monitoring of the back-end servers, the monitoring is ineffective.

You can have more control over SSL-based monitoring of back-end servers, by binding an SSL profile to a monitor. An SSL profile contains SSL parameters, cipher bindings, and ECC bindings. For example, you can set server authentication, ciphers, and protocol version in an SSL profile and bind the profile to a monitor. To perform server authentication, you must also bind a CA certificate to a monitor. To perform client authentication, you must bind a client certificate to the monitor. New parameters for the “bind lb monitor” command enable you to do so.

> **Note**
>
> The SSL settings take effect only if you add a secure monitor. Also, the SSL profile type must be **BackEnd**.

Monitor Types that Support SSL Profiles

SSL profiles can be bound to the following monitor types:

- HTTP
- HTTP-ECV
- TCP
- TCP-ECV
- HTTP-INLINE

To specify an SSL profile while adding a monitor by using the command line

At the command prompt, type:


```
add lb monitor <monitorName> <type> -secure  YES   -sslprofile <string>

set lb monitor <monitorName> <type> -secure YES -sslprofile <string>
<!--NeedCopy-->
```

Example:

```
add ssl profile prof1 -sslProfileType  BackEnd

add lb monitor mon1 HTTP -secure  YES -sslprofile prof1
<!--NeedCopy-->
```

To bind a certificate-key pair to a monitor by using the command line

At the command prompt, type:

```
bind monitor  <monitor name> -certkeyName <string>  [(-CA  [-crlCheck (
    Mandatory | Optional ) | -ocspCheck ( Mandatory | Optional )]
<!--NeedCopy-->
```

SIP service monitoring

September 14, 2021

A Citrix ADC has two built-in monitors that you can use to monitor SIP services: the **SIP-UDP** and **SIP-TCP** monitors. A SIP monitor periodically checks the SIP service to which the SIP monitor is bound, by sending SIP request methods to the SIP service. If the SIP service replies with a response code, the monitor marks the service as UP. If the SIP service does not respond, or responds incorrectly, it is marked as DOWN.

Parameter	Specifies
sipURI	SIP addressing schema of the SIP server.
sipmethod	Type of SIP request used to probe the SIP service. Specify one of the following methods: INVITE, OPTION (the default), REGISTER
respcode	SIP response code with which the SIP service responds the probe request. Default: 200.

RADIUS service monitoring

September 14, 2021

The Citrix ADC appliance RADIUS monitor periodically checks the state of the RADIUS service to which it is bound by sending an authentication request to the service. The RADIUS server authenticates the RADIUS monitor and sends a response. By default, the monitor expects to receive a response code of 2, the default Access-Accept response, from the RADIUS server. As long as the monitor receives the appropriate response, it marks the service UP.

Note: RADIUS monitor supports only PAP type authentication.

- If the client authenticated successfully, the RADIUS server sends an Access-Accept response. The default access-accept response code is 2, and this is the code that the appliance uses.
- If the client fails to authenticate successfully (such as when there is a mismatch in the user name, password, or secret key), the RADIUS server sends an Access-Reject response. The default access-reject response code is 3, and this is the code that the appliance uses.

Parameter	Specifies
`userName`	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
radKey	Shared secret key value that the RADIUS server uses during client authentication.
radNASid	NAS-ID that is encapsulated in the payload when an access request is made.
radNASip	The IP address that is encapsulated in the payload when an access-request is made. When radNASip is not configured, the Citrix ADC appliance sends the mapped IP address (MIP) to the RADIUS server as the NAS IP address.

To monitor a RADIUS service, you must configure the RADIUS server to which it is bound as follows:

1. Add the user name and password of the client that the monitor uses for authentication to the

RADIUS authentication database.

2. Add the IP address and secret key of the client to the appropriate RADIUS database.
3. Add the IP addresses that the appliance uses to send RADIUS packets to the RADIUS database. If the Citrix ADC appliance has more than one mapped IP address, or if a subnet IP address (SNIP) is used, you must add the same secret key for all the IP addresses.

   Caution: If the IP address used by the appliance is not added to the RADIUS database, the RADIUS server discards all packets.

To configure built-in monitors to check the state of the RADIUS server, see Configuring Monitors in a Load Balancing Setup.

Monitor accounting information delivery from a RADIUS server

September 14, 2021

You can configure a monitor called a *RADIUS accounting* monitor to determine whether the RADIUS server used for Authentication, Authorization, and Accounting (Citrix ADC AAA) is delivering accounting information as expected. The monitor is of type RADIUS_ACCOUNTING. The probe is generated by a Perl script called nsbmradius.pl, which is located in the /nsconfig/monitors/ directory. The script sends successive accounting request probes to the RADIUS server. The probe is considered successful only if the RADIUS accounting server responds with a packet whose Code field is set to 5, which, according to RFC 2866, indicates an Accounting-Response packet.

When configuring a RADIUS accounting monitor, you must specify a secret key. You can specify optional parameters, each of which represents a RADIUS attribute, such as Acct-Status-Type and Framed-IP-Address. For information about these attributes, see RFC 2865, "Remote Authentication Dial In User Service (RADIUS)," and RFC 2866, "RADIUS Accounting."

To configure a RADIUS accounting monitor by using the command line interface

At the command prompt, type the following commands to configure a RADIUS accounting monitor and verify the configuration:

```
add lb monitor <monitorName> RADIUS_ACCOUNTING [-userName <string>] {
 -password  }
  {
 -radKey  }
  [-radNASip <ip_addr>] [-radAccountType <positive_integer>] [-
     radFramedIP <ip_addr>] [-radAPN <string>] [-radMSISDN <string>] [-
     radAccountSession <string>]
```



```
6
7 show lb monitor <monitorName>
8 <!--NeedCopy-->
```

Example

```
1 add lb monitor radAccntMon RADIUS_ACCOUNTING -radKey "8d#>9jr4rV)L7%a2-
     zW13sM"
2 <!--NeedCopy-->
```

DNS and DNS-TCP service monitoring

September 14, 2021

The Citrix ADC appliance has two built-in monitors that can be used to monitor DNS services: DNS and DNS-TCP. When bound to a service, either monitor periodically checks the state of that DNS service by sending a DNS query to it. The query resolves to an IPv4 or IPv6 address. That IP address is then checked against the list of test IP addresses that you configure. The list can contain up to five IP addresses. If the resolved IP address matches at least one IP address on the list, the DNS service is marked as up. If the resolved IP does not match any IP addresses on the list, the DNS service is marked as down.

Parameter	Description
query	The DNS query (domain name) sent to the DNS service that is being monitored. Default value: “\007” If the DNS query succeeds, the service is marked as UP. Otherwise, it is marked as DOWN. For a reverse monitor, if the DNS query succeeds, the service is marked as DOWN. Otherwise, it is marked as UP. If no response is received, the service is marked as DOWN.
queryType	The type of DNS query that is sent. Possible values: Address, Zone.
IPAddress	List of IP addresses that are checked against the response to the DNS monitoring probe.
IPv6	Select this check box if the IP address uses the IPv6 format.

To configure the built-in DNS or DNS-TCP monitors, see Configuring Monitors in a Load Balancing Setup.

LDAP service monitoring

September 14, 2021

The Citrix ADC appliance has one built-in monitor that can be used to monitor LDAP services: the LDAP monitor. It periodically checks the LDAP service to which it is bound by authenticating and sending a search query to it. If the search is successful, the service is marked UP. If the LDAP server does not locate the entry, a failure message is sent to the LDAP monitor, and the service is marked DOWN.

Configure the LDAP monitor to define the search that it must perform when sending a query. You can use the Base DN parameter to specify a location in the directory hierarchy where the LDAP server must start the test query. You can use the Attribute parameter to specify an attribute of the target entity.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
baseDN	Base name for the LDAP monitor from where the LDAP search must start. If the LDAP server is running locally, the default value of base is dc=`netscaler`, dc=com.
bindDN	BDN name for the LDAP monitor.
filter	Filter for the LDAP monitor. Use the filter parameter in a query to limit the number of results. If you do not specify this parameter in the query, the filter applies for the entire object class, which might be a costly operation, such as a high CPU usage.
password	Password used in monitoring LDAP servers.
attribute	Attribute for the LDAP monitor.

To configure the built-in LDAP monitor, see Configuring Monitors in a Load Balancing Setup.

MySQL service monitoring

September 14, 2021

The Citrix ADC appliance has one built-in monitor that can be used to monitor MySQL services: the MySQL monitor. It periodically checks the MySQL service to which it is bound by sending a search query to it. If the search is successful, the service is marked UP. If the MySQL server does not respond or the search fails, a failure message is sent to the MySQL monitor, and the service is marked DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
database	Database that is used for the MySQL monitor.
sqlQuery	SQL query that is used for the MySQL monitor.

To configure a built-in MySQL monitor, see Configuring Monitors in a Load Balancing Setup.

To configure MySQL monitors by using CLI

Type the following command:

```
1 add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
     string>
2 <!--NeedCopy-->
```

Example:

```
1 add lb monitor mysql1 USER -scriptName nsmysql.pl -scriptArgs "database
     =cloud;user=cloud;password=password;query=show tables from cloud"
2 <!--NeedCopy-->
```

SNMP service monitoring

September 14, 2021

The Citrix ADC appliance has one built-in monitor that can be used to monitor SMNP services: the SNMP monitor. It periodically checks the SNMP agent on the service to which it is bound by sending a query for the enterprise identification ID (OID) that you configure for monitoring. If the query is successful, the service is marked UP. If the SNMP service finds the OID that you specified, the query

succeeds and the SNMP monitor marks the service UP. If it does not find the OID, the query fails and the SNMP monitor marks service DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
SNMPOID	OID that is used for the SNMP monitor.
snmpCommunity	Community that is used for the SNMP monitor.
snmpThreshold	Threshold that is used for the SNMP monitor.
snmpVersion	SNMP version that is used for load monitoring. Possible Values: V1, V2.

To configure the built-in SNMP monitor, see Configuring Monitors in a Load Balancing Setup.

NNTP service monitoring

September 14, 2021

The Citrix ADC appliance has one built-in monitor that can be used to monitor NNTP services: the NNTP monitor. It periodically checks the NNTP service to which it is bound by connecting to the service and checking for the existence of the newsgroup that you specify. If the newsgroup exists, the search is successful and the service is marked UP. If the NNTP service does not respond or the search fails, the service is marked DOWN.

Note: Monitor probes originate from the NSIP address.

The NNTP monitor can optionally be configured to post a test message to the newsgroup as well.

Parameter	Specifies
`userName`	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
group	Group name to be queried for NNTP monitor.

To configure the built-in NNTP monitor, see Configuring Monitors in a Load Balancing Setup.

POP3 service monitoring

September 14, 2021

The Citrix ADC appliance has one built-in monitor that can be used to monitor POP3 services: the POP3 monitor. It periodically checks the POP3 service to which it is bound by opening a connection with a POP3 server. If the POP3 server responds with the correct response codes within the configured time period, it marks the service UP. If the POP3 service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
userName	User name POP3 server. This user name is used in the probe.
password	Password used in monitoring POP3 servers.
scriptName	The path and name of the script to execute.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in POP3 monitor, see Configuring Monitors in a Load Balancing Setup.

To configure POP3 monitors by using CLI

Type the following command:

```
1 add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
     string>
2 <!--NeedCopy-->
```

Example:

```
1 add lb monitor pop31 USER -scriptName nspop3.pl -scriptArgs "user=
     test@lbmon1.net;password=Freebsd123"
2
```



```
<!--NeedCopy-->
```

SMTP service monitoring

September 14, 2021

The Citrix ADC appliance has a built-in monitor that can be used to monitor SMTP services: the SMTP monitor. The monitor checks the SMTP service to which it is bound by opening a connection with it and conducting a series of handshakes to ensure that the server is operating correctly. If the SMTP service completes the handshakes properly, the monitor marks the service UP. Else, if the SMTP service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
scriptName	The path and name of the script to run.
dispatcherIP	The IP Address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in SMTP monitor, see Configuring Monitors in a Load Balancing Setup.

RTSP service monitoring

September 14, 2021

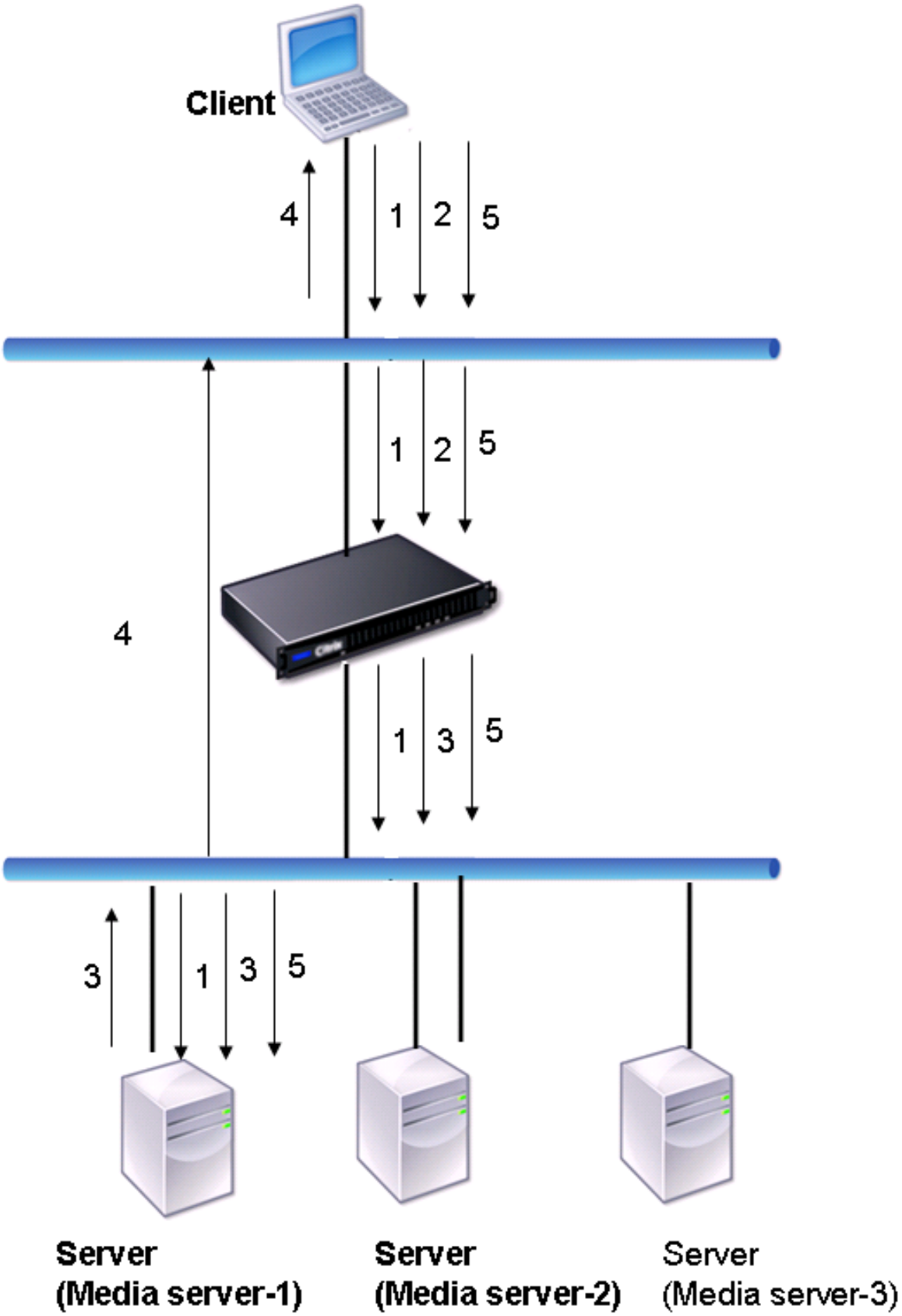
The Citrix ADC appliance has one built-in monitor that can be used to monitor RTSP services: the RTSP monitor. It periodically checks the RTSP service to which it is bound by opening a connection with the load balanced RTSP server. The type of connection that it opens, and the response that it expects, differs depending upon the network configuration. If the RTSP service responds as expected within the configured time period, it marks the service UP. If the service does not respond, or responds incorrectly, it marks the service DOWN.

The Citrix ADC appliance can be configured to load balance RTSP servers using two topologies: NAT-off and NAT-on. RTSP servers send their responses directly to the client, bypassing the appliance. The appliance must be configured to monitor RTSP services differently depending upon which topology

your network uses. The appliance can be deployed either in inline or non-inline mode in both NAT-off and NAT-on mode.

In NAT-off mode, the appliance operates as a router: it receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. If your load balanced RTSP servers are assigned publicly accessible FQDNs in DNS, the load balanced servers send their responses directly to the client, bypassing the appliance. The following figure demonstrates this configuration.

Figure 1. RTSP in NAT-off Mode



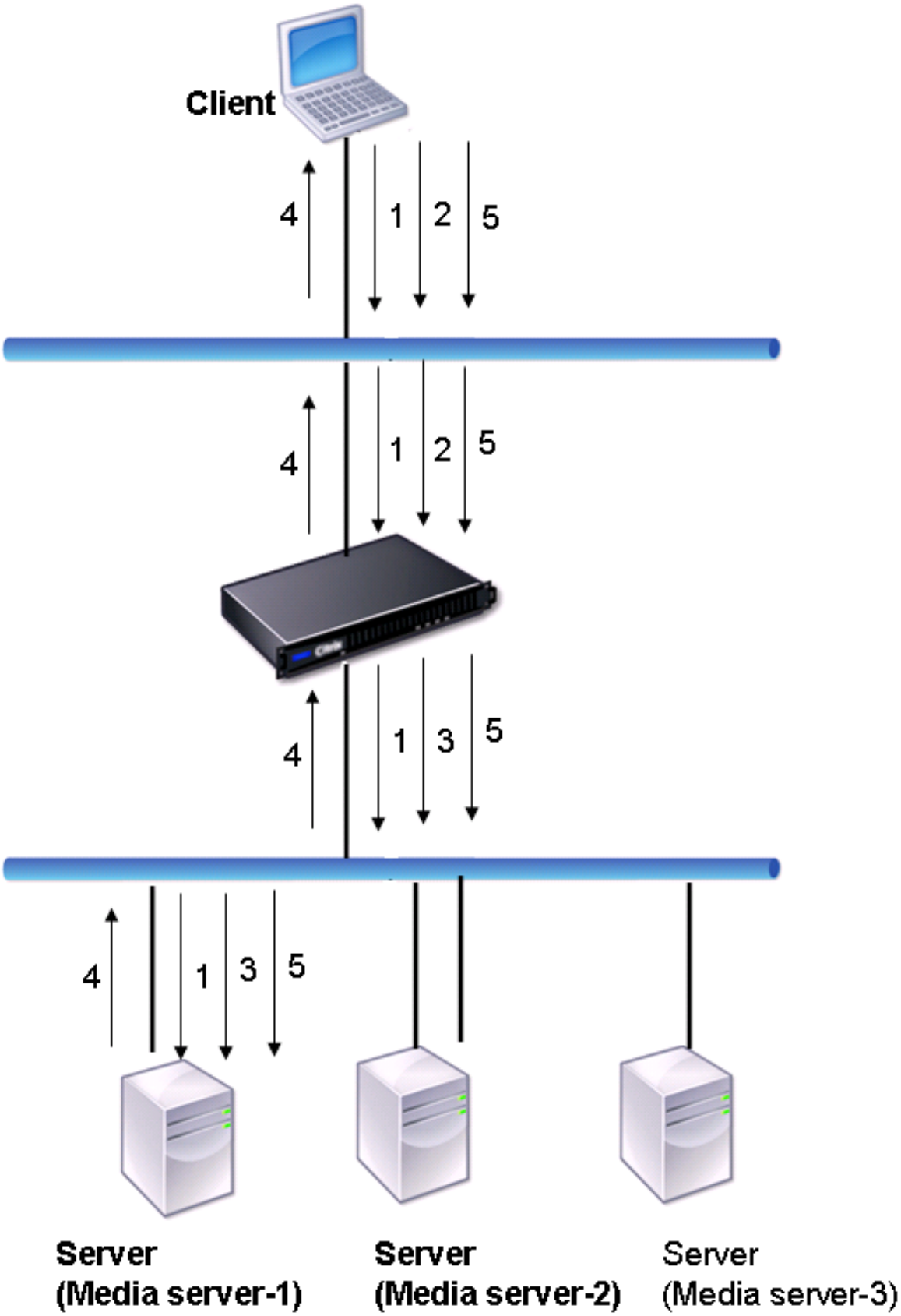
The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media Server-1.
2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:
   - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
   - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request might be sent to a different service.
3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.

   Note: The appliance does not perform NAT to identify the RTSP connection, because the RTSP connections bypass it.
4. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the media server. Media Server-1 performs the requested actions, such as play, forward, or rewind.

In NAT-on mode, the appliance receives RTSP requests from the client and routes those requests to the appropriate media server using the configured load balancing method. The media server then sends its responses to the client through the appliance, as illustrated in the following diagram.

Figure 2. RTSP in NAT-on Mode



The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media Server-1.
2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using the RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:
   - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
   - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request might be sent to a different service.
3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.
4. The appliance performs NAT to identify the client for RTSP data connections, and the RTSP connections pass through the appliance and are routed to the correct client.
5. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the appliance. The appliance uses RTSPSID persistence to identify the appropriate service, and routes the request to Media Server-1. Media Server-1 performs the requested action, such as play, forward, or rewind.

The RTSP monitor uses the RTSP protocol to evaluate the state of the RTSP services. The RTSP monitor connects to the RTSP server and conducts a sequence of handshakes to ensure that the server is operating correctly.

Parameter	Specifies
rtspRequest	The RTSP request string that is sent to the RTSP server (for example, OPTIONS *). The default value is 07. The length of the request must not exceed 163 characters.
respCode	Set of response codes that are expected from the service.

For instructions on configuring an RTSP monitor, see Configuring Monitors in a Load Balancing Setup.

XML Broker Service monitoring

September 14, 2021

The Citrix ADC appliance has a built-in monitor type, CITRIX-XML-SERVICE, with which you can create monitors to monitor the XML Broker Services. The XML Broker Services are used by Citrix XenApp. The monitor opens a connection to the service and periodically probes the XML services to which it is bound. If the server responds as expected within the configured time period, the monitor marks the service UP. If the service does not respond, or responds incorrectly, the monitor marks the service DOWN.

To configure a CITRIX-XML-SERVICE monitor, you need to specify the application name in addition to setting the standard parameters. The application name is the name of the application that has to be run to monitor the state of the XML Broker Service. The default application is Notepad.

To configure monitors for XML Broker Services, see Configuring Monitors in a Load Balancing Setup.

> **Note**
>
> The parameter "Application Name" for the Citrix-XML-Service monitor is invalid for XenApp and Citrix Virtual Desktops versions 7 and later. It is recommended not to use this parameter in XA/XD 7. In case you configure this parameter, this parameter is not used internally. The probing criteria is different starting XA/XD 7. However, you can use the "Application Name" parameters in versions earlier to XA/XD 7.

ARP request monitoring

September 14, 2021

The Citrix ADC appliance has one built-in monitor that can be used to monitor ARP requests: the ARP monitor. This monitor periodically sends an ARP request to the service to which it is bound, and listens for the expected response. If it receives the expected response, it marks the service UP. If it receives no response or the wrong response, it marks the service DOWN.

ARP locates a hardware address for a load balanced server when only the network layer address is known. ARP is compatible with IPv4 to translate IP addresses to Ethernet MAC addresses. ARP monitoring is not relevant to IPv6 networks, and is therefore not supported on those networks.

There are no special parameters for the ARP monitor.

For instructions on configuring an ARP monitor, see Configuring Monitors in a Load Balancing Setup.

XenDesktop Delivery Controller service monitoring

September 14, 2021

In desktop virtualization, the Citrix ADC appliance can be used to load balance the Web Interface (WI) servers and the XenDesktop Delivery Controller servers deployed by the Citrix XenDesktop environment. The Citrix ADC appliance provides a built-in monitor, `CITRIX-XD-DDC` monitor, which monitors the XenDesktop Delivery Controller servers. In addition to the health check, you can also verify whether the probe is sent by a valid user of the XenDesktop Delivery Controller server.

The monitor sends a probe to the XenDesktop Delivery Controller server in the form of an XML message. If the server responds to the probe with the identity of the server farm, the probe is considered to be successful and the server's status is marked as UP. If the HTTP response does not have a success code or the identity of the server farm is not present in the response, the probe is considered to be a failure and the server's status is marked as DOWN.

The Validate Credentials option determines the probe to be sent by the monitor to the XenDesktop Delivery Controller server, that is, whether to request only the server name or to also validate the login credentials.

Note: Regardless of whether the user credentials (user name, password, and domain) are specified on the
`CITRIX-XD-DDC` monitor, the XenDesktop Delivery Controller server validates the user credentials only if the option to validate credentials is enabled on the monitor.

If you use the wizard for configuring the load balancing of the XenDesktop servers, the `CITRIX-XD-DDC` monitor is automatically created and bound to the XenDesktop Delivery Controller services.

To add an XD-DDC monitor with the validate credentials option by using the command line interface

At the command prompt, type the following commands to add an XD-DDC monitor and verify the configuration:

```
add lb monitor <monitorName> <monitorType> -userName <userName> -
    password <password> -domain <domain_name> -validateCred YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -
    password E12Dc35450a1 -domain dhop -validateCred YES
```



```
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval......:..5 sec...Retries......:..3
Response timeout......:..2 sec...Down time......:..30 sec
Reverse......:..NO...Transparent......:..NO
Secure......:..NO...LRTM......:..ENABLED
Action......:..Not applicable...Deviation......:..0 sec
Destination IP......:..Bound service
Destination port......:..Bound service
Iptunnel......:..NO
TOS......:..NO...TOS ID......:..0
SNMP Alert Retries......:..0...Success Retries......:..1
Failure Retries......:..0

Special parameters:
User Name..........:"Administrator"
Password..........:*****
DDC Domain.........: "dhop"
Done
<!--NeedCopy-->
```

To specify the validate credentials option on an XD-DDC monitor by using the command line interface

At the command prompt, type:

```
set lb monitor <monitorName> <monitorType> -userName -password -domain
   <domain_name> -validateCred YES
<!--NeedCopy-->
```

Example:

```
 set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName
    Administrator -password D123S1R2A123 -domain dhop -validateCred YES
Done
<!--NeedCopy-->
```


To configure an XD-DDC monitor with the validate credentials option by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Monitors**, and create a monitor of type `Citrix-XD-DDC`.

Citrix StoreFront stores monitoring

September 14, 2021

You can configure a user monitor for a Citrix StoreFront store. The monitor determines the state of the StoreFront store by successively probing the account service, discovery service, and authentication endpoint (if the Citrix StoreFront Store is an authenticated store). If any of those services do not respond to the probe, the monitor probe fails, and the StoreFront store is marked as DOWN. The monitor sends probes to the IP address and port of the bound service. For more information, see Citrix StoreFront Store Services API.

Note: Monitor probes originate from the NSIP address. However, if the subnet of a StoreFront server is different from that of the appliance, then the subnet IP (SNIP) address is used.

Beginning with release 10.1 build 120.13, you can also bind a StoreFront monitor to a service group. A monitor is bound to each member of the service group and probes are sent to the IP address and port of the bound member (service). Also, because each member of a service group is now monitored by using the member's IP address, you can now use the StoreFront monitor to monitor StoreFront cluster nodes that are added as members of the service group.

In earlier releases, the StoreFront monitor tried to authenticate anonymous stores. As a result, a service can be marked as DOWN and you cannot launch XenApp or XenDesktop by using the URL of the load balancing virtual server.

From build 64.x, the probe order has changed. The monitor now determines the state of the StoreFront store by successively probing the account service, the discovery document, and then the authentication service, and skips authentication for anonymous stores.

The host name parameter for StoreFront monitors is deprecated. The secure parameter is now used to determine whether to use HTTP (the default) or HTTPS to send monitor probes.

To use HTTPS, set the secure option to Yes.

To create a StoreFront monitor by using the command line interface

At the command prompt, type the following commands to configure a StoreFront monitor and verify the configuration:


```
add lb monitor <monitorName> STOREFRONT <string> -storeName <string> [-
    storefrontacctservice ( YES | NO )] -secure ( YES | NO )

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example

```
add lb monitor storefront_ssl STOREFRONT -storename myStore -
    storefrontacctservice YES -secure YES
<!--NeedCopy-->
```

To create a StoreFront monitor by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Monitors**, and create a monitor of type **STOREFRONT**.

> **Note**
>
> For more information about the StoreFront monitors, see StoreFront documentation.

Custom monitors

September 14, 2021

In addition to built-in monitors, you can use custom monitors to check the state of your services. The Citrix ADC appliance provides several types of custom monitors based on scripts that are included with the Citrix ADC operating system. The scripts can be used to determine the state of services based on the load on the service or network traffic sent to the service. Custom monitors are the inline monitors, user monitors, and load monitors.

With these types of monitors, you can use the supplied functionality, or you can create your own scripts and use those scripts to determine the state of the service to which the monitor is bound.

Configure HTTP-inline monitors

September 14, 2021

Inline monitors analyze and probe the responses from the services to which they are bound only when those services receive client requests. The inline monitor is of type HTTP-INLINE and can only be configured with HTTP and HTTPS services. An inline monitor determines that the service to which it is bound is UP by checking its responses to the requests that are sent to it. When no client requests are sent to the service, the inline monitor probes the service by using the configured URL.

Note: Inline monitors cannot be bound to HTTP or HTTPS Global Server Load Balancing (GSLB) remote or local services because these services represent virtual servers rather than actual load balanced Web servers.

Inline monitors have a time-out value and a retry count when probes fail. You can select any of the following action types for the Citrix ADC appliance to take when a failure occurs:

- **NONE.** No explicit action is taken. You can view the service and monitor, and the monitor indicates the number of current contiguous error responses and cumulative responses checked.
- **LOG.** Logs the event in ns/syslog and displays the counters.
- **DOWN.** Marks the service down and does not direct any traffic to the service. This setting breaks any persistent connections to the service. This action also logs the event and displays counters.

After the service is down, the service remains DOWN for the configured downtime. After the downtime elapses, the inline monitor uses the configured URL to probe the service to see if it is available again. If the probe succeeds, the state of the service is changed to UP. Traffic is directed to the service, and monitoring resumes as before.

To configure inline monitors, see Configuring Monitors in a Load Balancing Setup.

To configure HTTP-inline monitors by using CLI

Type the following command:

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

Example:

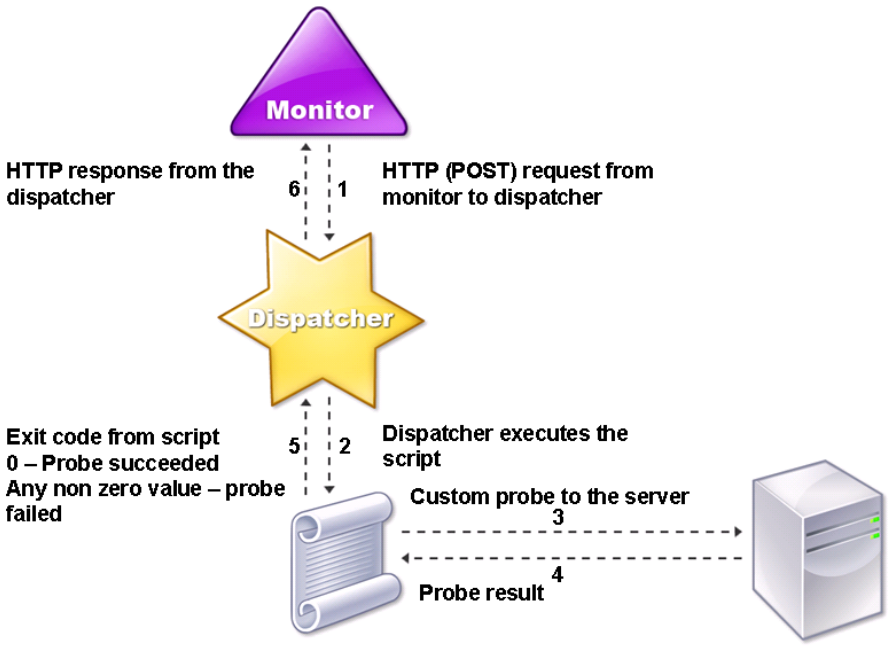
```
add lb monitor http_inline HTTP-INLINE -respCode 200 304 -httpRequest "
    HEAD /var/static/empty.htm" -resptimeout 4 -retries 1 -downTime 2 -
    action NONE
<!--NeedCopy-->
```


Understand user monitors

September 14, 2021

User monitors extend the scope of custom monitors. You can create user monitors to track the health of customized applications and protocols that the Citrix ADC appliance does not support. The following diagram illustrates how a user monitor works.

Figure 1. User Monitors



A user monitor requires the following components.

- **Dispatcher.** A process, on the appliance, that listens to monitoring requests. A dispatcher can be on the loopback IP address (127.0.0.1) and port 3013. Dispatchers are also known as internal dispatchers. A dispatcher can also be a web server that supports the Common Gateway Interface (CGI). Such dispatchers are also known as external dispatchers. They are used for custom scripts that do not run on the FreeBSD environment, such as .NET scripts.

  Note: You can configure the monitor and the dispatcher to use HTTPS instead of HTTP by enabling the “secure” option on the monitor, and configure it as an external dispatcher. However, an internal dispatcher understands only HTTP, and cannot use HTTPS.

In a HA setup, the dispatcher runs on both the primary and secondary Citrix ADC appliances. The dispatcher remains inactive on the secondary appliance.

Script. The script is a program that sends custom probes to the load balanced server and returns the response code to the dispatcher. The script can return any value to the dispatcher, but if a probe succeeds, the script must return a value of zero (0). The dispatcher considers any other value as probe failure.

The Citrix ADC appliance is bundled with sample scripts for commonly used protocols. The scripts exist in the /nsconfig/monitors directory. If you want to add a script, add it there. To customize an existing script, create a copy with a new name and modify it.

Important:

- Starting with Citrix ADC release 13.0 build 41.20, you can use the `nsntlm-lwp.pl` script to create a monitor for monitoring a secure NTLM server.
- Starting with release 10.1 build 122.17, the script files for user monitors are in a new location.

  If you upgrade an MPX or VPX virtual appliance to release 10.1 build 122.17 or later, the changes are as follows:

  - A new directory named conflicts is created in /nsconfig/monitors/ and all the built-in scripts of the previous builds are moved to this directory.
  - All new built-in scripts are available in the /netscaler/monitors/ directory. All custom scripts are available in the /nsconfig/monitors/ directory.
  - Save a new custom script in the /nsconfig/monitors/ directory.
  - After the upgrade is completed, if a custom script is created and saved in the /nsconfig/monitors/ directory, with the same name as the built-in script, the script in the /netscaler/monitors/ directory takes priority. The custom script does not run.

  If you provision a virtual appliance with release 10.1 build 122.17 or later, the changes are as follows:

  - All built-in scripts are available in the /netscaler/monitors/ directory.
  - The /nsconfig/monitors/ directory is empty.
  - If you create a custom script, you must save it in the /nsconfig/monitors/ directory.

For the scripts to function correctly:

- The maximum number of characters in the name of the script must not exceed 63.
- The maximum number of script arguments that can be provided to a script must not exceed 512.
- The maximum number of characters that can be provided in the parameter script arguments must not exceed 639.

To debug the script, you must run it by using the nsumon-debug.pl script from the CLI. You use the script name (with its arguments), IP address, and the port as the arguments of the nsumon-debug.pl

script. Users must use the script name, IP address, port, time-out, and the script arguments for the nsumon-debug.pl script.

At the CLI, type:

```
nsumon-debug.pl <scriptname> <IP> <port> <timeout> <partitionID> [
    scriptarguments][is_secure]
<!--NeedCopy-->
```

Important: Starting with release 10.5 build 57.x, and 11.0 script files for user monitors support IPv6 addresses and include the following changes:

- For the following protocols, new `pm files` have been included for IPv6 support.
  - RADIUS
  - NNTP
  - POP3
  - SMTP
- The following sample scripts in /netscaler/monitors/ has been updated for IPv6 support:
  - nsbmradius.pl
  - nsldap.pl
  - nsnntp.pl
  - nspop3 nssf.pl
  - nssnmp.pl
  - nswi.pl
  - nstftp.pl
  - nssmtp.pl
  - nsrdp.pl
  - nsntlm-lwp.pl
  - nsftp.pl
  - nsappc.pl

  After upgrading to release 10.5 build 57.x, or 11.0, if you want to use your existing custom scripts with IPv6 services, make sure that you update the existing custom scripts with the changes provided in the updated sample scripts in /netscaler/monitors/.

  Note: The sample script nsmysql.pl does not support the IPv6 address. If an IPv6 service is bound to a user monitor that uses nsmysql.pl, the probe fails.
- The following LB monitor types have been updated to support IPv6 addresses:

- USER
- SMTP
- NNTP
- LDAP
- SNMP
- POP3
- FTP_EXTENDED
- StoreFront
- APPC
- CITRIX_WI_EXTENDED

If you are creating a custom script that uses one of these LB monitors types, ensure to include IPv6 support in the custom script. Refer to the associated sample script in /netscaler/monitors/ for the changes that you have to make in the custom script for IPv6 support.

To track the status of the server, the monitor sends an HTTP POST request to the configured dispatcher. This POST request contains the IP address and port of the server, and the script that must be run. The dispatcher runs the script as a child process, with user-defined parameters (if any). Then, the script sends a probe to the server. The script sends the status of the probe (response code) to the dispatcher. The dispatcher converts the response code to an HTTP response and sends it to the monitor. Based on the HTTP response, the monitor marks the service as up or down.

The Citrix ADC appliance logs the error messages to the /var/nslog/nsumond.log file when user monitor probes fail. These detailed error messages are displayed in the GUI, and in the CLI for the `show service`/`service group` commands.

The following table lists the user monitors and the possible reasons for failure.

User monitor type	Probe failure reasons
SMTP	Monitor fails to establish a connection to the server.
NNTP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to find the NNTP group.

User monitor type	Probe failure reasons
LDAP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to bind to the LDAP server.
	Monitor fails to locate an entry for the target entity in the LDAP server.
FTP	The connection to the server times out.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
	Monitor fails to find the file on the server.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
	Preparation of the SQL query fails.
	Execution of the SQL query fails.
SNMP	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.

User monitor type	Probe failure reasons
	Logon fails.
	Monitor fails to create the SNMP session.
	Monitor fails to find the object identifier.
	The monitor threshold value setting is greater than or equal to the actual threshold of the monitor.
RDP (Windows Terminal Server)	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to create a socket.
	Mismatch in versions.
	Monitor fails to confirm connection.

You can view the log file from the CLI by using the following commands, which open a BSD shell, display the log file on the screen, and then close the BSD shell and return you to the CLI:

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
<!--NeedCopy-->
```

Before Citrix ADC release 13.0 build 52.X, the `show service/service group` command displayed a generic error message saying “probe failed” as the cause for the user monitor probe failure.

Example:

```
show service ftp

Monitor Name: mon2
State: UNKNOWN  Weight: 1   Passive: 0
Probes: 3    Failed [Total: 0 Current: 0]
Last response: Failure - Probe failed.
Response Time: 1071.838 millisec
<!--NeedCopy-->
```

From Citrix ADC release 13.0 build 52.X onwards, the `show service/service group` command displays the actual cause for the user monitor probe failure.

Example:

```
show service ftp

Monitor Name: mon2
State: DOWN   Weight: 1   Passive: 0
Probes: 729   Failed [Total: 726 Current: 726]
Last response: Failure - Login failed.
Response Time: 8000.0 millisec
<!--NeedCopy-->
```

User monitors also have a time-out value and a retry count for probe failures. You can use user monitors with non-user monitors. During high CPU utilization, a non-user monitor enables faster detection of a server failure.

If the user monitor probe times out during high CPU usage, the state of the service remains unchanged.

> **Note:**
>
> For scriptable monitors, the response timeout must be configured to a value equal to the expected timeout + 1 second.
>
> For example, if you expect the timeout to be 4 seconds, configure the response timeout as 5 seconds.
>
> **Example command:**

```
add lb monitor <name> USER  - scriptname <script-name> -resptimeout 5
    seconds
<!--NeedCopy-->
```

How to use a user monitor to check websites

September 14, 2021

You can configure a user monitor to check for specific website problems that are reported by HTTP servers using specific HTTP codes. The following table lists the HTTP response codes that this user monitor expects.

HTTP response code	Meaning
200 - success	Probe success.
503 - service unavailable	Probe failure.

HTTP response code	Meaning
404 - not found	Script not found or cannot run.
500 - Internal server error	Internal error/resource constraints in dispatcher (out of memory, too many connections, unexpected system error, or too many processes). The service is not marked DOWN.
400 - bad request	Error parsing HTTP request.
502 - bad gateway	Error decoding script's response.

You configure the user monitor for HTTP by using the following parameters.

Parameter	Specifies
scriptName	The path and name of the script to run.
scriptArgs	The strings that are added in the POST data. They are copied to the request verbatim.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.
localfileName	The name of a monitor script file on the local system.
destPath	A particular location on the Citrix ADC appliance where the uploaded local file is stored.

To create a user monitor to monitor HTTP, see Configuring Monitors in a Load Balancing Setup.

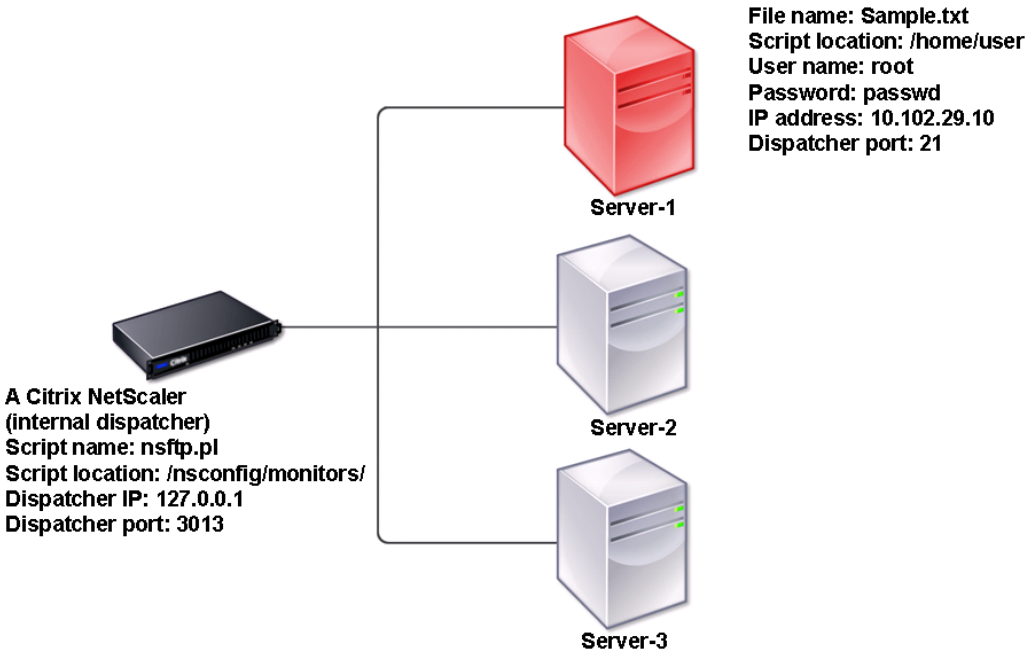
Understand the internal dispatcher

September 14, 2021

You can use a custom user monitor with the internal dispatcher. Consider a case where you need to track the health of a server based on the presence of a file on the server. The following diagram

illustrates this scenario.

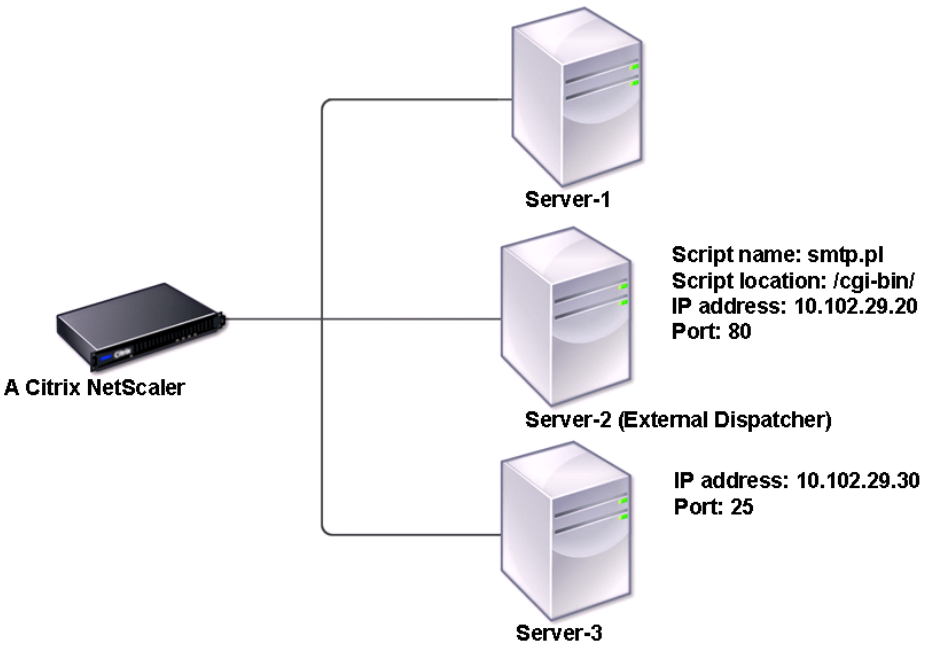
Figure 1. Using a user monitor with the internal dispatcher



A possible solution is to use a Perl script that initiates an FTP session with the server and checks for the presence of the file. You can then create a user monitor that uses the Perl script. The Citrix ADC appliance includes such a Perl script (nsftp.pl), in the /nsconfig/monitors/ directory.

You can use a user monitor with an external dispatcher. Consider a case where you must track the health of a server based on the state of an SMTP service on another server. This scenario is illustrated in the following diagram.

Figure 2. Using a user monitor with an external dispatcher



A possible solution would be to create a Perl script that checks the state of the SMTP service on the server. You can then create a user monitor that uses the Perl script.

Configure a user monitor

September 14, 2021

To configure a user monitor, you must write a script that the monitor uses to check the services that are bound to it. Upload the script to the /nsconfig/monitors directory on the Citrix ADC appliance. Give executable permission to the script. If the monitor type is a protocol that the Citrix ADC appliance does not support, only then you must use the monitor of type **USER**. The user monitors support Perl and Bash scripts. Python scripts are not supported.

> **Note**
>
> Monitor probes originate from the NSIP address. The `scriptargs` configured for the monitor of type USER is displayed in the running configuration and ns.conf files.

To configure a user monitor by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> USER -scriptname <NameOfScript> -
   scriptargs <Arguments>
<!--NeedCopy-->
```

Example:

```
add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file
   =/home/user/
sample.txt;user=root;password=passwd"
<!--NeedCopy-->
```

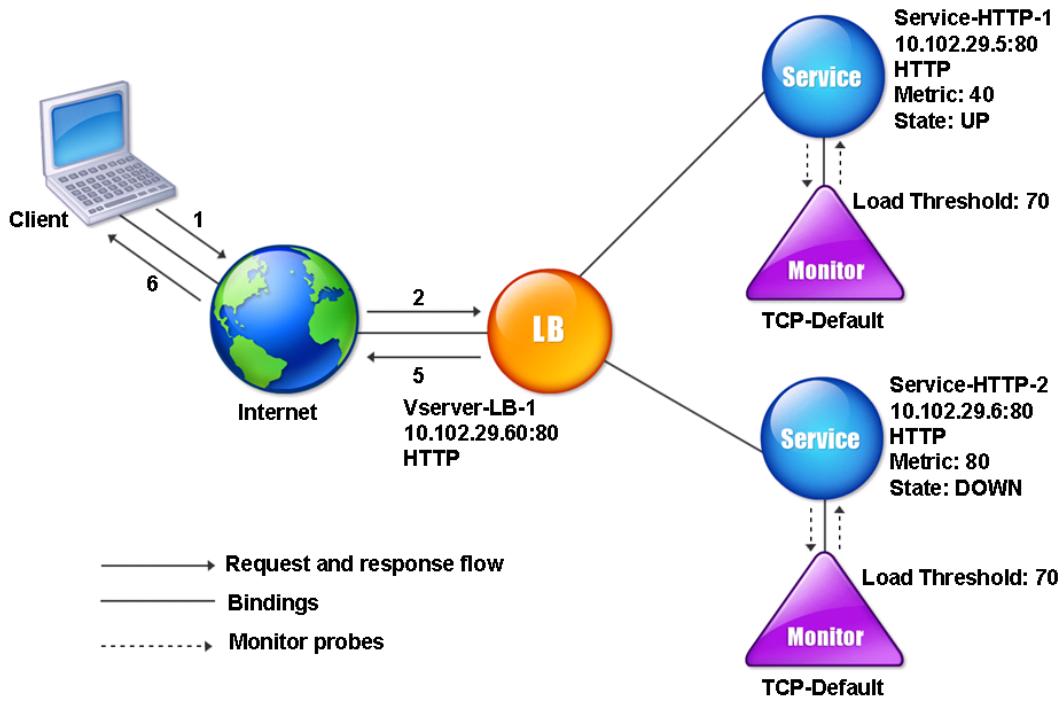
Understand load monitors

September 14, 2021

Load monitors use SNMP polled OIDs to calculate load. The load monitor uses the IP address of the service to which it is bound (the destination IP address) for polling. It sends an SNMP query to the service, specifying the OID for a metric. The metrics can be CPU, memory, or number of server connections. The server responds to the query with a metric value. The metric value in the response is compared with the threshold value. The Citrix ADC appliance considers the service for load balancing only if the metric is less than the threshold value. The service with the lowest load value is considered first.

The following diagram illustrates a load monitor configured for the services described in the basic load balancing setup discussed in Setting Up Basic Load Balancing.

Figure 1. Operation of Load Monitors



Note: The load monitor does not determine the state of the service. It only enables the appliance to consider the service for load balancing.

After you configure the load monitor, you must then configure the metrics that the monitor will use. For load assessment, the load monitor considers server parameters known as metrics, which are defined within the metric tables in the appliance configuration. Metric tables can be of two types:

- **Local.** By default, this table exists in the appliance. It consists of four metrics: connections, packets, response time, and bandwidth. The appliance specifies these metrics for a service, and SNMP queries are not originated for these services. These metrics cannot be changed.
- **Custom.** A user-defined table. Each metric is associated with an OID.

By default, the appliance generates the following tables:

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL
- FOUNDRY
- ALTEON

You can either add the appliance-generated metric tables, or you can add tables of your own choosing,

as shown in the following table. The values in the metric table are provided only as examples. In an actual scenario, consider the real values for the metrics.

Metric name	OIDs	Weight	Threshold
CPU	1.2.3.4	2	70
Memory	4.5.6.7	3	80
Connections	5.6.7.8	4	90

To calculate the load for one or more metrics, you assign a weight to each metric. The default weight is 1. The weight represents the priority given to each metric. If the weight is high, the priority is high. The appliance chooses a service based on the SOURCEIPDESTIP hash algorithm.

You can also set the threshold value for each metric. The threshold value enables the appliance to select a service for load balancing if the metric value for the service is less than the threshold value. The threshold value also determines the load on each service.

Configure load monitors

September 14, 2021

To configure a load monitor, first create the load monitor. For instructions on creating a monitor, see Creating Monitors. Next, select, or create the metric table to define a set of metrics that determine the state of the server, and (if you create a metric table) bind each metric to the metric table.

To create a metric table by using the command line interface

At the command prompt, type the following commands:

```
add lb metricTable <metricTableName>

bind lb metricTable <metricTableName> <metric> <SNMPOID>
<!--NeedCopy-->
```

Example:

```
add metricTable Table-Custom-1

bind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11
<!--NeedCopy-->
```


To create a metric table and bind metrics to it by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Metric Tables** and create a metric table.
2. To bind metrics, click **Bind** and specify a metric and an SNMP OID.

Unbind metrics from a metrics table

September 14, 2021

You can unbind metrics from a metrics table if the metrics need to be changed, or if you want to remove the metrics table entirely.

To unbind metrics from a metric table by using the command line interface

At the command prompt, type:

```
unbind lb metricTable <metricTable> <metric>
<!--NeedCopy-->
```

Example:

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
<!--NeedCopy-->
```

To unbind metrics from a metric table by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Metric Tables**.
2. Open a metric table, select a metric, and click **Delete**.

You can view the detail of all configured metric tables, such as name and type, to determine whether the metric table is internal or created and configured.

Configure reverse monitoring for a service

September 14, 2021

A reverse monitor marks a service as DOWN if the probe criteria are satisfied and UP if they are not satisfied. For example, if you want a backup service to receive traffic only when the primary service is DOWN, you can bind a reverse monitor to the secondary service but configure it to probe the primary service.

The Citrix ADC appliance supports the following reverse monitors:

- HTTP
- ICMP
- TCP (from release 11.1 build 49.x)

Configuring HTTP Reverse Monitoring for a Service

The following table describes the conditions of HTTP direct and reverse monitoring for a service:

Condition	Direct	Reverse
Connection not established.	Fail	Fail
HTTP response code matches the probe's specifications.	Success	Fail
HTTP response code does not match the probe's specifications.	Fail	Success
Probe timed out.	Fail	Fail

To configure HTTP reverse monitoring for a service by using the CLI

At the command prompt, type:

```
add lb monitor <Monitor_Name> HTTP -respCode 200 -httpRequest "HEAD /"
    -destIP <Primary_Service_IP_Address> -destPort 80 -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

Configuring ICMP Reverse Monitoring for a Service

The following table describes the conditions of ICMP direct and reverse monitoring for a service:

Condition	Direct	Reverse
ICMP echo reply is received.	Success	Fail
Probe timed out.	Fail	Success

To configure ICMP reverse monitoring for a service by using the CLI

At the command prompt, type:

```
add lb monitor <Monitor_Name> PING -destIP <Primary_Service_IP_Address>
    -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

Configuring TCP Reverse Monitoring for a Service

If a direct TCP monitor receives a RESET in response to a monitor probe, the service is marked DOWN. However, if a reverse TCP monitor receives a RESET response, the probe is considered successful, and the service is marked UP.

The following table describes the conditions of TCP reverse monitoring for a service:

Condition	Direct	Reverse
TCP connection is established.	Success	Fail
Probe timed out.	Fail	Fail
Response to probe is RESET.	Fail	Success

To configure TCP reverse monitoring for a service by using the CLI

At the command prompt, type:

```
add lb monitor <Monitor_Name> TCP  - destip <Primary_Service_IP_Address>
    -destport <primary_service_port>  - reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

To configure reverse monitoring by using the GUI

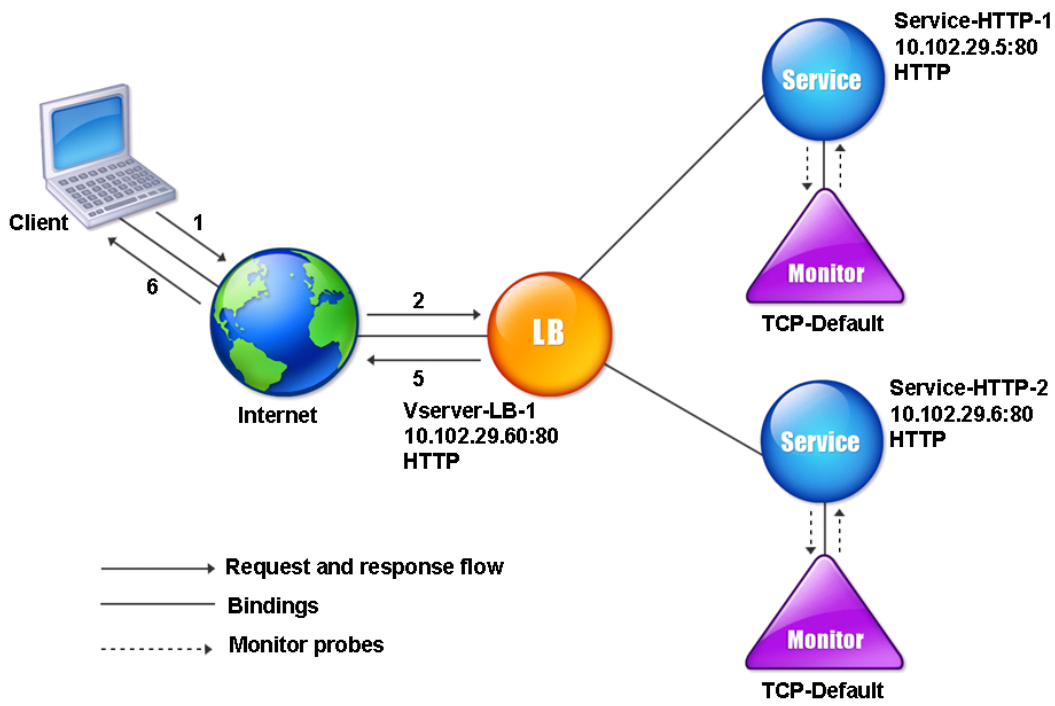
1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**.
2. Create an HTTP, ICMP, or TCP monitor and select **Reverse**.

Configure monitors in a load balancing setup

September 14, 2021

To configure monitors on a website, you first decide whether to use a built-in monitor or create your own monitor. If you create a monitor, you can choose between creating a monitor based on a built-in monitor, or creating a custom monitor that uses a script that you write to monitor the service. For more information about creating custom monitors, see Custom Monitors. Once you have chosen or created a monitor, you then bind it to the appropriate service. The monitor names can be up to 255 characters in length. The following conceptual diagram illustrates a basic load balancing setup with monitors.

Figure 1. How monitors operate



As shown, each service has a monitor bound to it. The monitor probes the load balanced server via its service. As long as the load balanced server responds to the probes, the monitor marks it UP. If the load balanced server fails to respond to the designated number of probes within the designated time period, the monitor marks it DOWN.

This section includes the following details:

Create monitors

September 14, 2021

The Citrix ADC appliance provides a set of built-in monitors. It also allows you to create custom monitors, either based on the built-in monitors or from scratch.

To create a monitor by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> [<interval>]

add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2
<!--NeedCopy-->
```

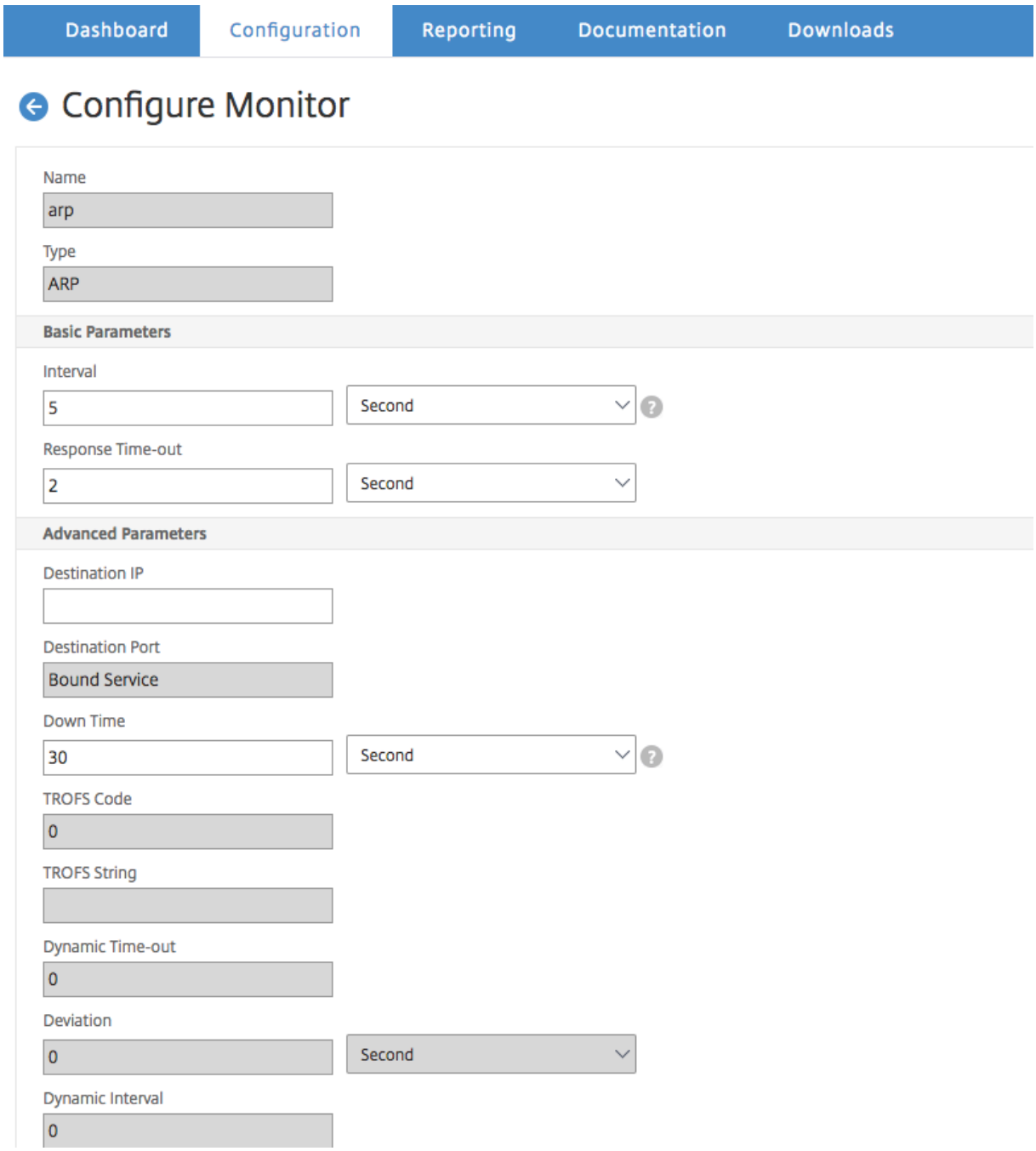
To create a monitor by using the GUI

1. Navigate to **Traffic Management**> **Load Balancing**> **Monitors**.
2. Click **Add** and create a monitor type that meets your requirement.

The Create Monitor screen contains two sections, **Basic Parameters** and **Advanced Parameters**.

Depending on the monitor type, the **Basic Parameters** section contains the parameters that must be set for each monitor. The **Advanced Parameters** section contains the parameters that can be used in advanced use cases.

The following figure is a sample of a Create Monitor page of the ARP monitor type.



Note

Prior to NetScaler release 12.0 build 56.20, Basic Parameters and Advanced Parameters are named Standard Parameters and Special Parameters respectively.

Configure monitor parameters to determine the service health

September 14, 2021

You can configure the following monitoring parameters to mark a service as DOWN based on the monitoring probes.

Retries

Maximum number of probes to send to establish the state of a service for which a monitoring probe fails.

failureRetries

Number of retries that must fail, out of the number specified for the Retries parameter, for a service to be marked as DOWN. For example, if the Retries parameter is set to 10 and the Failure Retries parameter is set to 6, out of the 10 probes sent, at least six probes must fail if the service is to be marked as DOWN.

alertRetries

Number of consecutive probe failures after which the appliance generates an SNMP trap called monProbeFailed.

Setting alertRetries to a value higher than the Retries value

The alertRetries parameter, which specifies the maximum number of consecutive monitoring-probe failures after which the Citrix ADC appliance generates an SNMP trap called monProbeFailed, can now be set to a value higher than the Retries value (which specifies the maximum number of probes to send to establish the state of a service for which a monitoring probe failed). If the alertRetries value is higher than the Retries value, the SNMP trap is not sent until after the service is DOWN.

For example, if you set Retries to 3, alertRetries to 12, and the time interval to 5 seconds, the service is marked DOWN after 15 seconds (3*5), but no alert is generated. If the monitor probes are still failing after 60 seconds (12*5), the Citrix ADC appliance generates a monProbeFailed trap. If a probe succeeds at some time between 15 and 60 seconds, the service is marked UP and no alert is generated.

Setting the alertRetries value to a value higher than the Retries value helps in generating only genuine alerts and avoid false positives during scheduled restarts.

To set the alertRetries parameter value to a higher value than the Retries value by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> [-retries <integer>] [-alertRetries <
    integer>]
<!--NeedCopy-->
```

Example:

add lb monitor monitor-HTTP-1 HTTP -retries 3 -alertRetries 12

To set the alertRetries parameter value to a higher value than the Retries value by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Monitors**.
2. Click **Add** to add a new monitor or select an existing monitor and click **Edit**.
3. In the **Retries** box, type the value for the Retries parameter.
4. In the **SNMP Alert Retries** box, type the value for the `alertRetries` parameter.

Bind monitors to services

September 14, 2021

After creating a monitor, you bind it to a service. You can bind one or multiple monitors to a service. If you bind one monitor to a service, that monitor determines whether the service is marked UP or DOWN.

If you bind multiple monitors to a service, the Citrix ADC appliance checks the state of all the monitors and then decides the state of the service. You can configure different weights to a monitor. The weight of a monitor specifies how much that monitor contributes to designating the service as UP or DOWN. A monitor with a greater weight has a higher preference in marking the service UP or DOWN. Default weight is 1. Therefore, even if one of the monitors fails, the service is marked as DOWN. For more information, see Set a threshold value for the monitors bound to a service.

Note: The destination IP address of a monitor probe can be different than the server IP address and port.

To bind a monitor to a service by using the CLI

At the command prompt, type:


```
bind service <name> (-monitorName <string>)
<!--NeedCopy-->
```

Example:

```
bind service s1 -monitorName tcp
<!--NeedCopy-->
```

To bind a monitor to a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Open the service, and add a monitor.

Modify monitors

September 14, 2021

You can modify the settings for any monitor that you created.

Note: Two sets of parameters apply to monitors: those that apply to all monitors, regardless of type, and those that are specific to a monitor type. For information on parameters for a specific monitor type, see the description for that type of monitor.

To modify an existing monitor by using the CLI

At the command prompt, type:

```
set lb monitor <monitorName> <type> -interval <interval> -resptimeout <
    resptimeout>
<!--NeedCopy-->
```

Example:

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
<!--NeedCopy-->
```

To modify an existing monitor by using the GUI

Navigate to **Traffic Management > Load Balancing > Monitors**, and open a monitor to modify.

Enable and disable monitors

September 14, 2021

By default, monitors bound to services and service groups are enabled. When you enable a monitor, the monitor begins probing the services to which it is bound. If you disable a monitor bound to a service, the state the service is determined using the other monitors bound to the service. If the service is bound to only one monitor, and if you disable the monitor, the state of the service is determined using the default monitor.

To enable a monitor by using the CLI

At the command prompt, type:

```
enable lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
enable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

To enable a monitor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Select a monitor, and from the Action list, select Enable or Disable.

To disable a monitor by using the CLI

At the command prompt, type:

```
disable lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
disable lb mon monitor-HTTP-1
<!--NeedCopy-->
```


Unbind monitors

September 14, 2021

You can unbind monitors from a service and service group. When you unbind a monitor from the service group, the monitors are unbound from the individual services that constitute the service group. When you unbind a monitor from a service or a service group, the monitor does not probe the service or the service group.

Note: When you unbind all user-configured monitors from a service or a service group, the default monitor is bound to the service and the service group. The default monitors then probes the service or the service groups.

To unbind a monitor from a service by using the CLI

At the command prompt, type:

```
unbind lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
unbind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

To unbind a monitor from a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service to modify.
2. Click in the Monitors section, select a monitor, and click **Unbind**.

Remove monitors

September 14, 2021

After you unbind a monitor that you created from its service, you can remove that monitor from the Citrix ADC configuration. (If a monitor is bound to a service, it cannot be removed.)

Note: When you remove monitors bound to a service, the default monitor is bound to the service. You cannot remove default monitors.

To remove a monitor by using the CLI

At the command prompt, type:

```
rm lb monitor <monitorName> <type>
<!--NeedCopy-->
```

Example:

```
rm lb monitor monitor-HTTP-1 HTTP
<!--NeedCopy-->
```

To remove a monitor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Select a monitor, and click **Delete**.

View monitors

September 14, 2021

You can view the services and service groups that are bound to a monitor. You can verify the settings of a monitor to troubleshoot your Citrix ADC configuration. The following procedure describes the steps to view the bindings of a monitor to the services and service groups.

To view monitor bindings by using the CLI

At the command prompt, type:

```
show lb monbindings <MonitorName>
<!--NeedCopy-->
```

Example:

```
show lb monbindings monitor-HTTP-1
<!--NeedCopy-->
```

To view monitor bindings by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Select a monitor, and in the Action list, click **Show Bindings**.

To view monitors by using the CLI

At the command prompt, type:

```
show lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
show lb mon monitor-HTTP-1
<!--NeedCopy-->
```

To view monitors by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**. The details of the available monitors appear in the Monitors pane.

Close monitor connections

September 14, 2021

The Citrix ADC appliance sends probes to the services through the monitors bound to the services. By default, the monitor on the appliance and the physical server follow the complete handshake procedure even for monitor probes. However, this procedure adds overhead to the monitoring process and might not be always necessary.

For the TCP type monitor, you can configure the appliance to close a monitor-probe connection after receiving SYN-ACK from the service. To do so, set the value of the monitorConnectionClose parameter to RESET. If you want the monitor-probe connection to go through the complete procedure, set the value to FIN.

Note: The monitorConnectionClose setting is applicable only for type TCP and TCP-Default monitors.

To configure monitor-connection closure by using the command line interface:

At the command prompt, type:

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
<!--NeedCopy-->
```

Example

```
set lb parameter -monitorConnectionClose RESET
<!--NeedCopy-->
```


To configure monitor-connection closure by using the configuration utility:

1. Navigate to **Traffic Management > Load Balancing > Configure Load Balancing Parameters**.
2. Select **FIN** or **Reset**.

Closing Monitor Connections at the Service or Service Group Level

You can also configure the appliance to close a monitor-probe connection at the service and service group level by setting the monConnectionClose parameter. If this parameter is not set, the monitor connection is closed by using the value set in the global load balancing parameters. If this parameter is set at the service or service group level, the monitor connection is closed by sending a connection termination message, with the FIN or RESET bit set, to the service or service group.

To configure monitor-connection closure at the service level by using the CLI

At the command prompt, type:

```
set service <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

To configure monitor-connection closure at the service group level by using the CLI

At the command prompt, type:

```
 set serviceGroup <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

To configure monitor-connection closure at the service level by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Add or edit a service, and in **Basic Settings**, set the **Monitoring Connection Close Bit**.

To configure monitor-connection closure at the service group level by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Add or edit a service group, and in **Basic Settings**, set the **Monitoring Connection Close Bit**.

> **Note:** For closing a monitor-probe connection using global load balancing parameters, you can configure monitorConnectionClose to FIN or RESET. When you configure the monitorConnectionClose parameter to;
>
> - FIN: The appliance performs a complete TCP handshake.

> - RESET: The appliance closes the connection after receiving the SYN-ACK from the service.
>
> In the lighter version of Citrix ADC CPX, the monitorConnectionClose parameter value is set to RESET by default and cannot be changed to FIN at the global level. However, you can change the monitorConnectionClose parameter to FIN at the service level.

Ignore the upper limit on client connections for monitor probes

September 14, 2021

Depending on considerations such as the capacity of a physical server, you can specify a limit on the maximum number of client connections made to any service. If you have set such a limit on a service, the Citrix ADC appliance stops sending requests to the service when the threshold is reached and resumes sending connections to the service after the number of existing connections falls to within the limits. You can configure the appliance to skip this check when it sends monitor-probe connections to a service.

Note: You cannot skip the maximum-client-connections check for an individual service. If you specify this option, it applies to all the monitors bound to all the services configured on the Citrix ADC appliance.

To set the Skip MaxClients for Monitor Connections option by using the CLI

At the command prompt, type:

```
set lb parameter -monitorSkipMaxClient (ENABLED|DISABLED)
<!--NeedCopy-->
```

Example:

```
set lb parameter -monitorSkipMaxClient enabled
<!--NeedCopy-->
```

To set the Skip MaxClients for Monitor Connections option by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Configure Load Balancing Parameters**.
2. Select **Skip MaxClients for Monitoring Connections**.

Manage a large scale deployment

September 14, 2021

The Citrix ADC appliance contains several features that are helpful when you are configuring a large load balancing deployment. Instead of configuring virtual servers and services individually, you can create groups of virtual servers and services. You can also create a range of virtual servers and services, and you can translate or mask virtual server and service IP addresses.

You can set persistence for a group of virtual servers. You can bind monitors to a group of services. Creating a range of virtual servers and services of identical type allows you to set up and configure those servers in a single procedure. This significantly shortens the time required to configure those virtual servers and services.

By translating or masking IP addresses, you can take down virtual servers and services. You can then make changes to your infrastructure, without extensive reconfiguration of your service and virtual server definitions.

Ranges of virtual servers and services

September 14, 2021

When you configure load balancing, you can create ranges of virtual servers and services, eliminating the need to configure virtual servers and services individually. For example, you can use a single procedure to create three virtual servers with three corresponding IP addresses. When more than one argument uses a range, the ranges must be of the same size.

The following are the types of ranges you can specify when adding services and virtual servers to your configuration:

- **Numeric ranges.** Instead of typing a single number, you can specify a range of consecutive numbers.

  For example, you can create a range of virtual servers by specifying a starting IP address, such as 10.102.29.30, and then typing a value for the last byte that indicates the range, such as 34. In this example, five virtual servers are created with IP addresses that range between 10.102.29.30 and 10.102.29.34.

  Note: The IP addresses of the virtual servers and services must be consecutive.

- **Alphabetic ranges.** Instead of typing a literal letter, you can substitute a range for any single letter, for example, [C-G]. This results in all letters in the range being included, in this case C, D, E, F, and G.

For example, if you have three virtual servers named `Vserver-x`, `Vserver-y`, and `Vserver-z`, instead of configuring them separately, you can type `vserver [x-z]` to configure them all.

Creating a range of virtual servers

To create range of virtual servers by using the CLI

At the command prompt, type one of the following commands:

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@[<rangeValue>]> <protocol> <IPAddress[<rangeValue
    >]> [<port>]
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

OR

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80

vserver "vserverP" added

vserver "vserverQ" added

vserver "vserverR" added

Done
<!--NeedCopy-->
```

To create range of virtual servers by using the CLI

At the command prompt, type one of the following commands:

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@**[**<rangeValue>**]**> <protocol> <IPAddress[<
    rangeValue>]> [<port>]
<!--NeedCopy-->
```


Example:

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

OR

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
Done
<!--NeedCopy-->
```

To create range of virtual servers by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Add a virtual server, and specify a range.

Creating a range of services

If you specify a range for the service name, specify a range for the IP address too.

To create range of services by using the CLI

At the command prompt, type the command:

```
add service <name>@ <IP>@ <protocol> <port>
<!--NeedCopy-->
```

Example:

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
 Done
<!--NeedCopy-->
```


Configure service groups

September 14, 2021

Configuring a service group enables you to manage a group of services as easily as a single service. For example, if you enable or disable any option, such as compression, health monitoring, or graceful shutdown, for a service group, the option gets enabled for all the members of the service group.

After creating a service group, you can bind it to a virtual server, and you can add services to the group. You can also bind monitors to service groups.

The members of a service group are identified by IP address or server name.

Using domain-name based service (DBS) group members is advantageous because you need not reconfigure the member on the Citrix ADC appliance if the IP address of the member changes. The appliance automatically senses such changes through the configured name server. This feature is useful in cloud scenarios, where the service provider can change a physical server or change the IP address for a service. If you specify a DBS group member, the appliance learns the IP address dynamically.

You can bind both IP-based and DBS members to the same service group.

Note: If you use DBS service group members, make sure that either a name server is specified or a DNS server is configured on the Citrix ADC appliance. A domain name is resolved into an IP address only if the corresponding address record is present on the appliance or the name server.

Create service groups

You can configure up to 8192 service groups on the Citrix ADC appliance.

To create a service group by using the command line

At the command prompt, type:

```
add servicegroup <ServiceGroupName> <Protocol>
<!--NeedCopy-->
```

Example:

```
add servicegroup Service-Group-1 HTTP
<!--NeedCopy-->
```

To create a service group by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**, and add a service group.

Bind a service group to a virtual server

When you bind a service group to a virtual server, the member services are bound to the virtual server.

To bind a service group to a virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name>@ <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

To bind a service group to a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **Service Groups**.

Bind a member to a service group

Adding services to a service group enables the service group to manage the servers. You can add the servers to a service group by specifying the IP addresses or the names of the servers.

In the GUI, if you want to add a domain-name based service group member, select **Server Based**.

With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.

To add members to a service group by using the command line interface

To configure a service group, at the command prompt, type:

```
bind servicegroup <serviceGroupName> (<IP>@ | <serverName>) <port>
<!--NeedCopy-->
```

Examples:

```
bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a
    :888b 80
```



```
4
5 bind servicegroup CitrixEdu s1.citrite.net
6 <!--NeedCopy-->
```

To add members to a service group by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Service Groups** and open a service group.
2. Click in the Service Group section, and do one of the following:
   - To add an IP based service group member, select IP Based.
   - To add a server-name based service group member, select Server Based.

   If you want to add a domain-name based service group member, select **Server Based**. With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.
3. If adding a new IP based member, in the IP Address text box, type the IP address. If the IP address uses IPv6 format, select the IPv6 check box and then enter the address in the IP Address text box

   Note: You can add a range of IP addresses. The IP addresses in the range must be consecutive. Specify the range by entering the starting IP address in the IP Address text box (for example, 10.102.29.30). Specify the end byte of the IP address range in the text box under Range (for example, 35). In the Port text box type the port (for example, 80), and then click Add.
4. Click Create.

Bind a monitor to a service group

When you create a service group, the default monitor of the type appropriate for the group is automatically bound to it. Monitors periodically probe the servers in the service group to which they are bound and update the state of the service groups.

You can bind a different monitor of your own choice to the service group.

To bind a monitor to a service group by using the command line interface

At the command prompt, type:

```
1 bind serviceGroup <serviceGroupName> -monitorName <string> -monState (
     ENABLED | DISABLED)
2 <!--NeedCopy-->
```

Example:


```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

To a bind monitor to a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Open a service group and, in Advanced Settings, click **Monitors**.

Retain the original state of a service group member after disabling and enabling a virtual server

From build 64.x, a new global option, –retainDisableServer, enables you to retain a service-group member's state when a server is disabled and reenabled.

Previously, a member's state would change from DISABLED to ENABLED under the following set of conditions:

- Two applications are deployed on the same port on a virtual server.
- Two service groups with a common member are bound to this virtual server, and the common member is enabled in one group and disabled in the other.
- The server is disabled and then reenabled.

Under these conditions, disabling the server disables all the service group members, and reenabling the server enables all the members, by default, regardless of their earlier states. To bring the members back to the original states, you must manually disable those members in the service group. This is a cumbersome task and prone to errors.

Manage service groups

September 14, 2021

You can change the settings of the services in a service group, and you can perform tasks such as enabling, disabling, and removing service groups. You can also unbind members from a service group. For more information about service groups, see Configure service groups.

Modify a service group

You can modify the attributes of service group members. You can set several attributes of the service group, such as maximum client, Sure Connect, and compression. The attributes are set on the indi-

vidual servers in the service group. You cannot set parameters on the service group such as transport information (IP address and port), weight, and server ID.

Note: A parameter you set for a service group is applied to the member servers in the group, not to individual services.

To modify a service group by using the command line interface

At the command prompt, type the following command with one or more of the optional parameters:

```
set servicegroup <serviceGroupName> [-type <type>] [-maxClient <
    maxClient>] [-maxReq <maxReq>] [-cacheable (YES|NO)] [-cip (ENABLED|
    DISABLED)] [-cipHeader <cipHeader>] [-usip (YES|NO)] [-sc (ON|OFF)]
    [-sp (ON|OFF)] [-cltTimeout <cltTimeout>] [-svrTimeout <svrTimeout>]
     [-cka (YES|NO)] [-TCPB (YES|NO)] [-CMP (**YES**|**NO**)] [-
    maxBandwidth <maxBandwidth>] [-maxThreshold <maxThreshold>] [-state
    (ENABLED|DISABLED)] [-downStateFlush (ENABLED|DISABLED)
<!--NeedCopy-->
```

Example:

```
set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES
<!--NeedCopy-->
```

To modify a service group by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**, and open the service group to modify.

Remove a service group

When you remove a service group, the servers bound to the group retain their individual settings and continue to exist on the Citrix ADC appliance.

To remove a service group by using the command line interface

At the command prompt, type:


```
rm servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
rm servicegroup Service-Group-1
<!--NeedCopy-->
```

To remove a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and click **Delete**.

Unbind a member from a service group

When you unbind a member from the service group, the attributes set on the service group no longer apply to the member that you unbound. The member services retain its individual settings, however, and continue to exist on the Citrix ADC appliance.

To unbind members from a service group by using the command line interface

At the command prompt, type:

```
unbind servicegroup <serviceGroupName> <IP>@ [<port>]
<!--NeedCopy-->
```

Example:

```
unbind servicegroup Service-Group-1 10.102.29.30 80
<!--NeedCopy-->
```

To unbind members from a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Open a service group, and click in the Service Group Members section.
3. Select a service group member, and click **Unbind**.

Unbind a service group from a virtual server

When you unbind a service group from a virtual server, the member services are unbound from the virtual server and continue to exist on the Citrix ADC appliance.

To unbind a service group from a virtual server by using the command line interface

At the command prompt, type:

```
unbind lb vserver <name>@ <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
unbind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

To unbind a service group from a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open the virtual server, and click in the Service Group section.
3. Select the service group, and click **Unbind**.

Unbind monitors from service groups

When you unbind a monitor from a service group, the monitor that you unbound no longer monitors the individual services that constitute the group.

To unbind a monitor from a service group using the command line interface

At the command prompt, type:

```
unbind serviceGroup <serviceGroupName> -monitorName <string>
<!--NeedCopy-->
```

Example:

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

To unbind a monitor from a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Open a service group, and click in the Monitors section.
3. Select a monitor, and click **Unbind**.

Enable or Disable a service group

When you enable a service group and the servers, the services belonging to the service group are enabled. Similarly, when a service belonging to a service group is enabled, the service group and the service are enabled. By default, service groups are enabled.

After disabling an enabled service, you can view the service using the configuration utility or the command line to see the amount of time that remains before the service goes DOWN.

To disable a service group by using the command line interface

At the command prompt, type:

```
disable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
disable servicegroup Service-Group-1
<!--NeedCopy-->
```

To disable a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and in the Action list, click **Disable**.

To enable a service group by using the command line interface

At the command prompt, type:

```
enable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
enable servicegroup Service-Group-1
<!--NeedCopy-->
```

To enable a service group by using the configuration utility

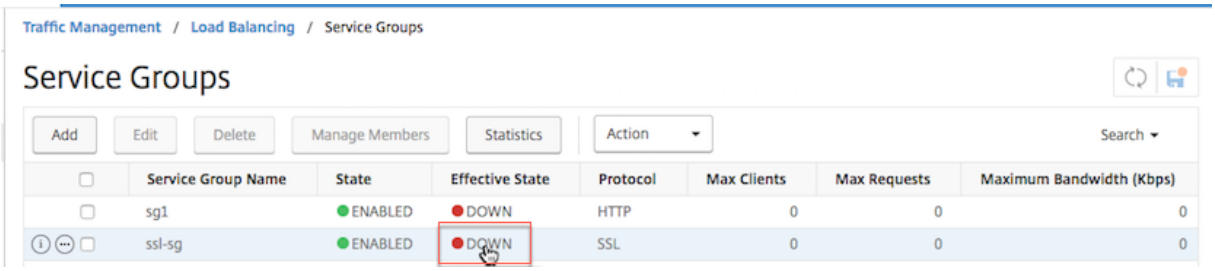
1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and in the Action list, click **Enable**.

View the status of service groups members

Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.

In the Service Groups page, the **Effective State** column displays the status of the service groups. Status UP/DOWN in the **Effective State** column is clickable. You can click the status and get the list of members along with their status in the same view. Select a member and click the **Monitor Details** button to view the reason for the status being DOWN.

Note: Before NetScaler release 12.0 build 56.20, the status in the **Effective State** column was not clickable.



Viewing the properties of a service group

You can view the following settings of the configured service groups:

- Name
- IP address
- State
- Protocol
- Maximum client connections
- Maximum requests per connection
- Maximum bandwidth
- Monitor threshold

Viewing the details of the configuration can be helpful for troubleshooting your configuration.

To view the properties of a service group by using the command line interface

At the command prompt, type one of the following commands to display the group properties or the properties and the group members:

```
show servicegroup <ServiceGroupName>

show servicegroup <ServiceGroupName> -includemembers
<!--NeedCopy-->
```

Example:


```
show servicegroup Service-Group-1
<!--NeedCopy-->
```

To view the properties of a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Click the arrow next to the service group.

Viewing service group statistics

You can view service-group statistical data, such as rate of requests, responses, request bytes, and response bytes. The Citrix ADC appliance uses the statistics of a service group to balance the load on the services.

To view the statistics of a service group by using the command line interface

At the command prompt, type:

```
stat servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
stat servicegroup Service-Group-1
<!--NeedCopy-->
```

To view the statistics of a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and click **Statistics**.

Load balancing virtual servers bound to a service group

In large-scale deployments, the same service group can be bound to multiple load balancing virtual servers. In such a case, instead of viewing each virtual server to see the service group it is bound to, you can view a list of all the load balancing virtual servers bound to a service group. You can view the following details of each virtual server:

- Name
- State

- IP address
- Port

To display the virtual servers bound to a service group by using the command line interface

At the command prompt, type the following command to display the virtual servers bound to a service group:

```
show servicegroupbindings <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
<!--NeedCopy-->
```

To display the virtual servers bound to a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and in the Action list, click **Show Bindings**.

Configure a desired set of service group members for a service group in one NITRO API call

September 14, 2021

Support is added to configure a desired set of service group members for a service group in one NITRO API call. A new API, Desired State API, is added to support this configuration. Using Desired State API, you can:

- Provide a list of service group members in a single PUT request on “servicegroup_servicegroupmemberlist_b resource.
- Provide their weight and state (optional) in that PUT request.

- Effectively synchronize the appliance configuration with deployment changes around application servers.

The Citrix ADC appliance compares the requested desired member set with the configured member set. Then, it automatically binds the new members and unbinds the members that are not present in the request.

Note:

- This feature is supported only for service groups of type `API`.
- You can only bind IP address based services using Desired State API, domain name based services are not allowed.
- Previously, only one service group member can be bound in a NITRO call.

> **Important**
>
> Desired State API for ServiceGroup membership is supported in Citrix ADC cluster deployment.

Use case: Synchronize deployment changes to Citrix ADC appliance in large scale deployments, such as Kubernetes

In large scale and highly dynamic deployments (for example Kubernetes), the challenge is to keep the appliance configuration up-to-date with the rate of change of deployments to accurately serve the application traffic. In such deployments, controllers (Ingress or E-W Controller) are responsible for updating ADC configuration. Whenever there are changes to deployment, `kube-api server` sends the effective set of endpoints through 'Endpoints event' to the controller. The controller uses the Read-Delta-Modify approach where it performs the following:

- Fetches the currently configured endpoint set (service group member set of a service group) for the service from ADC appliance.
- Compares the configured endpoint set with the set in the received event.
- Binds the new endpoints (service group members) or unbinds the deleted endpoints.

Because the rate of change and the size of services is high in this environment, this configuration method is not efficient and might delay configuration updates.

Desired State API solves the problem by accepting the intended member set for a service group in a single API, and effectively updates the configuration.

Create a service group of type API by using the CLI

At the command prompt, type;

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>]
```


Example:

```
add serviceGroup svg1 HTTP -autoScale API
```

You can configure the autoDisablegraceful and autoDisabledelay and autoScale parameters by using add serviceGroup or set serviceGroup command.

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>] [-autoDisablegraceful ( YES | NO)] [-autoDisabledelay <
    secs>]

add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (API |
    CLOUD | DISABLED| DNS |POLICY)]

set serviceGroup <serviceGroupName [-autoDisablegraceful ( YES | NO)]
    [-autoDisabledelay <secs>]

set serviceGroup <serviceGroupName [-autoScale (API |CLOUD | DISABLED|
    DNS |POLICY)]
```

Example:

```
add serviceGroup svg1 HTTP autoDisablegraceful YES -autoDisabledelay
    100

add serviceGroup svg1 HTTP -autoScale API

set serviceGroup svg1 -autoDisablegraceful  YES -autoDisabledelay 100

set serviceGroup svg1 -autoScale API
```

Arguments

autoDisablegraceful

Indicates graceful shutdown of the service. If this option is enabled, the appliance waits for all outstanding connections to this service to be closed before deleting the service. For clients that already have a persistent session on the system, new connections or requests continue to be sent to this service. The service member is deleted only if there are no outstanding connections. Default value: NO

autoDisabledelay

Indicates the time allowed (in seconds) for a graceful shutdown. During this period new connections or requests continue to be sent to this service for clients that already have a persistent session on

the system. Connections or requests from new clients that do not have persistence sessions on the system are not sent to the service. Instead, they are load balanced among other available services. After the delay time expires, the service member is deleted.

Autoscale API

Enables using Desired State API for binding the member set to an intended service group. You can set the service group from non-autoscale to Autoscale type of Desired State API, if all provided conditions match.

The set serviceGroup Autoscale command might fail if the existing member bindings meet any of these conditions:

- If the server bound to the service group is either a name server or a domain-based server.
- If the name of the server bound to the service group is an IP address, then it must match the actual server IP address. In the following example, the server name and server IP address do not match.
  - **CLI:** add server *IP address server name*
  - **Example:** add server 1.2.3.4 4.3.2.1
- If the loopback server name is anything other than 127.0.0.1 or 0000:0000:0000:0000:0000:0000:0000:0001.
- If you choose different types of Autoscale (Cloud, API, DNS, and Policy) in a set serviceGroup command and add serviceGroup command.

Important:

- The autoDisablegraceful and autoDisabledelay parameters are applicable only for the service groups of Autoscale type “API” and “CLOUD.”
- If the autoDisablegraceful or the autoDisabledelay parameters are not configured, then service members are deleted immediately.

Unbind a service group member gracefully

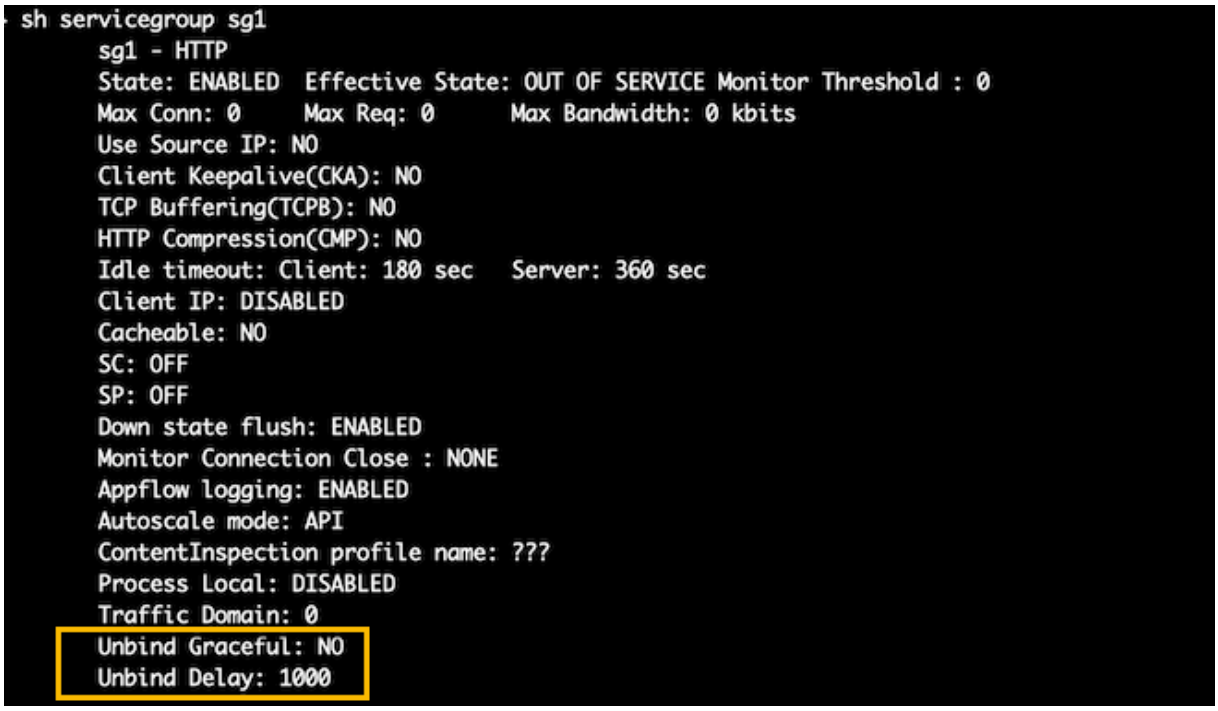
If any of the service group members is not in the desired state list, those members are gracefully unbound based on the `autoDisablegraceful` or `autoDisabledelay` parameter configuration.

- If one of these parameters is set, then the service group member is unbound gracefully.
- If none of these parameters are set, then the service group member is unbound immediately.

Note:

- Service group members identified for graceful unbind are displayed only when the show service group command is run.
- You cannot perform any operation (such as set, unset) on the service group member identified for graceful unbind.

The following figure displays a sample show service group command.

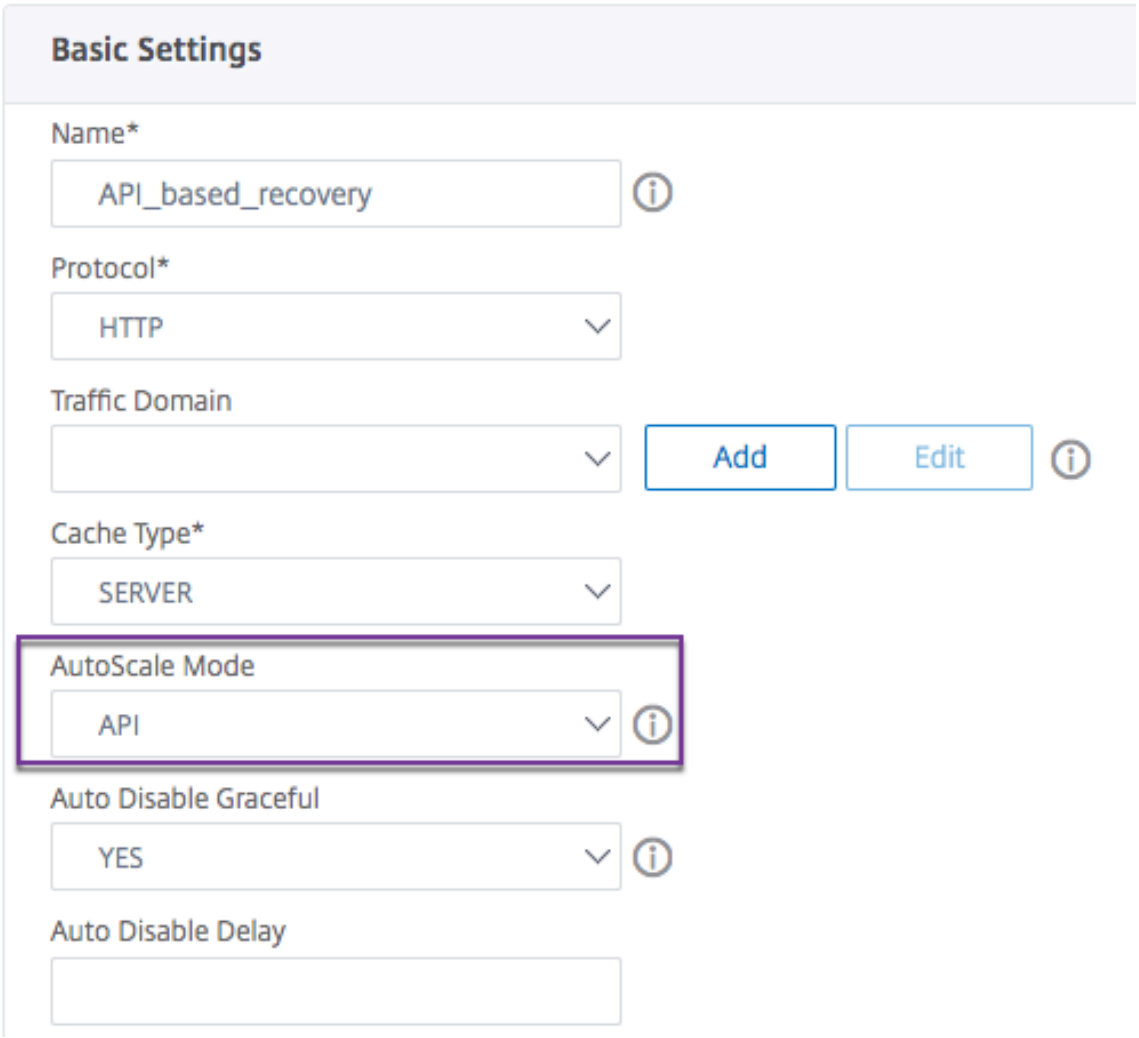


Create a service group of type API by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Service Groups**, and click **Add**.
2. In **AutoScale Mode**, select **API**.

Configure graceful shutdown or a time delay for an API type service group by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Service Groups**.



2. In **AutoScale Mode**, select **API**.
3. In **Auto Disable Graceful**, select **YES**.
4. In **Auto Disable Delay**, enter the wait time for a graceful shutdown.

Note: The **Auto Disable Graceful** or **Auto Display Delay** fields are enabled only if you select **API** or **CLOUD** in **AutoScale Mode**.

Configure automatic domain based service group scaling

September 14, 2021

A domain based service group consists of members whose IP addresses are obtained by resolving the domain names of servers that are bound to the service group. The domain names are resolved by a

name server whose details you configure on the appliance. A domain based service group can also include IP-address based members.

The process of name resolution for a domain based server might return more than one IP address. The number of IP addresses in the DNS response is determined by the number of address (A) records configured for the domain name, on the name server. Even if the name resolution process returns multiple IP addresses, only one IP address is bound to the service group. To scale up or scale down a service group, you need to manually bind and unbind other domain based servers to and from the service group, respectively.

However, you can configure a domain based service group to scale automatically based on the complete set of IP addresses returned by a DNS name server for a domain based server. To configure automatic scaling, when binding a domain based server to a service group, enable the automatic scaling option. Following are the steps for configuring a domain based service group that scales automatically:

- Add a name server for resolving domain names. For more information about configuring a name server on the appliance, see Adding a Name Server.
- Add a domain-based server. For information about adding a domain-based server, see Configuring a server object.
- Add a service group and associate the domain based server to the service group, with the Autoscale option set to DNS. For information about adding a service group, see Configuring Service Groups.

When a domain based server is bound to a service group and the automatic scaling option is set on the binding, a UDP monitor and a TCP monitor are automatically created and bound to the domain based server. The two monitors function as resolvers. The TCP monitor is disabled by default, and the appliance uses the UDP monitor to send DNS queries to the name server to resolve the domain name. If the DNS response is truncated (has the TC flag set to 1), the appliance falls back to TCP and uses the TCP monitor to send the DNS queries over TCP. Thereafter, the appliance continues to use only the TCP monitor.

The DNS response from the name server might contain multiple IP addresses for the domain name. With the automatic scaling option set, the appliance polls each of the IP addresses by using the default monitor, and then includes in the service group only those IP addresses that are up and available. After the IP address records expire, as defined by their time-to-live (TTL) values, the UDP monitor (or the TCP monitor, if the appliance has fallen back to using the TCP monitor) queries the name server for domain resolution and includes any new IP addresses in the service group. If an IP address that is part of the service group is not present in the DNS response, the appliance removes that address from the service group after gracefully closing existing connections to the group member, a process during which it does not allow any new connections to be established with the member. If a domain name that resolved successfully in the past results in an NXDOMAIN response, all the service group members associated with that domain are removed.

Static (IP-address based) members and dynamically scaling domain based members can coexist in a service group. You can also bind members with different domain names to a service group with the automatic scaling option set. However, each domain name associated with a service group must be unique within the service group. You must enable the automatic scaling option for each domain based server that you want to use for automatic service group scaling. If an IP address is common to one or more domains, the IP address is added to the service group only once.

> **Important**
>
> - DNS Autoscale is supported in a cluster deployment.
> - Path monitoring for Autoscale service groups is not supported in cluster deployment.

To configure a service group to scale automatically by using the command line interface

At the command prompt, type the following commands to configure the service group and verify the configuration:

```
add serviceGroup <serviceGroupName> -autoScale (YES | NO)

show serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

Example

In the following example, server1 is a domain based server. The DNS response contains multiple IP addresses. Five addresses are available and are added to the service group.

```
> add serviceGroup servGroup server1 80 -autoScale YES
 Done
> sh servicegroup servGroup
       servGroup - HTTP
        State: ENABLED  Monitor Threshold : 0
              . . .
              . . .
        1)   192.0.2.31:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1

               Monitor Name: tcp-default       State: UP
               Probes: 2       Failed [Total: 0 Current: 0]
               Last response: Success - TCP syn+ack received.

        2)   192.0.2.32:80   State: UP       Server Name: server1 (Auto
             scale)    Server ID: None Weight: 1
```



```
15
16                  Monitor Name: tcp-default       State: UP
17                  Probes: 2       Failed [Total: 0 Current: 0]
18                  Last response: Success - TCP syn+ack received.
19
20          3)   192.0.2.36:80   State: UP       Server Name: server1 (Auto
                scale)    Server ID: None Weight: 1
21
22                  Monitor Name: tcp-default       State: UP
23                  Probes: 2       Failed [Total: 0 Current: 0]
24                  Last response: Success - TCP syn+ack received.
25
26          4)   192.0.2.55:80   State: UP       Server Name: server1 (Auto
                scale)    Server ID: None Weight: 1
27
28                  Monitor Name: tcp-default       State: UP
29                  Probes: 2       Failed [Total: 0 Current: 0]
30                  Last response: Success - TCP syn+ack received.
31
32          5)   192.0.2.80:80   State: UP       Server Name: server1 (Auto
                scale)    Server ID: None Weight: 1
33
34                  Monitor Name: tcp-default       State: UP
35                  Probes: 2       Failed [Total: 0 Current: 0]
36                  Last response: Success - TCP syn+ack received.
37   Done
38  <!--NeedCopy-->
```

To configure a service group to scale automatically by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Create a service group, and set the Autoscale mode to DNS.

Overwriting TTL values

> **Note**: This option is supported from Citrix ADC 12.1 build 51.xx and later.

Citrix ADC appliance is configured to periodically query the DNS server for any update in the SRV record associated with the application during application startup. By default, the periodicity for this query depends on the TTL published in the SRV record. In microservice or cloud world application, deployments change more dynamically. As a result, proxies have to be quicker in absorbing any changes to application deployment. Therefore, users are recommended to set the domain based service TTL

parameter explicitly to a value that is lower than the SRV record TTL and is optimal for your deployment. You can overwrite the TTL value by two methods:

- While binding a member to the service group
- Setting the TTL value globally by using the set lb parameter command.

In case the TTL value is configured both while binding the service group member and also globally, then the TTL value specified while binding the service group member takes precedence.
If the TTL value is not specified either while binding a service group member or at the global level, the DBS monitor interval is derived from the TTL value in the DNS response.

Overwriting the TTL values using the CLI

- To overwrite the TTL value while binding, at the command prompt, type:

```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
   >])
<!--NeedCopy-->
```

Example:

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- To overwrite the TTL value globally, at the command prompt, type:

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

Example:

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

Overwriting the TTL values using the GUI

To overwrite the TTL value while binding:

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. In **Service Groups** page, select the service group that you have created and click **Edit**.
3. In **Load Balancing Service Groups** page, click **Service Group Members**.
4. In **Service Group Members Binding** page, select the server that you have created and click **Edit**.

5. In **Domain Based Service TTL**, enter the TTL value.

To overwrite the TTL value at the global level:

1. Navigate to **Traffic Management** > **Load Balancing** > **Change Load Balancing Parameters**.
2. In **Domain Based Service TTL**, enter the TTL value.

> **Note:** If the domain based server TTL value is set to 0, then the TTL value from the data packet is used.

Specifying different name servers for service group and domain name bindings

> **Note:** This option is supported from Citrix ADC 12.1 build 51.xx and later.

You can configure different name servers for different domain names in a specific group. Setting the nameServer parameter is optional while binding a DBS server to the service group. When a nameserver is not specified while binding a member to the service group, the globally configured nameserver is considered.

Specifying name servers while binding a server to service groups using the CLI

At the command prompt, type:

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
   ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

Example:

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
   -dbsTTL 10
<!--NeedCopy-->
```

Specifying name servers while binding a server to service groups using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. In **Service Groups** page, select the service group that you have created and click **Edit**.
3. In **Load Balancing Service Groups** page, click **Service Group Members**.
4. In **Service Group Members Binding** page, select the server that you have created and click **Edit**.
5. In **Name Server**, specify the nameserver name to which the query for the bound domain must be sent.

Service discovery using DNS SRV records

September 14, 2021

An SRV record (service record) is a specification of data in the Domain Name System that defines the location, that is the host name and the port number of servers for specified services. The record also defines the weight and priority of each server.

Example of an SRV record:

_http._tcp.example.com. 100 IN SRV 10 60 5060 a.example.com.

The following table describes each item in an SRV record:

Service	Protocol	Name	TTL	Class	SRV	Priority	Weight	Port	Target
HTTP	TCP	example.com	100	IN	SRV	10	60	5060	a.example.com

You can use the DNS SRV records to discover the service endpoints. Citrix ADC appliance is configured to periodically query the DNS servers with the SRV record associated with a service. On receiving the SRV record, each of the target host published in the SRV record is bound to a service group associated with the service. Each of the bindings inherits the port, priority, and weight from the SRV record. For each service deployment the user has to configure the Citrix ADC appliance once while bringing it up, thus making it a single touch deployment for applications.

Important: The weight of dynamically learned service group members cannot be modified using the CLI or the GUI.

Use case: Load balancing microservices

Applications are moving toward microservice architecture from monolithic architectures. Movement to microservice architecture along with back-end server autoscale solution, is making application deployment more dynamic. To support such a dynamic deployment, the proxies or ADC must be able to dynamically detect the back-end application or service instances and absorb them into the proxy configuration.
The service discovery using DNS SRV records feature aids configuration of the Citrix ADC appliance in such a dynamic deployment scenario. Application developers can use some of the orchestration platforms to deploy the application. Orchestration platforms while instantiating containers during application deployment, might not assign the protocol specific standard port for each of these containers. In such scenarios, discovering the port information becomes the key to configuring the Citrix ADC appliance. SRV records are helpful in such a scenario. SRV record parameters such as the priority and weight can be used for better load balancing of applications.

- Priority parameter can be used to dictate the priority of the server pool.

- Weight parameter can be used to dictate the capacity of the back-end service instances and hence can be used for weighted load balancing.
- Whenever there is a change in the back-end server pool, for example a back-end instance is removed from the pool, the instance is removed graciously only after all the existing client connections are honored.

> **Note:**
>
> - An A/AAAA records based service discovery, all resolved IP addresses have the same weight because you assign the weight to the domain being resolved.
> - If the weight in SRV response is greater than 100, then services are not created.

Priority based load balancing using SRV records

You can use SRV records to perform priority-based load balancing. The priority based server pool can be an alternative for the backup virtual servers. The ns.conf file requires minimal configuration compared to the backup virtual servers.

In priority based load balancing using SRV records, a priority number is assigned to each of the server pool. The least number has the highest priority. One of the servers in the highest priority pool is selected for load balancing based on the server's health and availability. If all the servers in the highest priority server pool are down, then the servers that have the next highest priority are selected for load balancing. However, if the servers in the highest priority server pool are up again, then the servers are selected from the highest priority pool again.

Switching from one priority server pool to another server pool occurs graciously by bleeding the existing client transactions. Therefore, the current clients do not see any break in the application access.

To enable querying for SRV records using the CLI

Perform the following tasks to enable querying for SRV records:

1. Create a server by specifying the query type parameter as SRV.

   At the command prompt, type:

   ```
   add server <name> <domain> [-queryType <queryType>])
   <!--NeedCopy-->
   ```

   Example:

   ```
   add server web_serv example.com -queryType SRV
   <!--NeedCopy-->
   ```

   Note:

- By default, IPv4 queries are sent. To send IPv6 queries, you must enable the IPv6 domain.
- The SRV target domain name must not exceed 127 characters.

2. Create a service group with the autoscale mode as DNS.

   At the command prompt, type:

```
add serviceGroup <serviceGroupName> <serviceType> [-autoScale <
   autoScale>]
<!--NeedCopy-->
```

   Example:

```
add servicegroup svc_grp_1 http -autoscale dns
<!--NeedCopy-->
```

3. Bind the server created in step 1 to the service group as a member.

   At the command prompt, type:

```
bind serviceGroup <serviceGroupName> <serverName>
<!--NeedCopy-->
```

   Example:

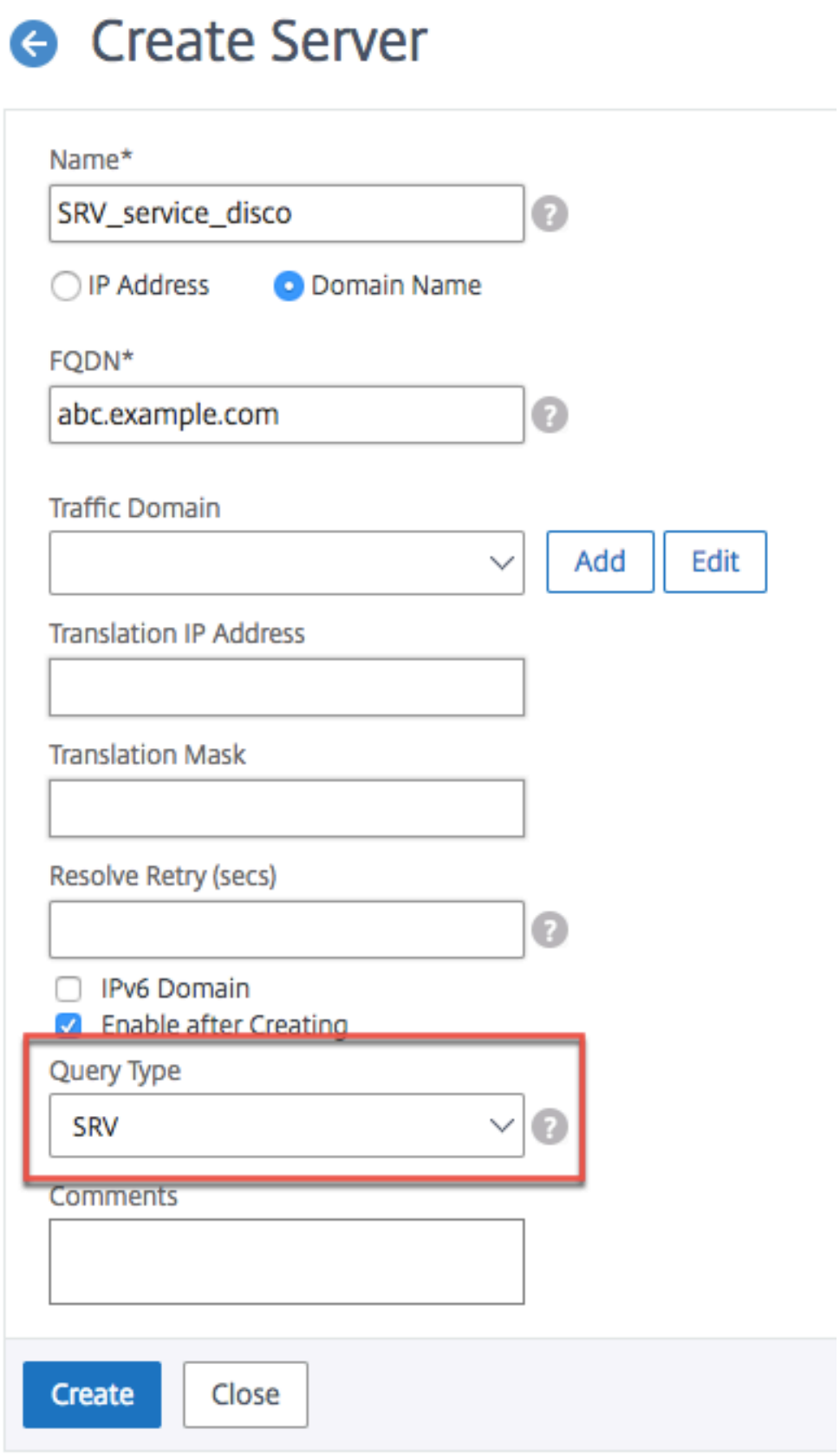
```
bind servicegroup svc_grp_1 web_serv
<!--NeedCopy-->
```

> **Note:**
>
> - When binding servers to service group members, you do not have to enter the port number for SRV server types. In case you specify a port number for SRV server type, an error message appears.
> - You can optionally specify a name server and a TTL value while binding a server to the service group.

To enable querying for SRV records using the GUI

Create a server

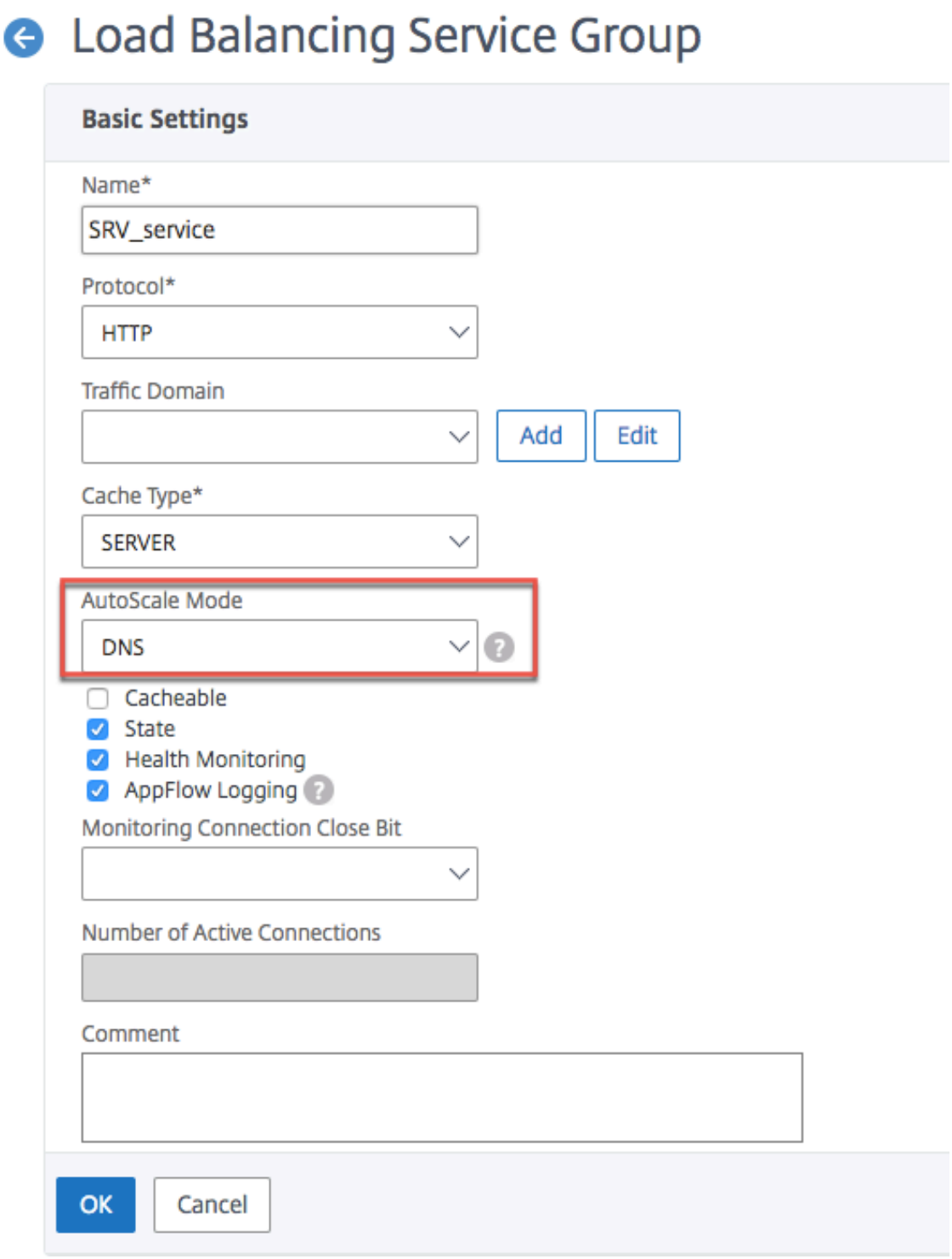
1. Navigate to **Traffic Management** > **Load Balancing** > **Servers** and click **Add**.



2. In **Create Server** page, select domain name.
3. Enter the details of all the required parameters.
4. In **Query Type**, select **SRV**.
5. Click **Create**.

Create a service group with autoscale mode as DNS

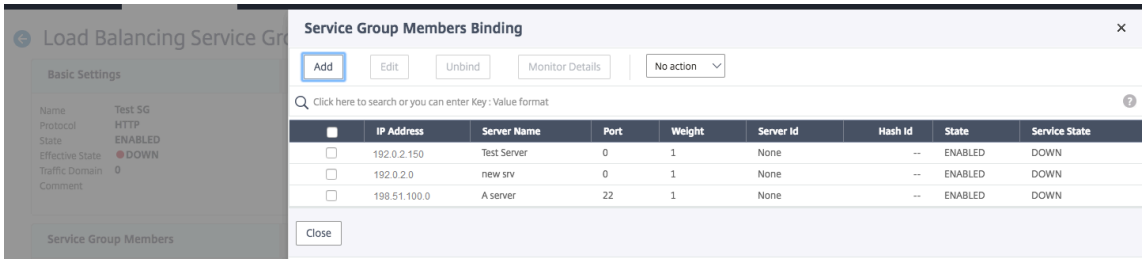
1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. In **Load Balancing Service Group** page, enter details of all the required parameters.
3. In **AutoScale Mode**, select **DNS**.



4. Click **OK**.

Bind server to the service group member

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. In **Service Groups** page, select the service group that you have created and click **Edit**.
3. In **Load Balancing Service Groups** page, click **Service Group Members**.
4. In **Service Group Members Binding** page, select the server that you have created and click **Close**.



Note:

- While binding, you do not have to enter the port number for SRV server types. In case you enter a port number for SRV server type, an error message appears.
- You can optionally specify a name server and a TTL value while binding a server to the service group.

Overwriting TTL values

Citrix ADC appliance is configured to periodically query the DNS server for any update in the SRV record associated with the application during application startup. By default, the periodicity for this query depends on the TTL published in the SRV record. In microservice or cloud world application, deployments change more dynamically. As a result, proxies have to be quicker in absorbing any changes to application deployment. Therefore, users are recommended to set the domain based service TTL parameter explicitly to a value that is lower than the SRV record TTL and is optimal for your deployment. You can overwrite the TTL value by two methods:

- While binding a member to the service group
- Setting the TTL value globally by using the set lb parameter command.

In case the TTL value is configured both while binding the service group member and also globally, then the TTL value specified while binding the service group member takes precedence.
If the TTL value is not specified either while binding a service group member or at the global level, the DBS monitor interval is derived from the TTL value in the DNS response.

Overwriting the TTL values using the CLI

- To overwrite the TTL value while binding, at the command prompt, type:


```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

Example:

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- To overwrite the TTL value globally, at the command prompt, type:

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

Example:

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

Overwriting the TTL values using the GUI

To overwrite the TTL value while binding:

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. In **Service Groups** page, select the service group that you have created and click **Edit**.
3. In **Load Balancing Service Groups** page, click **Service Group Members**.
4. In **Service Group Members Binding** page, select the server that you have created and click **Edit**.
5. In **Domain Based Service TTL**, enter the TTL value.

To overwrite the TTL value at the global level:

1. Navigate to **Traffic Management** > **Load Balancing** > **Change Load Balancing Parameters**.
2. In **Domain Based Service TTL**, enter the TTL value.

> **Note**: If the domain based server TTL value is set to 0, then the TTL value from the data packet is used.

Specifying different name servers for service group and domain name bindings

You can configure different name servers for different domain names in a specific group. Setting the nameServer parameter is optional while binding a DBS server to the service group. When a name-

server is not specified while binding a member to the service group, the globally configured name-server is considered.

Specifying name servers while binding a server to service groups using the CLI

At the command prompt, type:

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

Example:

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```

Specifying name servers while binding a server to service groups using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. In **Service Groups** page, select the service group that you have created and click **Edit**.
3. In **Load Balancing Service Groups** page, click **Service Group Members**.
4. In **Service Group Members Binding** page, select the server that you have created and click **Edit**.
5. In **Name Server**, specify the nameserver name to which the query for the bound domain must be sent.

Translate the IP address of a domain-based server

September 14, 2021

To simplify maintenance on the Citrix ADC appliance and on the domain-based servers that are connected to it, you can configure IP address masks and translation IP addresses. These functions work together to parse incoming DNS packets and substitute a new IP address for a DNS-resolved IP address.

When configured for a domain-based server, IP address translation enables the appliance to locate an alternate server IP address when you take the server down for maintenance or if you make any other infrastructure changes that affect the server.

When configuring the mask, you must use standard IP mask values (a power of two, minus one) and zeros, for example, 255.255.0.0. Non-zero values are only permitted in the starting octets.

When you configure a translation IP for a server, you create a 1:1 correspondence between a server IP address and an alternate server that shares leading or trailing octets in its IP address. The mask blocks particular octets in the original server's IP address. The DNS-resolved IP address is transformed to a new IP address by applying the translation IP address and the translation mask.

For example, you can configure a translation IP address of 10.20.0.0 and a translation mask of 255.255.0.0. If a DNS-resolved IP address for a server is 40.50.27.3, this address is transformed to 10.20.27.3. In this case, the translation IP address supplies the first two octets of the new address, and the mask passes through the last two octets from the original IP address. The reference to the original IP address, as resolved by DNS, is lost. Monitors for all services to which the server is bound also report on the transformed IP address.

When configuring a translation IP address for a domain-based server, you specify a mask and an IP address to which the DNS-resolved IP address is to be translated.

Note: Translation of the IP address is only possible for domain-based servers. You cannot use this feature for IP-based servers. The address pattern can be based on IPv4 addresses only.

To configure a translation IP address for a server by using the command line interface

At the command prompt, type:

```
1 add server <name>@ <serverDomainName> -translationIp <
     translationIPAddress> -translationMask <netMask> -state <ENABLED|
     DISABLED>
2 <!--NeedCopy-->
```

Example:

```
1 add server myMaskedServer www.example.com -translationIp 10.10.10.10 -
     translationMask
2 255.255.0.0 -state ENABLED
3 <!--NeedCopy-->
```

To configure a translation IP address for a server by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Servers**, create a domain-based server, and specify a translation IP address.

Mask a virtual server IP address

September 17, 2021

You can configure a mask and a pattern instead of a fixed IP address for a virtual server. This enables traffic that is directed to any of the IP addresses that match the mask and pattern to be rerouted to a particular virtual server. For example, you can configure a mask that allows the first three octets of an IP address to be variable, so that traffic to 111.11.11.198, 22.22.22.198, and 33.33.33.198 is all sent to the same virtual server.

By configuring a mask for a virtual server IP address, you can avoid reconfiguration of your virtual servers due to a change in routing or another infrastructure change. The mask allows the traffic to continue to flow without extensive reconfiguration of your virtual servers.

The mask for a virtual server IP address works differently from the IP pattern definition for a server described in Translating the IP Address of a Domain-Based Server. For a virtual server IP address mask, a non-zero mask is interpreted as an octet that is considered. For a service, the non-zero value is blocked.

Also, for a virtual server IP address mask, either leading or trailing values can be considered. If the virtual server IP address mask considers values from the left of the IP address, this is known as a forward mask. If the mask considers the values to the right side of the address, this is known as a reverse mask.

Note: The Citrix ADC appliance evaluates all forward mask virtual servers before evaluating reverse mask virtual servers.

When masking a virtual server IP address, you also need to create an IP address pattern for matching incoming traffic with the correct virtual server. When the appliance receives an incoming IP packet, it matches the destination IP address in the packet with the bits that are considered in the IP address pattern, and after it finds a match, it applies the IP address mask to construct the final destination IP address.

Consider the following example:

- Destination IP address in the incoming packet: 10.102.27.189
- IP address pattern: 10.102.0.0
- IP mask: 255.255.0.0
- Constructed (final) destination IP address: 10.102.27.189.

In this case, the first 16 bits in the original destination IP address match the IP address pattern for this virtual server, so this incoming packet is routed to this virtual server.

If a destination IP address matches the IP patterns for more than one virtual server, the longest match takes precedence. Consider the following example:

- Virtual Server 1: IP pattern 10.10.0.0, IP mask 255.255.0.0
- Virtual Server 2: IP pattern 10.10.10.0, IP mask 255.255.255.0
- Destination IP address in the packet: 10.10.10.45.
- Selected virtual server: Virtual Server 2.

The pattern associated with Virtual Server 2 matches more bits than that associated with Virtual Server 1, so IPs that match it is sent to Virtual Server 2.

Note: Ports are also considered if a tie-breaker is required.

To configure a virtual server IP address mask by using the command line interface

At the command prompt, type:

```
add lb vserver <name>@ http -ipPattern <ipAddressPattern> -ipMask <
   ipMask> <listenPort>
<!--NeedCopy-->
```

Example:

Pattern matching based on prefix octets:

```
add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask
   255.255.0.0 80
<!--NeedCopy-->
```

Pattern matching based on trailing octets:

```
add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask
   0.0.255.255 80
<!--NeedCopy-->
```

Modify a pattern-based virtual server:

```
set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255
<!--NeedCopy-->
```

If you configure the Virtual Server 1 as follows:

```
add lb vserver vs1 HTTP -ippattern 100.1.1.0 -ipmask 255.255.255.0 80
<!--NeedCopy-->
```

The Citrix ADC appliance will not respond to an ARP request on all the IP addresses. However, it responds to the virtual server traffic routed to all the IP addresses in that pattern.

To configure a virtual server IP address mask by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the Address Type list, select IP Pattern, and specify an IP pattern and IP mask.

Configure load balancing for commonly used protocols

September 14, 2021

In addition to websites and web-based applications, other types of network-deployed applications that use other common protocols often receive large amounts of traffic and therefore benefit from load balancing. Several of these protocols require specific configurations for load balancing to work properly. Among them are FTP, DNS, SIP, and RTSP.

If you configure your Citrix ADC appliance to use domain names for your servers rather than IPs, you might also need to set up IP translation and masking for those servers.

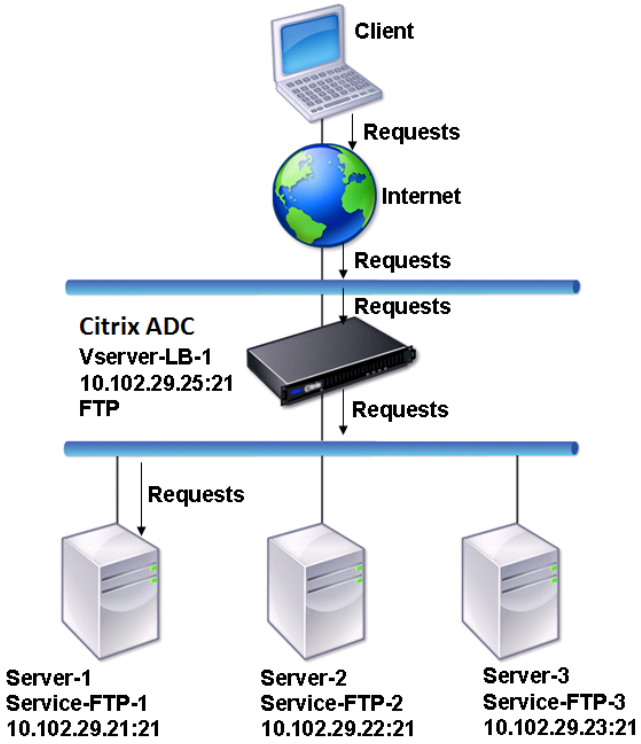
Load balance a group of FTP servers

September 14, 2021

The Citrix ADC appliance can be used to load balance FTP servers. FTP requires that the user initiate two connections on two different ports to the same server: the control connection, through which the client sends commands to the server, and the data connection, through which the server sends data to the client. When the client initiates an FTP session by opening a control connection to the FTP server, the appliance uses the configured load balancing method to select an FTP service, and forwards the control connection to it. The load balanced FTP server then opens a data connection to the client for information exchange.

The following diagram describes the topology of a load balancing configuration for a group of FTP servers.

Figure 1. Basic Load Balancing Topology for FTP Servers



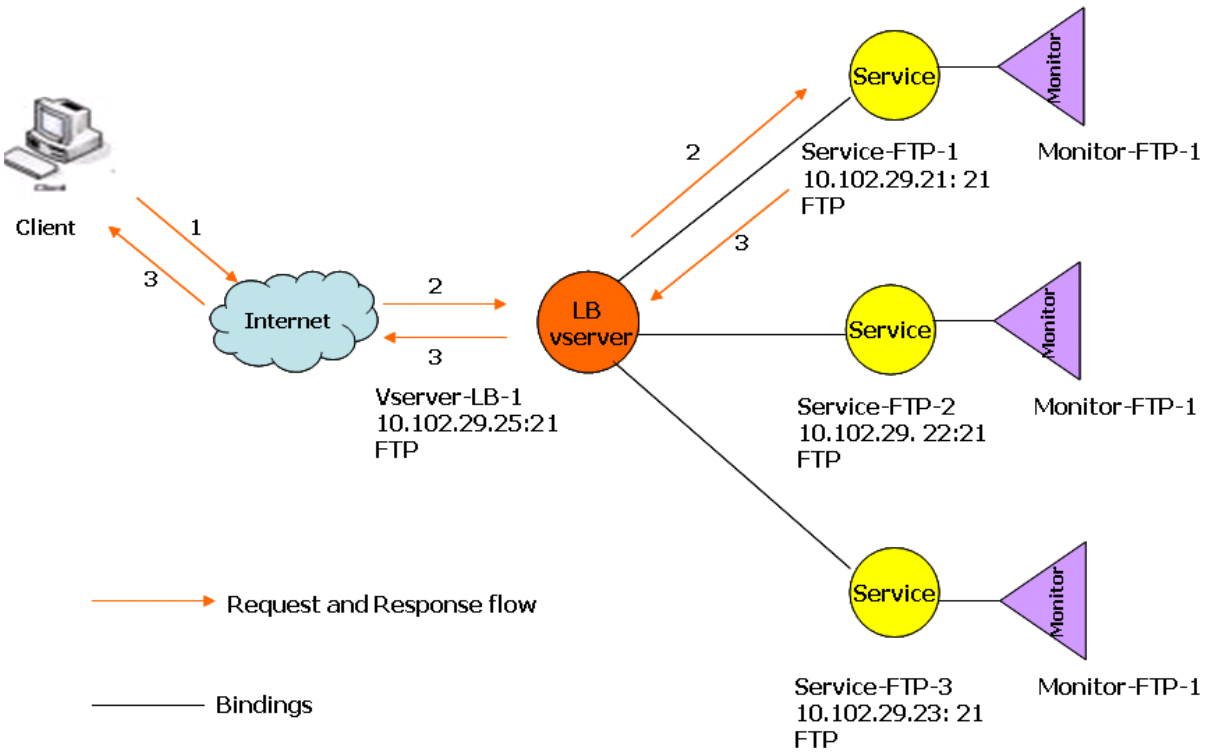
In the diagram, the services Service-FTP-1, Service-FTP-2, and Service-FTP-3 are bound to the virtual server Vserver-LB-1. Vserver-LB-1 forwards the client's connection request to one of the services using the least connection load balancing method. Subsequent requests are forwarded to the service that the appliance initially selected for load balancing.

The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
Services	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP
Monitors	FTP	None	None	None

The following diagram shows the load balancing entities, and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing FTP Servers Entity Model



The appliance can also provide a passive FTP option to access FTP servers from outside of a firewall. When a client uses the passive FTP option and initiates a control connection to the FTP server, the FTP server also initiates a control connection to the client. It then initiates a data connection to transfer a file through the firewall.

To create services and virtual servers of type FTP, see Setting Up Basic Load Balancing. Name the entities and set the parameters to the values described in the columns of the previous table. When you configure a basic load balancing setup, a default monitor is bound to the services.

Next, bind the FTP monitor to the services by following the procedure described in the section Binding Monitors to Services.

To create FTP monitors by using the CLI

At the command prompt, type:

```
add lb monitor <MonitorName> FTP -interval <Interval> -userName <
    UserName> -password <Password>
<!--NeedCopy-->
```

Example:


```
add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password
    User
<!--NeedCopy-->
```

To create FTP monitors by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Create a monitor of type FTP, and in Special Parameters, specify a user name and password.

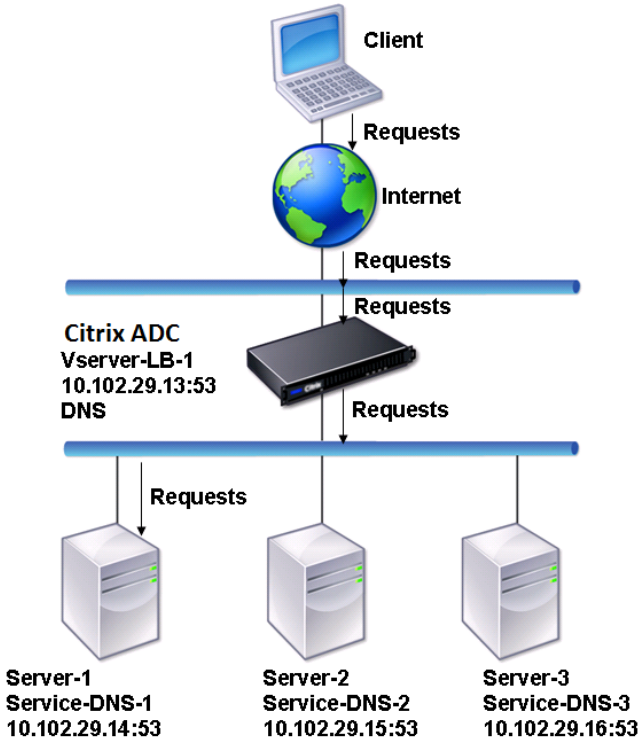
Load balance DNS servers

September 14, 2021

When you request DNS resolution of a domain name, the Citrix ADC appliance uses the configured load balancing method to select a DNS service. The DNS server to which the service is bound then resolves the domain name and returns the IP address as the response. The appliance can also cache DNS responses and use the cached information to respond to future requests for resolution of the same domain name. Load balancing DNS servers improve DNS response times.

The following diagram describes the topology of a load balancing configuration that load balances a group of DNS services.

Figure 1. Basic Load Balancing Topology for DNS Servers

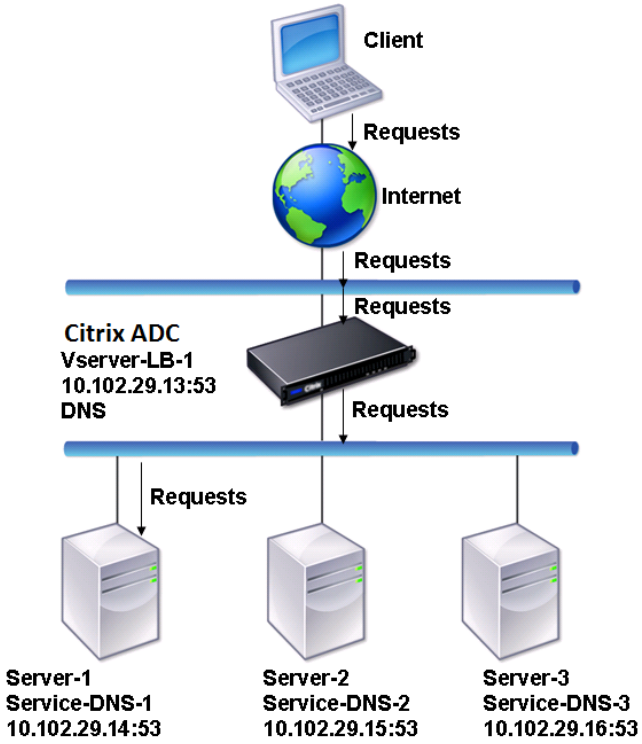


In the diagram, the services Service-DNS-1, Service-DNS-2, and Service-DNS-3 are bound to the virtual server Vserver-LB-1. The virtual server Vserver-LB-1 forwards client requests to a service using the least connection load balancing method. The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.13	53	DNS
Services	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
Monitors	monitor-DNS-1	None	None	None

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing DNS Servers Entity Model



To configure a basic DNS load balancing setup, see Setting Up Basic Load Balancing. Follow the procedures to create services and virtual servers of type DNS, naming the entities and setting the parameters using the values described in the previous table. When you configure a basic load balancing setup, the default ping monitor is bound to the services. For instructions on binding a DNS monitor to DNS services, you can also see Binding Monitors to Services.

The following procedure describes the steps to create a monitor that maps a domain name to the IP address based on a query.

To configure DNS monitors by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> DNS -query <domainName> -queryType <
    Address|ZONE> -IPAddress <ipAddress>
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType
    Address -IPAddress 10.102.29.66
```



```
2
3 add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType
     Address -IPAddress
4 1000:0000:0000:0000:0005:0600:700a::888b-888d
5 <!--NeedCopy-->
```

To configure DNS monitors by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Create a monitor of type DNS, and in Special Parameters, specify a query and query type.

Load balance domain-name based services

September 14, 2021

When you create a service for load balancing, you can provide an IP address. Alternatively, you can create a server using a domain name. The server name (domain name) can be resolved using an IPv4 or IPv6 name server, or by adding an authoritative DNS record (A record for IPv4 or AAAA record for IPv6) to the Citrix ADC configuration.

When you configure services with domain names instead of IP addresses, and if the name server resolves the domain name to a new IP address, the monitor bound to the service runs a health check on the new IP address, and updates the service IP address only when the IP address is found to be healthy. The monitor can be the default monitor bound to the service or you can bind any other supported monitor. It probes the service at regular intervals defined in the monitor parameters. If the domain name resolves to a new IP address, the monitor sends a fresh probe to check the health of the service. All subsequent probes are at the predefined interval.

Note: When you change the IP address of a server, the corresponding service is marked down for the first client request. The name server resolves the service IP address to the changed IP address for the next request, and the service is marked UP.

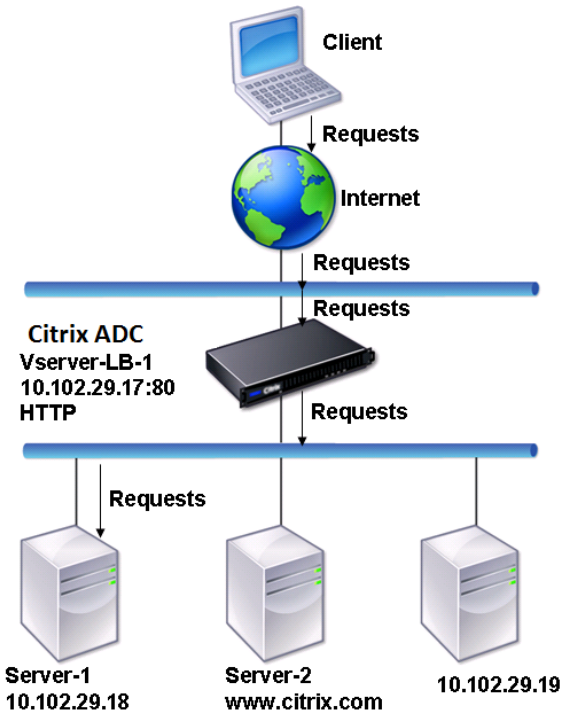
Domain-name based services have the following restrictions:

- The maximum domain name length is 255 characters.
- The Maximum Client parameter is used to configure a service that represents the domain name-based server. For example, a maxClient of 1000 is set for the services bound to a virtual server. When the connection count at the virtual server reaches 2000, the DNS resolver changes the IP address of the services. However, because the connection counter on the service is not reset, the virtual server cannot take any new connections until all the old connections are closed.
- When the IP address of the service changes, persistence is difficult to maintain.

- If the domain name resolution fails due to a timeout, the appliance uses the old information (IP address).
- When monitoring detects that a service is down, the appliance performs a DNS resolution on the service (representing the domain name-based server) to obtain a new IP address.
- Statistics are collected on a service and are not reset when the IP address changes.
- If a DNS resolution returns a code of “name error” (3), the appliance marks the service down and changes the IP address to zero.

When the appliance receives a request for a service, it selects the target service. This way, the appliance balances the load on your services. The following diagram describes the topology of a load balancing configuration that load balances a group of domain-name based servers (DBS).

Figure 1. Basic Load Balancing Topology for DBS Servers



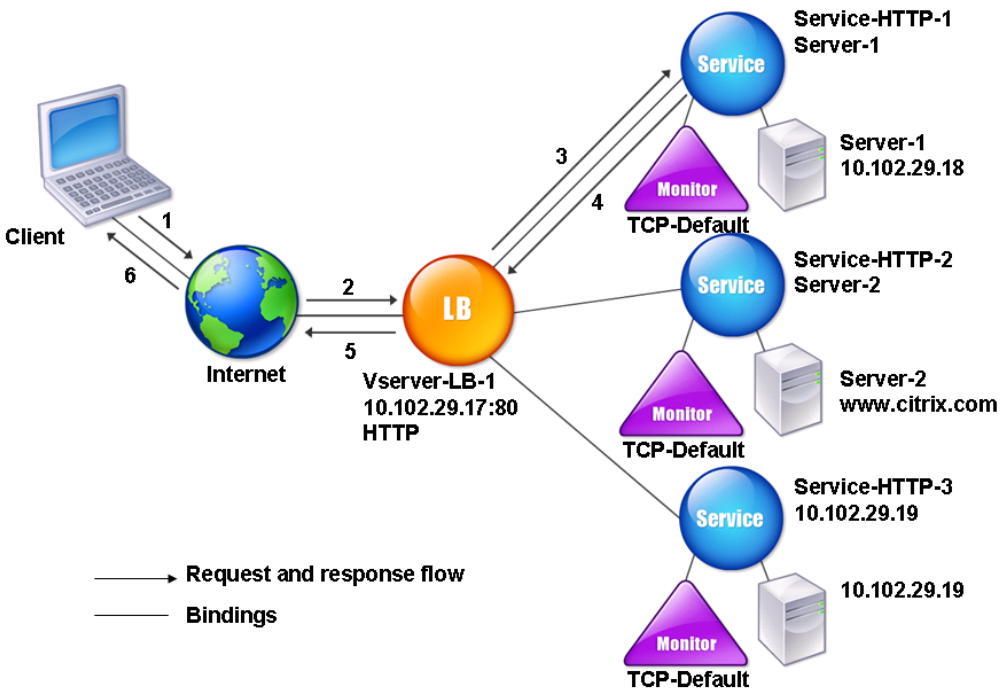
The services Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 are bound to the virtual server Vserver-LB-1. The virtual server Vserver-LB-1 uses the least connection load balancing method to choose the service. The IP address of the service is resolved using the name server Vserver-LB-2.

The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.17	80	HTTP
	Vserver-LB-2	10.102.29.20	53	DNS
Servers	server-1	10.102.29.18	80	HTTP
	server-2	www.citrix.com	80	HTTP
Services	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
Monitors	Default	None	None	None
Name Server	None	10.102.29.19	None	None

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing DBS Servers Entity Model



To configure a basic load balancing setup, see Setting Up Basic Load Balancing. Create the services and virtual servers of type HTTP, and name the entities and set the parameters using the values described in the previous table.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

To add a name server by using the command line interface

At the command prompt, type:

```
add dns nameServer <dnsVserverName>
<!--NeedCopy-->
```

Example:

```
add dns nameServer Vserver-LB-2
<!--NeedCopy-->
```

To add a name server by using the configuration utility

1. Navigate to **Traffic Management** > **DNS** > **Name Servers**.
2. Create a DNS name server of type DNS Virtual Server, and select a server from the DNS Virtual Server list.

You can also add an authoritative name server that resolves the domain name to an IP address.

> **Note**
>
> You can add a name server of type TCP, UDP, or UDP_TCP to resolver DBS probes. However, if TCP and UDP name servers coexists, and a UDP name server receives a response with the truncated bit, this response is not retried over the TCP name server.

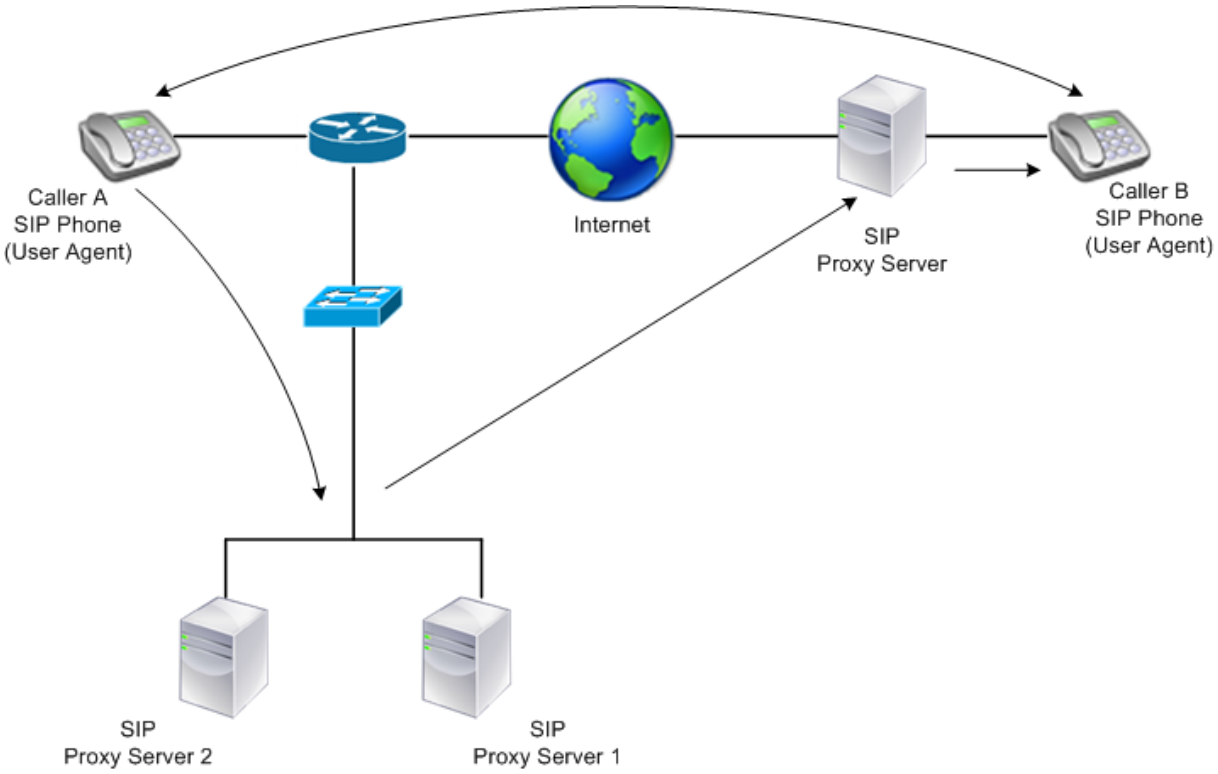
Load balance a group of SIP servers

September 14, 2021

The Session Initiation Protocol (SIP) is designed to initiate, manage, and terminate multimedia communications sessions. It has emerged as the standard for Internet telephony (VoIP). SIP messages can be transmitted over TCP or UDP. SIP messages are of two types: request messages and response messages.

The traffic in a SIP based communication system is routed through dedicated devices and applications (entities). In a multimedia communication session, these entities exchange messages. The following figure shows a basic SIP based communication system:

Figure 1. SIP Based Communication System



A Citrix ADC enables you to load balance SIP messages over UDP or over TCP (including TLS). You can configure the Citrix ADC to load balance SIP requests to a group of SIP proxy servers. To do so, you create a load balancing virtual server with the load balancing method and the type of persistence set to one of the following combinations:

- Call-ID hash load balancing method with no persistence setting
- Call-ID based persistence with least connection or round robin load balancing method
- Rule based persistence with least connection or round robin load balancing method

Also, by default, the Citrix ADC appends RPORT via the header of the SIP request, so that the server sends the response back to the source IP address and port from which the request originated.

Note: For load balancing to work, you must configure the SIP proxies so that they do not add private IP addresses or private domains to the SIP header/payload. SIP proxies must add to the SIP header a domain name that resolves to the IP address of the SIP virtual server. Also, the SIP proxies must communicate with a common database to share registration information.

Server Initiated Traffic

For SIP-server initiated outbound traffic, configure RNAT on the Citrix ADC so that the private IP addresses used by the clients are translated into public IP addresses.

If you have configured SIP parameters that include the RNAT source or destination port, the appliance compares the values of the source and destination ports of the request packets with the RNAT source port and RNAT destination port. If one of the values matches, the appliance updates the VIA header with RPORT. The SIP response from the client then traverses the same path as the request.

For server-initiated SSL traffic, the Citrix ADC uses a built-in certificate-key pair. If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the Citrix ADC internal service named **nsrnatsip-127.0.0.1-5061**.

Support for Policies and Expressions

The Citrix ADC default expressions language contains several expressions that operate on Session Initiation Protocol (SIP) connections. These expressions can be bound only to SIP based (sip_udp, sip_tcp or sip_ssl) virtual servers, and to global bind points. You can use these expressions in content switching, rate limiting, responder, and rewrite policies.

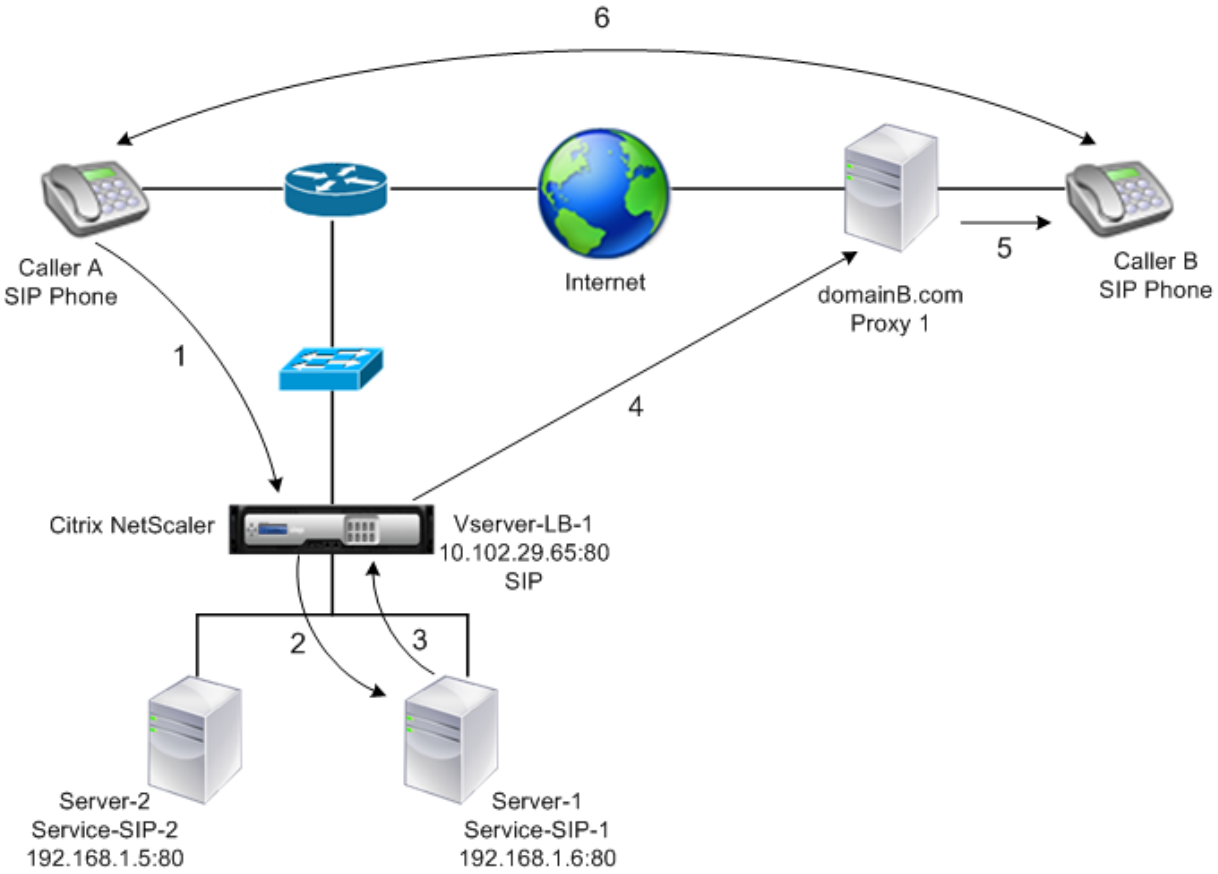
Configuring Load Balancing for SIP Signaling Traffic over TCP or UDP

The Citrix ADC can load balance SIP servers that send requests over UDP or TCP, including TCP traffic secured by TLS. The ADC provides the following service types to load balance the SIP servers:

- SIP_UDP – Used when SIP servers send SIP messages over UDP.
- SIP_TCP – Used when SIP servers send SIP messages over TCP.
- SIP_SSL – Used to secure SIP signaling traffic over TCP by using SSL or TLS. The Citrix ADC supports the following modes:
  - End-to-end TLS connection between the client, the ADC, and the SIP server.
  - TLS connection between the client and the ADC, and TCP connection between the ADC and the SIP server.
  - TCP connection between the client and the ADC, and TLS connection between the ADC and the SIP server.

The following figure shows the topology of a setup configured to load balance a group of SIP servers sending SIP messages over TCP or UDP.

Figure 2. SIP Load Balancing Topology



Entity type	Name	IP address	Port	Service type / Protocol
Virtual Server	Vserver-LB-1	10.102.29.65	80	SIP_UDP / SIP_TCP / SIP_SSL
Services	Service-SIP-1	192.168.1.6	80	SIP_UDP / SIP_TCP / SIP_SSL
	Service-SIP-2	192.168.1.5	80	SIP_UDP / SIP_TCP / SIP_SSL
Monitors	Default	None	80	SIP_UDP / SIP_TCP / SIP_SSL

Following is an overview of configuring basic load balancing for SIP traffic:

1. Configure services, and configure a virtual server for each type of SIP traffic that you want to load balance:

   - **SIP_UDP** – If you are load balancing the SIP traffic over UDP.
   - **SIP_TCP** – If you are load balancing the SIP traffic over TCP.
   - **SIP_SSL** – If you are load balancing and securing the SIP traffic over TCP.

   Note: If you use SIP_SSL, be sure to create an SSL certificate-key pair. For more information, see Adding a Certificate Key Pair.

2. Bind the services to the virtual servers.
3. If you want to monitor the states of the services with a monitor other than the default (**tcp-default**), create a custom monitor and bind it to the services. The Citrix ADC provides two custom monitor types, **SIP-UDP** and **SIP-TCP**, for monitoring SIP services.
4. If using a SIP_SSL virtual server, bind an SSL certificate-key pair to the virtual server.
5. If you are using the Citrix ADC as the gateway for the SIP servers in your deployment, configure RNAT.
6. If you want to append RPORT to the SIP messages that are initiated from the SIP server, configure the SIP parameters.

To configure a basic load balancing setup for SIP traffic by using the command line interface

Create one or more services. At the command prompt, type:

```
add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

Example:

```
add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80
<!--NeedCopy-->
```

Create as many virtual servers as necessary to handle the services that you created. The virtual server type must match the type of services that you bind to it. At the command prompt, type:

```
add lb vserver <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80
<!--NeedCopy-->
```


Bind each service to a virtual server. At the command prompt, type:

```
bind lb vserver <name> <serverName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-SIP-UDP-1
<!--NeedCopy-->
```

(Optional) Create a custom monitor of type SIP-UDP or SIP-TCP, and bind the monitor to the service. At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> [<interval>]

bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

Example:

```
add lb monitor mon1 sip-UDP -sipMethod REGISTER -sipuRI sip:mon@test.
   com -sipregURI sip:mon@test.com -respcode 200

bind monitor mon1 Service-SIP-UDP-1
<!--NeedCopy-->
```

If you created a SIP_SSL virtual server, bind an SSL certificate key pair to the virtual server. At the command prompt, type: At the command prompt, type:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -
   CA - skipCAName
<!--NeedCopy-->
```

Example:

```
bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1
<!--NeedCopy-->
```

Configure RNAT as required by your network topology. At the command prompt, type one of the following commands to create, respectively, an RNAT entry that uses a network address as the condition and SNIP as the NAT IP address, an RNAT entry that uses a network address as the condition and a unique IP address as the NAT IP address, an RNAT entry that uses an ACL as the condition and a SNIP as the NAT IP address, or an RNAT entry that uses an ACL as a condition and a unique IP address as the NAT IP address:


```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat
<!--NeedCopy-->
```

Example:

```
add rnat RNAT-1 192.168.1.0 255.255.255.0

bind rnat RNAT-1 -natip 10.102.29.50
<!--NeedCopy-->
```

If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the Citrix ADC internal service named nsrnatsip-127.0.0.1-5061.

```
add ssl certKey <certkeyName> -cert <string> [-key <string>]

bind ssl service <serviceName> -certkeyName <string>
<!--NeedCopy-->
```

Example:

```
add ssl certKey c1 -cert cert.epm -key key.ky

bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1
<!--NeedCopy-->
```

If you want to append RPORT to the SIP messages that the SIP server initiates, type the following command at the command prompt:

```
set lb sipParameters -rnatSrcPort <rnatSrcPort> -rnatDstPort<
    rnatDstPort> -retryDur <integer> -addRportVip <addRportVip> -
    sip503RateThreshold <sip503_rate_threshold_value>
<!--NeedCopy-->
```

Sample Configuration for load balancing the SIP traffic over UDP

```
add service service-UDP-1 10.102.29.5 SIP_UDP 80

Done

```



```
5  add lb vserver vserver-LB-1 SIP_UDP 10.102.29.60 80
6
7  Done
8
9  bind lb vserver vserver-LB-1 service-UDP-1
10
11 Done
12
13 add lb mon mon1 sip-udp -sipMethod REGISTER -sipuRI sip:mon@test.com -
       sipregURI sip:mon@test.com -respcode 200
14
15 Done
16
17 bind mon mon1 service-UDP-1
18
19 Done
20
21 add rnat RNAT-1 192.168.1.0 255.255.255.0
22
23 Done
24
25 set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
       -addRportVip ENABLED -sip503RateThreshold 1000
26
27 Done
28 <!--NeedCopy-->
```

Sample Configuration for load balancing the SIP traffic over TCP

```
1  add service service-TCP-1 10.102.29.5 SIP_TCP 80
2
3  Done
4
5  add lb vserver vserver-LB-1 SIP_TCP 10.102.29.60 80
6
7  Done
8
9  bind lb vserver vserver-LB-1 service-TCP-1
10
11 Done
12
13 add lb mon mon1 sip-tcp -sipMethod REGISTER -sipuRI sip:mon@test.com -
       sipregURI sip:mon@test.com -respcode 200
14
```



```
Done

bind mon mon1 service-TCP-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
    -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

Sample Configuration for load balancing and securing SIP traffic over TCP

```
add service service-SIP-SSL-1 10.102.29.5 SIP_SSL 80

Done

add lb vserver vserver-LB-1 SIP_SSL 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-SIP-SSL

Done

add lb mon mon1 sip-tCP -sipMethod REGISTER -sipuRI sip:mon@test.com -
   sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-SIP-SSL

Done

bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1

Done

add rnat RNAT-1 192.168.1.0 255.255.255.0
```



```
26
27 Done
28
29 set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
       -addRportVip ENABLED -sip503RateThreshold 1000
30
31 Done
32 <!--NeedCopy-->
```

To configure a basic load balancing setup for SIP traffic by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and add a virtual server of type SIP_UDP, SIP_TCP, or SIP_SSL.
2. Click the **Service** section, and add a service of type SIP_UDP, SIP_TCP, or SIP_SSL.
3. (Optional) Click the **Monitor** section, and add a monitor of the type: SIP-UDP or SIP-TCP.
4. Bind the monitor to the service, and bind the service to the virtual server.
5. If you created a SIP_SSL virtual server, bind an SSL certificate key pair to the virtual server. Click the Certificates section, and bind a certificate key pair to the virtual server.
6. Configure RNAT as required by your network topology. To configure RNAT:
   a) Navigate to **System > Network > Routes**.
   b) On the Routes page, click the **RNAT** tab.
   c) In the details pane, click **Configure RNAT**.
   d) In the Configure RNAT dialog box, do one of the following:
      - If you want to use the network address as a condition for creating an RNAT entry, click **Network** and set the following parameters:
        - Network
        - Netmask
      - If you want to use an extended ACL as a condition for creating an RNAT entry, click **ACL** and set the following parameters:
        - ACL Name
        - Redirect Port
   e) To set a SNIP address as a NAT IP address, skip to step 7.
   f) To set a unique IP address as a NAT IP, in the Available NAT IP (s) list, select the IP address that you want to set as the NAT IP, and then click Add. The NAT IP you selected appears in the Configured NAT IPs' list.
   g) Click Create, and then click Close.

   If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the Citrix ADC internal service named **nsrnatsip-127.0.0.1-5061**. To bind the pair:

a) Navigate to **Traffic Management > Load Balancing > Services** and click the Internal Services tab.
b) Select nsrnatsip-127.0.0.1-5061 and click **Edit**.
c) Click the **Certificates** section and bind a certificate key pair to the internal service.

7. If you want to append RPORT to the SIP messages that the SIP server initiates, configure the SIP parameters. Navigate to **Traffic Management > Load Balancing** and click Change SIP settings, set the various SIP parameters.

SIP Expression and Policy Example: Compression Enabled in Client Requests

A Citrix ADC cannot process compressed client SIP requests, so the client SIP request fails.

You can configure a responder policy that intercepts the SIP NEGOTIATE message from the client and looks for the compression header. If the message includes a compression header, the policy responds with "400 Bad Request," so that the client resends the request without compressing it.

At the command prompt, type the following commands to create the responder policy:

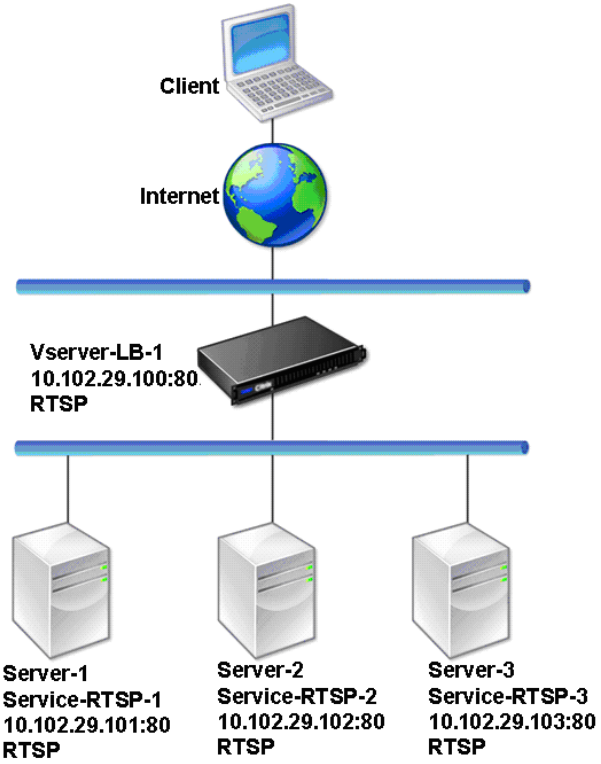
```
add responder action sipaction1 respondwith q{
 "SIP/2.0 400 Bad Requestrnrn" }


Done

add responder policy sippol1

add responder policy sippol1 "SIP.REQ.METHOD.EQ("NEGOTIATE")&&SIP.REQ.
   HEADER("Compression").EXISTS" sipaction1
<!--NeedCopy-->
```

Load balance RTSP servers

September 14, 2021

The Citrix ADC appliance can balance the load on RTSP servers to improve the performance of audio and video streams over networks. The following diagram describes the topology of a load balancing setup configured to load balance a group of RTSP servers.

Figure 1. Load Balancing Topology for RTSP

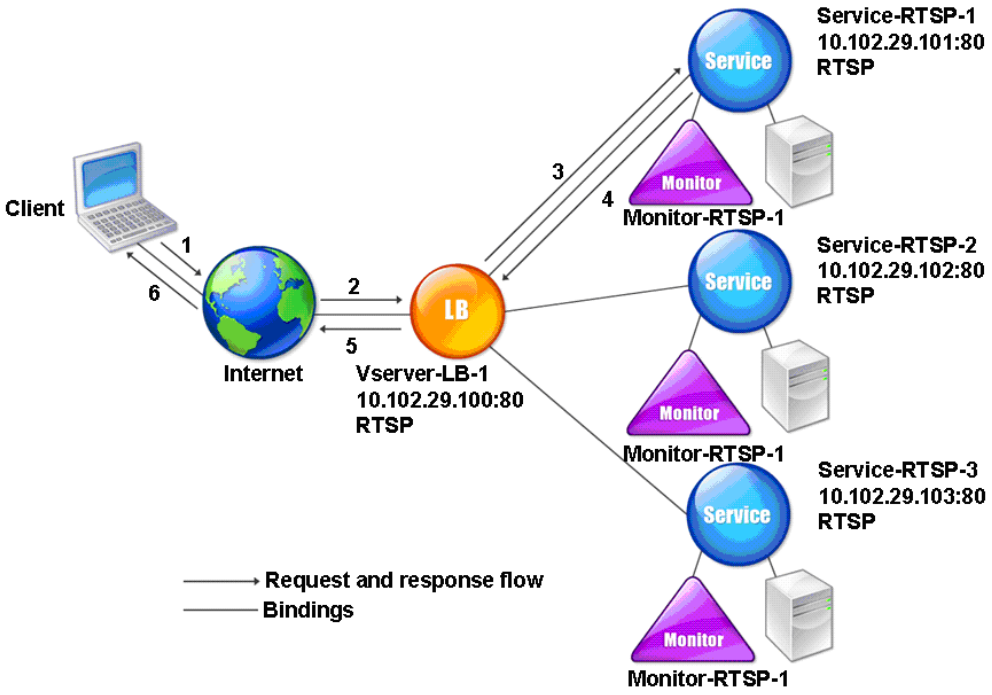


In the example, the services Service-RTSP-1, Service-RTSP-2, and Service-RTSP-3 are bound to the virtual server Vserver-LB-1. The following table lists the names and values of the example entities.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.100	554	RTSP
Services	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
Monitors	Monitor-RTSP-1	None	554	RTSP

The following diagram shows the load balancing entities used in the RTSP configuration.

Figure 2. Load Balancing RTSP Servers Entity Model



To configure a basic load balancing setup for RTSP servers, see Setting Up Basic Load Balancing. Create services and virtual servers of type RTSP. When you configure a basic load balancing setup, the default TCP-default monitor is bound to the services. To bind an RTSP monitor to these services, see Binding Monitors to Services. The following procedure describes how create a monitor that checks RTSP servers.

To configure RTSP monitors by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <type>
<!--NeedCopy-->
```

Example:

```
add lb monitor Monitor-RTSP-1 RTSP
<!--NeedCopy-->
```


To configure RTSP monitors by using the GUI

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type RTSP.

Load balance remote desktop protocol servers

September 14, 2021

Remote Desktop Protocol (RDP) is a multichannel-capable protocol that allows for separate virtual channels for carrying presentation data, serial device communication, licensing information, highly encrypted data (keyboard and mouse activity), and so on.

RDP is used for providing a GUI to another computer on the network. RDP is used with Windows terminal servers for providing fast access with almost real-time transmission of mouse movements and key presses even over low-bandwidth connections.

When multiple terminal servers are deployed to provide remote desktop services, the Citrix ADC appliance provides load balancing of the terminal servers (Windows 2003 and 2008 Server Enterprise Editions). Sometimes, a user who is accessing an application remotely may want to leave the application running on the remote machine but shut down the local machine. The user therefore closes the local application without logging out of the remote application. After reconnecting to the remote machine, the user must be able to continue with the remote application. To provide this functionality, the Citrix ADC RDP implementation honors the routing token (cookie) set by the Terminal Services Session Directory or Broker so that the client can reconnect to the same terminal server to which it was connected previously. The Session Directory, implemented on Windows 2003 Terminal Server, is referred to as Broker on Windows 2008 Terminal Server.

When a TCP connection is established between the client and the load balancing virtual server, the Citrix ADC applies the specified load balancing method and forwards the request to one of the terminal servers. The terminal server checks the session directory to determine whether the client has a session running on any other terminal server in the domain.

If there is no active session on any other terminal server, the terminal server responds by serving the client request, and the Citrix ADC appliance forwards the response to the client.

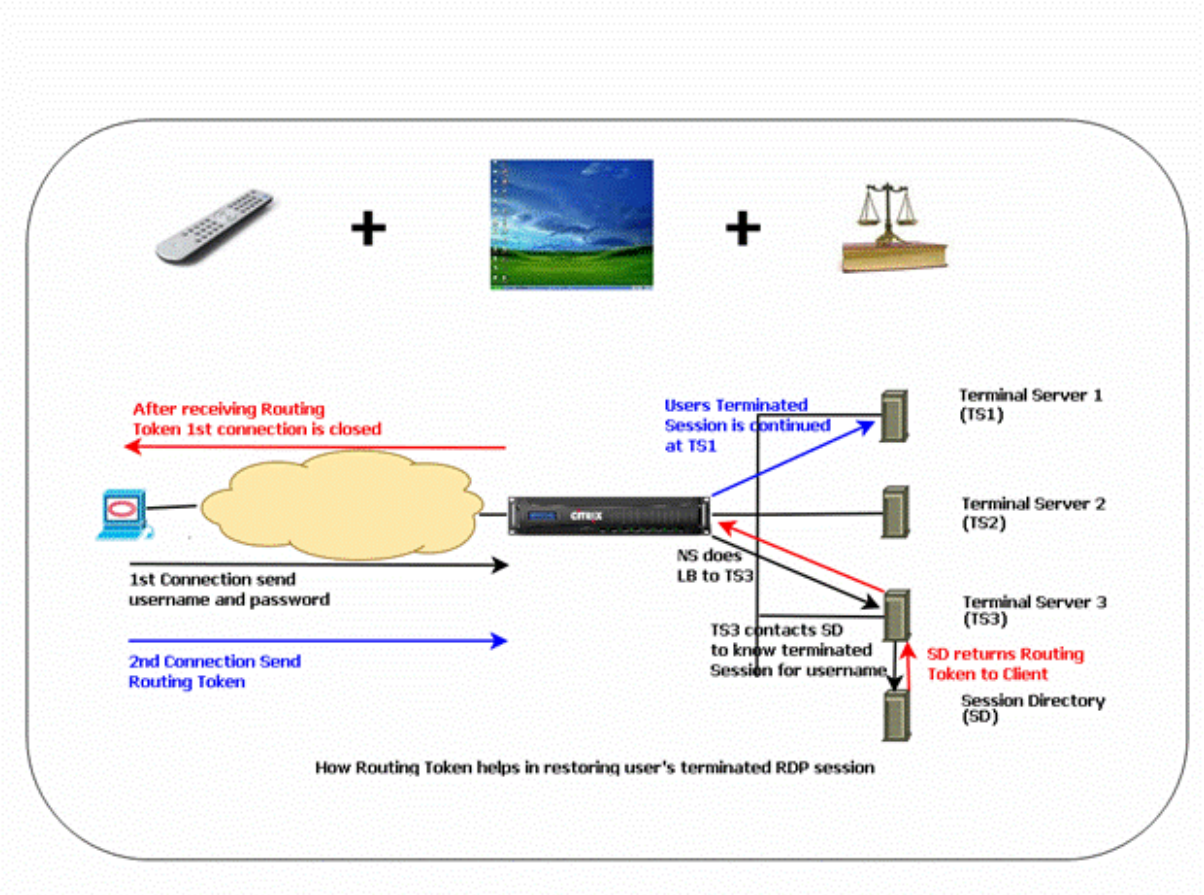
If there is an active session on any other terminal server, the terminal server that receives the request inserts a cookie (referred to as the routing token) with the details of the active session and returns the packets to the Citrix ADC appliance, which returns the packet to the client. The server closes the connection with the client. When the client retries to connect, the Citrix ADC reads the cookie information and forwards the packet to the terminal server on which the client has an active session.

The user on the client machine experiences a continuation of the service and does not have to take any specific action.

Note: The Windows Session Directory feature requires the Remote Desktop client that was first released with Windows XP. If a session with a Windows 2000 or Windows NT 4.0 Terminal Server client is disconnected and the client reconnects, the server with which the connection is established is selected by the load balancing algorithm.

The following diagram describes RDP load balancing.

Figure 1. Load Balancing Topology for RDP



Note

- When an RDP service is configured, persistence is automatically maintained by using a routing token. You need not enable persistence explicitly.
- The Citrix ADC appliance supports only IP-based cookies.
- The nsrdp.pl script is not supported on any current version of Windows servers.

Ensure that the disconnected RDP sessions are cleared on the terminal servers at the back end to avoid flapping between two terminal servers when an RDP session is disconnected without logging out. For more information, see `https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758177(v=ws.10)##BKMK_2`

When you add an RDP service, by default, Citrix ADC adds a monitor of the type TCP and binds it to the

service. The default monitor is a simple TCP monitor that checks whether a listening process exists at the 3389 port on the server specified for the RDP service. If there is a listening process at 3389, Citrix ADC marks this service as UP and if there is no listening process, it marks the service as DOWN.

For more efficient monitoring of an RDP service, in addition to the default monitor, you can configure a script monitor that is meant for the RDP protocol. When you configure the scripting monitor, the Citrix ADC opens a TCP connection to the specified server and sends an RDP packet. The monitor marks the service as UP only if it receives a confirmation of the connection from the physical server. Therefore, from the scripting monitor, the Citrix ADC can know whether the RDP service is ready to service a request.

The monitor is a user-type monitor and the script is located on the Citrix ADC at /nsconfig/monitors/nsrdp.pl. When you configure the user monitor, the Citrix ADC runs the script automatically. To configure the scripting monitor, add the monitor and bind it to the RDP service.

To configure RDP load balancing, create services of type RDP and bind them to an RDP virtual server.

To configure RDP load balancing services by using the command line interface

At the command prompt, type the following commands to configure an RDP load balancing setup and verify the configuration:

```
add service <name>@ <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Note: Repeat the preceding command to add more services.

Example

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
        State: UP
…
            Server Name: 10.102.27.182
            Server ID : 0           Monitor Threshold : 0
        Down state flush: ENABLED
…
1)      Monitor Name: tcp-default
                State: UP       Weight: 1
…
                Response Time: 4.152 millisec
```



```
 Done
<!--NeedCopy-->
```

To configure RDP load balancing services by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services**, and create services of type RDP.

To configure an RDP load balancing virtual server by using the command line interface

At the command prompt, type the following commands to configure an RDP load balancing virtual server and verify the configuration:

```
add lb vserver <name>@ <serviceType> <ipAddress> <port>

bind lb vserver <name>@ <serviceName>

Bind all the RDP services to be load balanced to the virtual server.
<!--NeedCopy-->
```

Example:

This example has two RDP services bound to the RDP virtual server.

```
add lb vs v1 rdP 10.102.27.186 3389
Done

bind lb vs v1 ser1
service "ser1" bound

bind lb vs v1 ser2
service "ser2" bound
Done

sh lb vs v1
v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
State: UP
…
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION
  Current Method: Round Robin, Reason: A new service is bound
Mode: IP
Persistence: NONE
  L2Conn: OFF

```



```
1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
<!--NeedCopy-->
```

To configure an RDP load balancing virtual server by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, create a virtual server of type RDP, and bind RDP services to this virtual server.

To configure a scripting monitor for RDP services by using the command line interface

At the command prompt, type the following commands:

```
add lb monitor <monitorName> USER -scriptName nsrdp.pl

bind lb monitor <monitorName> <rdpServiceName>
<!--NeedCopy-->
```

Example:

```
add service ser1 10.102.27.182 RDP 3389

add lb monitor RDP_MON USER -scriptName nsrdp.pl

bind lb monitor RDP_MON ser1

<!--NeedCopy-->
```

To configure a scripting monitor for RDP services by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors**, and create a monitor of type USER.
2. In Special Parameters, in the Script Name list, select nsrdp.pl, and then bind this monitor to an RDP service.

Load balance the Microsoft Exchange server

September 14, 2021

This document provides the recommended configuration examples for load balancing of the Microsoft Exchange server using the Citrix ADC appliance.

Citrix ADM StyleBooks simplifies Citrix ADC load balancing configurations for Exchange. For more information, see Microsoft Exchange StyleBook.

> Note:
>
> Load balancing of Microsoft Exchange is not possible using a single load balancing virtual server. Instead, follow the recommended configurations provided in this document.

Differences in Microsoft Exchange 2016 and newer versions

- You need not configure static Remote Procedure Call (RPC) ports on Exchange 2016 because RPC ports are not used.
- All sections named "for versions of Exchange below 2016" are not necessary with Exchange 2016.
- If you have configured any of the non-2016 versions already and you migrate to 2016, you do not have to remove them. Because even if they exist there are no issues.

Points to note

- For Remote Procedure Calls (RPC) with the Exchange server below 2016, the Exchange CAS servers must be configured for Static port assignments. For more information, see Exchange 2010 Client Access Server: Configure Static RPC Ports Microsoft documentation.
- This configuration assumes using the Citrix ADC appliance for SSL Offload. For more information, see How to Configure SSL Offloading in Exchange 2010 or Configuring SSL offloading in Exchange 2013.
- If you do not want to use the SSL Offload feature of the Citrix ADC appliance, change the service group `CAS_servicegroup_http` and monitors to type `SSL` and its bindings to port `443`.
- Surge Protection is not compatible with Microsoft Exchange. Do not enable it on any service or service group related to Microsoft Exchange. Enabling Surge Protection causes connectivity and reliability issues.
- Replace the following Variables with the proper information:
  - {HTTP Public IP}—IP Address for public Exchange HTTP endpoint
  - {RPC Public IP}—IP Address for public Exchange RPC endpoint (can be the same as HTTP Public IP)
  - {Timeout}—Desired timeout (in seconds). Recommended to be as long as standard work shift time (that is, 8 hours)

- {PersTimeout}—Desired timeout (in minutes). Must correspond to the preceding Timeout setting.
- {AB Port}—RPC Address Book TCP Port (usually 59601)
- {CA Port}—RPC Client Access TCP Port (usually 59600)
- {CertKey}—SSL Certificate Key
- {CAS-1 Server}—IP Address of CAS Server
- {CAS-2 Server}—IP Address of CAS Server

Recommended configuration examples for all versions of Microsoft Exchange server

Service Groups:

```
add serviceGroup CAS_servicegroup_http HTTP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP YES
add serviceGroup CAS_servicegroup_rpc_epm TCP -maxClient 0 -maxReq 0 -
   cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_http {
 CAS-1 Server }
  80 -CustomServerID "\"None\""
bind serviceGroup CAS_servicegroup_http {
 CAS-2 Server }
  80 -CustomServerID "\"None\""
bind serviceGroup CAS_servicegroup_rpc_epm {
 CAS-1 Server }
  135 -CustomServerID "\"None\""
bind serviceGroup CAS_servicegroup_rpc_epm {
 CAS-2 Server }
  135 -CustomServerID "\"None\""
<!--NeedCopy-->
```

Monitors:

```
add lb monitor CAS_monitor_rpc_epm TCP -LRTM ENABLED -destPort 135
set lb monitor http-ecv HTTP-ECV -recv 403 -LRTM DISABLED
bind serviceGroup CAS_servicegroup_http -monitorName http-ecv
```



```
bind serviceGroup CAS_servicegroup_rpc_epm -monitorName
   CAS_monitor_rpc_epm
<!--NeedCopy-->
```

Load balancing virtual servers:

```
add lb vserver CAS_vserver_owa SSL 0.0.0.0 0 -persistenceType
   COOKIEINSERT -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_as SSL 0.0.0.0 0 -persistenceType RULE -
   timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -rule "HTTP.REQ.HEADER(\"Authorization\")"
     -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_oa SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_ews SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add lb vserver CAS_vserver_ad SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

set ssl vserver CAS_vserver_owa -sslRedirect ENABLED
bind ssl vserver CAS_vserver_owa -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_oab -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_as -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_oa -certkeyName {
```



```
 CertKey }

bind ssl vserver CAS_vserver_ews -certkeyName {
 CertKey }

bind ssl vserver CAS_vserver_ad -certkeyName {
 CertKey }

bind lb vserver CAS_vserver_owa CAS_servicegroup_http
bind lb vserver CAS_vserver_oab CAS_servicegroup_http
bind lb vserver CAS_vserver_as CAS_servicegroup_http
bind lb vserver CAS_vserver_oa CAS_servicegroup_http
bind lb vserver CAS_vserver_ews CAS_servicegroup_http
bind lb vserver CAS_vserver_ad CAS_servicegroup_http
add lb vserver CAS_vserver_rpc_epm TCP {
 RPC Public IP }
  135 -timeout {
 PersTimeout }
  -cltTimeout {
 Timeout }
  -comment "vserver for RPC End Point Mapper"
bind lb vserver CAS_vserver_rpc_epm CAS_servicegroup_rpc_epm
<!--NeedCopy-->
```

Persistency group:

```
add lb group CAS_persistency_group_sourceip
bind lb group CAS_persistency_group_sourceip CAS_vserver_oa
bind lb group CAS_persistency_group_sourceip CAS_vserver_oab
bind lb group CAS_persistency_group_sourceip CAS_vserver_ews
bind lb group CAS_persistency_group_sourceip CAS_vserver_ad
bind lb group CAS_persistency_group_sourceip CAS_vserver_rpc_epm
set lb group CAS_persistency_group_sourceip -persistenceType SOURCEIP -
    timeout {
 PersTimeout }

<!--NeedCopy-->
```

Content Switching for HTTP services:

```
add cs vserver CAS_vserver_cs SSL {
 Public IP }
  443 -cltTimeout {
 Timeout }
  -caseSensitive OFF -comment "Exchange CS VServer"
```



```
bind ssl vserver CAS_vserver_cs -certkeyName {
 CertKey }

add cs action CAS_action_cs_owa -targetLBVserver CAS_vserver_owa
add cs action CAS_action_cs_oab -targetLBVserver CAS_vserver_oab
add cs action CAS_action_cs_as -targetLBVserver CAS_vserver_as
add cs action CAS_action_cs_oa -targetLBVserver CAS_vserver_oa
add cs action CAS_action_cs_ews -targetLBVserver CAS_vserver_ews
add cs action CAS_action_cs_autodiscover -targetLBVserver
   CAS_vserver_ad
add cs policy CAS_policy_cs_owa -rule "HTTP.REQ.URL.SET_TEXT_MODE(
   IGNORECASE).STARTSWITH(\"/owa\")" -action CAS_action_cs_owa
add cs policy CAS_vserver_oab -rule "HTTP.REQ.URL.SET_TEXT_MODE (
   IGNORECASE).STARTSWITH(\"/OAB\")"
add cs policy CAS_policy_cs_as -rule "HTTP.REQ.URL.SET_TEXT_MODE(
   IGNORECASE).STARTSWITH(\"/Microsoft-Server-ActiveSync\")" -action
   CAS_action_cs_as
add cs policy CAS_policy_cs_autodiscover -rule "HTTP.REQ.URL.
   SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/Autodiscover\")" -action
   CAS_action_cs_autodiscover
add cs policy CAS_policy_cs_oa -rule "HTTP.REQ.URL.SET_TEXT_MODE(
   IGNORECASE).STARTSWITH(\"/rpc\")" -action CAS_action_cs_oa
add cs policy CAS_policy_cs_ews -rule "HTTP.REQ.URL.SET_TEXT_MODE(
   IGNORECASE).STARTSWITH(\"/EWS\")" -action CAS_action_cs_ews

bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_oa -priority
   90
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_owa -priority
   100
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_oab -priority
   100
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_as -priority
   110
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_autodiscover -
   priority 120
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_ews -priority
   130
bind cs vserver CAS_vserver_cs -lbvserver CAS_vserver_owa
<!--NeedCopy-->
```

Recommended configuration examples for versions of Microsoft Exchange server below 2016

Additional service groups:


```
add serviceGroup CAS_servicegroup_rpc_ca TCP -maxClient 0 -maxReq 0 -
    cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
add serviceGroup CAS_servicegroup_rpc_ab TCP -maxClient 0 -maxReq 0 -
    cip DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_rpc_ca {
 CAS-1 Server }
  {
 CA Port }
  -CustomServerID "\"None\""
bind serviceGroup CAS_servicegroup_rpc_ca {
 CAS-2 Server }
  {
 CA Port }
  -CustomServerID "\"None\""
bind serviceGroup CAS_servicegroup_rpc_ab {
 CAS-1 Server }
  {
 AB Port }
  -CustomServerID "\"None\""
bind serviceGroup CAS_servicegroup_rpc_ab {
 CAS-2 Server }
  {
 AB Port }
  -CustomServerID "\"None\""
<!--NeedCopy-->
```

Additional monitors:

```
add lb monitor CAS_monitor_rpc_ca TCP -LRTM ENABLED -destPort {
 CA Port }

add lb monitor CAS_monitor_rpc_ab TCP -LRTM ENABLED -destPort {
 AB Port }

bind serviceGroup CAS_servicegroup_rpc_ca -monitorName
    CAS_monitor_rpc_ca
```



```
bind serviceGroup CAS_servicegroup_rpc_ab -monitorName
    CAS_monitor_rpc_ab
<!--NeedCopy-->
```

Additional load balancing virtual servers:

```
add lb vserver CAS_vserver_rpc_ab TCP {
 RPC Public IP }
  {
 AB Port }
  -timeout {
 PersTimeout }
  -cltTimeout {
 Timeout }
  -comment "vserver for  RPC Address Book"
add lb vserver CAS_vserver_rpc_ca TCP {
 RPC Public IP }
  {
 CA Port }
  -timeout {
 PersTimeout }
  -cltTimeout {
 Timeout }
  -comment "vserver for  RPC Client Access"
bind lb vserver CAS_vserver_rpc_ab CAS_servicegroup_rpc_ab
bind lb vserver CAS_vserver_rpc_ca CAS_servicegroup_rpc_ca
<!--NeedCopy-->
```

Additional persistency group:

```
bind lb group CAS_persistency_group_sourceip CAS_vserver_rpc_ab
bind lb group CAS_persistency_group_sourceip CAS_vserver_rpc_ca
<!--NeedCopy-->
```

Recommended configuration examples for versions of Microsoft Exchange server 2016 and newer

Additional load balancing virtual server:

```
add lb vserver CAS_vserver_mapi SSL 0.0.0.0 0 -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

```



```
bind ssl vserver CAS_vserver_mapi -certkeyName {
 CertKey }

bind lb vserver CAS_vserver_mapi CAS_servicegroup_http
<!--NeedCopy-->
```

Additional persistency group:

```
bind lb group CAS_persistency_group_sourceip CAS_vserver_mapi
<!--NeedCopy-->
```

Content switching for HTTP services:

```
add cs action CAS_action_cs_mapi -targetLBVserver CAS_vserver_mapi
add cs policy CAS_policy_cs_mapi -rule "HTTP.REQ.URL.SET_TEXT_MODE(
    IGNORECASE).STARTSWITH(\"/mapi\")" -action CAS_action_cs_mapi
bind cs vserver CAS_vserver_cs -policyName CAS_policy_cs_mapi -priority
     140
<!--NeedCopy-->
```

Optional configurations

HTTPS redirect for Outlook Web App (OWA):

```
add lb vserver CAS_vserver_owa_http_redirect HTTP {
 HTTP Public IP }
  80 -persistenceType COOKIEINSERT -timeout {
 PersTimeout }
  -lbMethod ROUNDROBIN -redirectURL "https://mail.example.com/owa" -
     cltTimeout {
 Timeout }

<!--NeedCopy-->
```

NOTE: Replace with proper HTTPS Redirect URL.

Policy for /owa rewrite:

```
add rewrite action owa_rewrite replace http.REQ.URL "\"/owa\""
add rewrite policy owa_rewrite_policy "http.req.url.eq(\"/\")"
    owa_rewrite
bind lb vserver CAS_vserver_owa -policyName owa_rewrite_policy -
    priority 100 -gotoPriorityExpression END -type REQUEST
add responder action action_responder_owa redirect "\"https://www.
    example.com/owa\""
```



```
add responder policy policy_responder_owa HTTP.REQ.IS_VALID
    action_responder_owa
set responder param -undefAction NOOP
bind lb vserver CAS_vserver_owa -policyName policy_responder_owa -
    priority 100 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

NOTE: Replace with proper HTTPS Redirect URL.

Support for SMTP:

For the following configuration, USIP must be enabled so that the CAS servers can see the sending SMTP server's IP address for validation. This configuration also requires that the default gateway of the CAS server is configured to point at the ADC appliance's SNIP address.

```
add lb vserver CAS_vserver_smtp TCP {
 HTTP Public IP }
  25 -persistenceType SOURCEIP -timeout 60 -lbMethod LEASTCONNECTION -
     cltTimeout 30
add serviceGroup CAS_servicegroup_smtp TCP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip YES -SP OFF -useproxyport YES -cltTimeout 30 -
    svrTimeout 30 -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_smtp {
 CAS-1 Server }
  25 -CustomServerID "\"None\"" bind serviceGroup CAS_servicegroup_smtp
      {
 CAS-2 Server }
  25 -CustomServerID "\"None\""
bind lb vserver CAS_vserver_smtp CAS_servicegroup_smtp
<!--NeedCopy-->
```

Support for Post Office Protocol version 3 (POP3):

```
add lb vserver CAS_vserver_pop3 TCP {
 HTTP Public IP }
  110 -persistenceType SOURCEIP -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add serviceGroup CAS_servicegroup_pop3 TCP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
```



```
bind serviceGroup CAS_servicegroup_pop3 {
 CAS-1 Server }
  110 -CustomServerID "\"None\"" bind serviceGroup
     CAS_servicegroup_pop3 {
 CAS-2 Server }
  110 -CustomServerID "\"None\""
bind lb vserver CAS_vserver_pop3 CAS_servicegroup_pop3
<!--NeedCopy-->
```

> Note:
>
> You can perform the preceding configuration for SSL-encrypted POP3 by changing the port to 995 and the virtual server/service types to SSL. Also bind a suitable SSL certificate.

Support for IMAP:

```
add lb vserver CAS_vserver_imap TCP {
 HTTP Public IP }
  143 -persistenceType SOURCEIP -timeout {
 PersTimeout }
  -lbMethod LEASTCONNECTION -cltTimeout {
 Timeout }

add serviceGroup CAS_servicegroup_imap TCP -maxClient 0 -maxReq 0 -cip
    DISABLED -usip NO -SP OFF -useproxyport YES -cltTimeout {
 Timeout }
  -svrTimeout {
 Timeout }
  -CKA NO -TCPB NO -CMP NO
bind serviceGroup CAS_servicegroup_imap {
 CAS-1 Server }
  143 -CustomServerID "\"None\"" bind serviceGroup
     CAS_servicegroup_imap {
 CAS-2 Server }
  143 -CustomServerID "\"None\""
bind lb vserver CAS_vserver_imap CAS_servicegroup_imap
<!--NeedCopy-->
```

> Note:
>
> You can perform the preceding configuration for SSL-encrypted IMAP by changing the port to 993 and the virtual server/service types to SSL. Also bind a suitable SSL certificate.

Other Resources

- Configuring Load Balancing Servers for Microsoft Exchange with Email Security Filtering
- Deploying NetScaler with Microsoft Exchange 2016

Use case 1: SMPP load balancing

September 14, 2021

Millions of short messages are exchanged daily between individuals and value-added service providers, such as banks, advertisers, and directory services, by using the short message peer to peer (SMPP) protocol. Often, message delivery is delayed because servers are overloaded and traffic is not optimally distributed among the servers. The Citrix ADC supports SMPP load balancing and provides optimal distribution of messages across your servers, preventing poor performance and outages.

The Citrix ADC performs load balancing on the server side when messages are received from clients and on the client side when messages are received from servers.

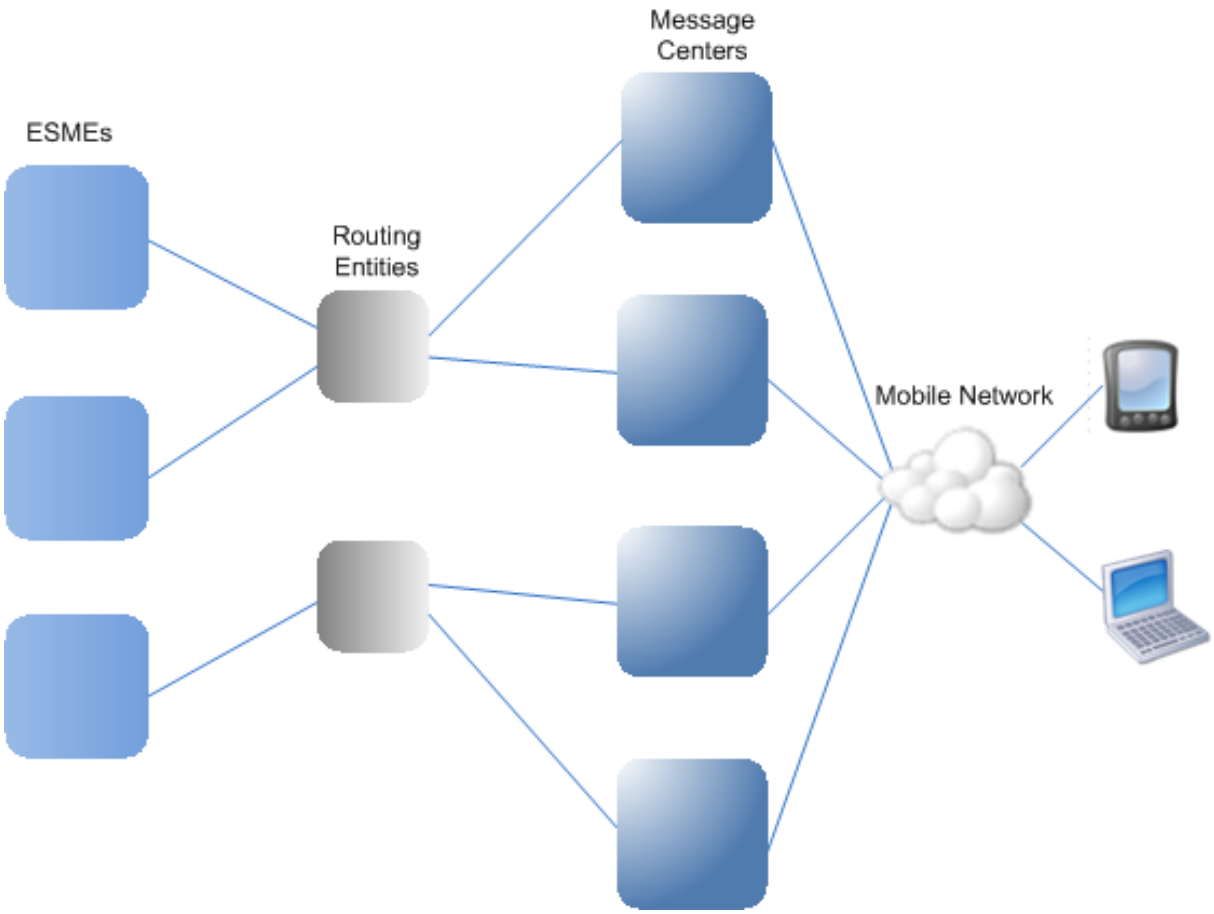
Load balancing of SMPP messages by the Citrix ADC provides the following benefits:

- Better load distribution on servers, which translates to faster response time to end users
- Server health monitoring and better failover capabilities
- Quick and easy addition of new servers (message centers) without changing the client configuration
- High availability

Introduction to SMPP

SMPP is an application layer protocol for the transfer of short messages between External Short Message Entities (ESME), Routing Entities (RE) and Message Centers (MC) over long-lived TCP connections. It is used for sending short message service (SMS) messages between friends, contacts, and third parties such as banks (mobile banking), advertisers (mobile commerce), and directory services. Messages from an ESME (non-mobile entity) arrive at the MC, which distributes them to short message entities (SMEs) such as mobile phones. SMPP is also used by SMEs to send short messages to third parties (for example, for purchase of products, bill payment, and funds transfer). These messages arrive at the MC and are forwarded to the destination MC or ESME.

The following diagram shows the different SMPP entities: ESMEs, REs, and MCs, in a mobile network.



Architecture Overview of the Different SMPP Entities in a Mobile Network

Note: The terms client and ESME are used interchangeably throughout the document.

An ESME (client) opens a connection to the MC in one of the three modes: as a transmitter, a receiver, or a transceiver. As a transmitter, it can only submit messages for delivery. As a receiver, it can only receive messages. As a transceiver, the ESME can both submit and receives messages. The ESME sends the MC one of the three messages (also known as PDUs): bind_transmitter, bind_receiver, or bind_transceiver. The MC responds with a bind_transmitter_resp, bind_receiver_resp, or bind_transceiver_resp, as appropriate for the request.

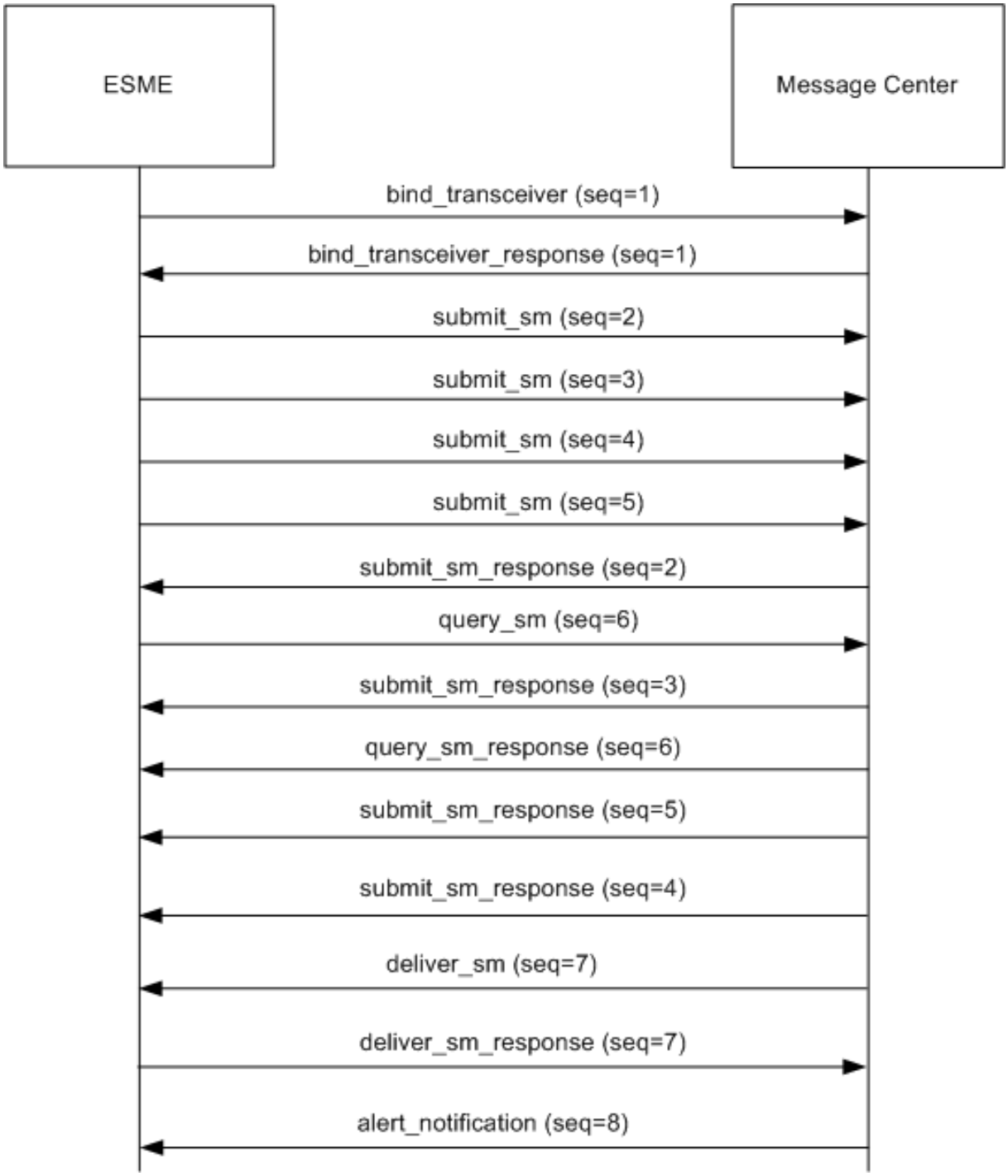
After the connection is established, the ESME can, depending on the mode in which it is bound to the MC, send a submit_sm or data_sm message, receive a deliver_sm or data_sm message, or send and receive any of these types of messages. The ESME can also send ancillary messages, such as query_sm, replace_sm, and cancel_sm, to query the status of an earlier message delivery, replace an earlier message with a new message, or cancel an undelivered message.

If a message is not delivered because an ESME is not available or a mobile subscriber is not online, the message is queued. Later, when the MC detects that the mobile subscriber is now reachable, it sends

an alert_notification PDU to the ESME over a receiver or transceiver session, requesting delivery of any queued messages.

Each request PDU has a unique sequence number. The response PDU has the same sequence number as the original request. Because message exchange over SMPP can be in asynchronous mode, an ESME or an MC can send multiple requests at a time. The sequence number plays a crucial role in returning the response in the same SMPP session. In other words, the sequence number makes request and response matching possible.

The following diagram shows how the traffic flow uses the various PDUs when the ESME binds as a transceiver.



Limitation:

The Citrix ADC appliance does not support out bound operations. That is, a message center cannot initiate an SMPP session with an ESME through the Citrix ADC appliance.

How SMPP Load Balancing Works on the Citrix ADC

An ESME (client) sends a bind message to open a connection to the Citrix ADC. The ADC authenticates each ESME and, if successful, responds with an appropriate message. The Citrix ADC establishes a connection with each message center and load balances all the messages among these message centers. When the ADC receives a message from a client, it reuses an open connection to the message center or sends a bind request to a message center if an open connection is not available.

The ADC can load balance messages originating from the clients and from the servers. It can monitor the health of the message centers and handle concatenated messages. It also provides content switching support for the message centers.

Messages Originating from the ESMEs

Each ESME must be added as a user on the Citrix ADC for authentication. The client establishes a TCP connection with an SMPP virtual server configured on the ADC by sending a bind request. The ADC authenticates the client and, if successful, parses the bind message. The ADC then sends the request to the message center selected by the configured load balancing method. If a connection to the message center is not available for reuse, the ADC opens a TCP connection with the message center by sending a new bind request to the message center.

Before forwarding the response (submit_sm_resp or data_sm_resp) from the message center to the client, the ADC adds a custom server ID to the message ID to identify the message center for ancillary operations, such as query, replace, or cancel requests for a message, by the client. Requests from other clients are load balanced in the same way.

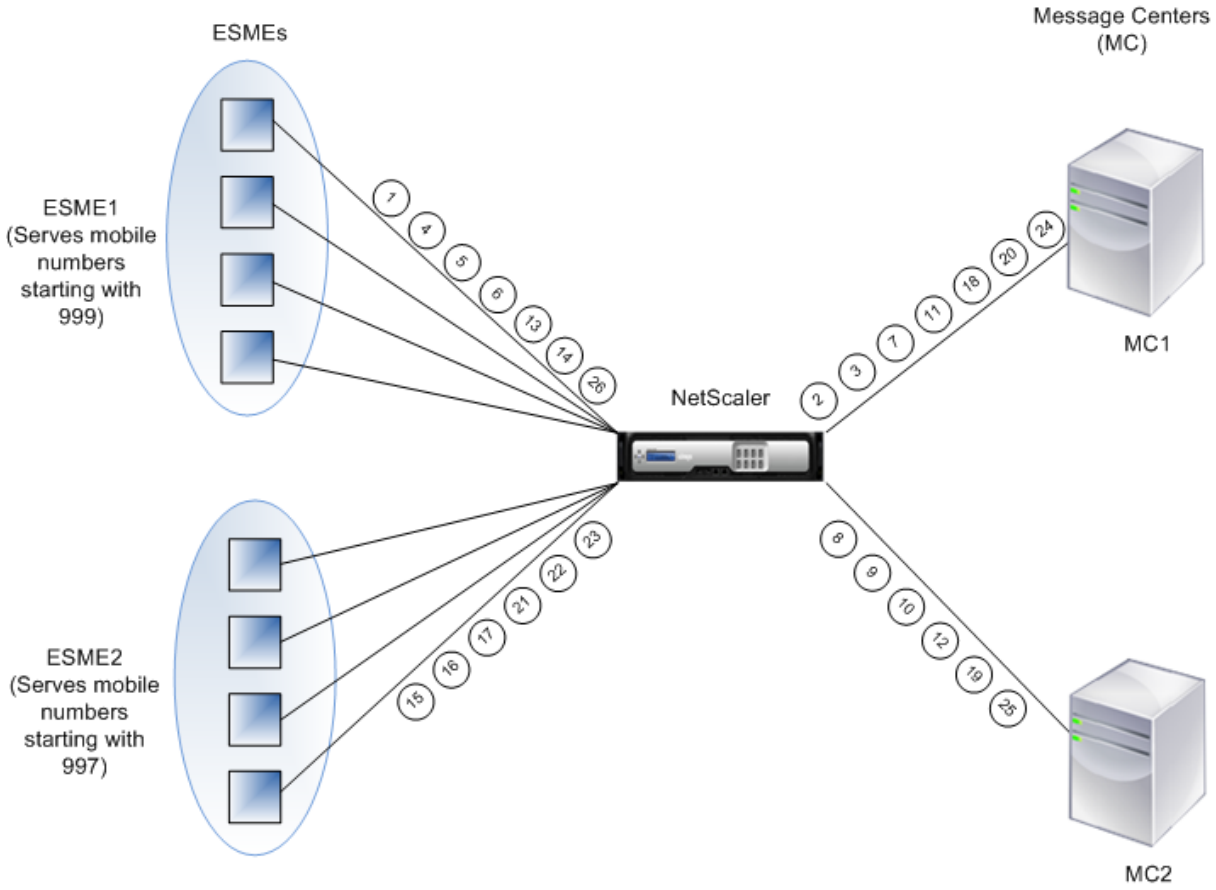
In the original bind request, a client specifies the address range that it can serve. This range is used for forwarding deliver_sm or data_sm messages from the message centers to the clients.

Messages Originating from a Message Center

ESMEs that can handle a specific address range are grouped into a cluster. All the nodes in a cluster provide the same credentials. Within a cluster, only the round robin method is used for load balancing. To deliver mobile originated (MO) messages, the message center sends a deliver_sm message to the Citrix ADC. If a cluster that can serve the destination address range (for example, numbers starting with 998) is bound to the ADC, it selects that cluster, and then load balances the message among the ESME nodes in that cluster.

If an ESME that can serve deliver_sm messages for the address range is not bound to the ADC, and message queuing is enabled, the message is queued until such a client binds to the ADC in a receiver or transceiver mode. You can specify the size of the queue.

The following diagram illustrates the internal flow of PDUs between ESMEs, Citrix ADC, and the message centers. For simplicity, only two ESMEs and two message centers are shown.



Flow of messages (PDUs):

1. ESME1 sends bind request to NetScaler
2. NetScaler sends bind request to MC1
3. MC1 sends bind response to NetScaler
4. NetScaler sends bind response to ESME1
5. ESME1 sends submit_sm(1) to NetScaler
6. ESME1 sends submit_sm(2) to NetScaler
7. NetScaler forwards submit_sm(1) to MC1
8. NetScaler sends bind request to MC2
9. MC2 sends bind response to NetScaler
10. NetScaler forwards submit_sm(2) to MC2
11. MC1 sends submit_sm_resp(1) to NetScaler
12. MC2 sends submit_sm_resp(2) to NetScaler
13. NetScaler forwards submit_sm_resp(1) to ESME1
14. NetScaler forwards submit_sm_resp(2) to ESME1
15. ESME2 sends bind request to NetScaler
16. NetScaler sends bind response to ESME2
17. ESME2 sends submit_sm(3) to NetScaler

18. NetScaler forwards submit_sm(3) to MC1
19. MC2 sends deliver_sm to NetScaler (ESME2 serves the address range specified in the message)
20. MC1 sends submit_sm_resp(3) to NetScaler
21. NetScaler forwards submit_sm_resp(3) to ESME2
22. NetScaler forwards deliver_sm to ESME2
23. ESME2 sends deliver_sm_resp to NetScaler
24. MC1 sends alert_notification to NetScaler (ESME1 serves the address range specified in the message)
25. NetScaler forwards deliver_sm_resp to MC2
26. NetScaler forwards the alert_notification to ESME1

Health Monitoring of Message Centers

By default, a TCP_default monitor is bound to an SMPP service, but you can bind a custom monitor of type SMPP. The custom monitor opens a TCP connection to the message center and sends an enquire_link packet. Depending on the success or failure of the probe, the service is marked UP or DOWN.

Content Switching on Message Centers

Message centers can accept multiple connections (or bind requests) from ESMEs. You can configure the Citrix ADC to content switch these requests based on the SMPP bind parameters. Following are some common expressions for configuring methods to select a message center:

- Based on the address range: In the following sample expression, the ADC selects a specific message center if the address range starts at 988.

Example:

SMPP.BINDINFO.ADDRESS_RANGE.CONTAINS("^988")

- Based the ESME ID: In the following sample expression, the ADC selects a specific message center if the ESME ID equals ESME1.

Example:

SMPP.BINDINFO.SYSTEM_ID.EQ("ESME1")

- Based on the ESME type: In the following sample expression, the ADC selects a specific message center if the ESME type is VMS. VMS stands for voice mail system.

Example:

SMPP.BINDINFO.SYSTEM_TYPE.EQ("VMS")

- Based on the type of number (TON) of the ESME: In the following sample expression, the ADC selects a specific message center if TON equals 1 (1 stands for an international number.)

Example:

SMPP.BINDINFO.ADDR_TON.EQ(1)

- Based on the number plan indicator (NPI) of the ESME: In the following sample expression, the ADC selects a specific message center if NPI equals 0 (0 stands for an unknown connection.)

Example:

SMPP.BINDINFO.ADDR_NPI.EQ(0)

- Based on the bind type: In the following sample expression, the ADC selects a specific message center if the bind type is TRANSCEIVER. (A transceiver can send and receive messages.)

Example:

SMPP.BINDINFO.TYPE.EQ(TRANSCEIVER)

Concatenated Message Handling

An SMS can hold a maximum of 140 bytes. Longer messages must be broken down into smaller parts. If the destination mobile is capable, the messages are combined and delivered as one long SMS. The Citrix ADC forwards the fragments of a message to the same message center. Each message contains a reference number, a sequence number, and the total number of fragments. The reference number is the same for each fragment of a long message. The sequence number specifies that position of the particular fragment in the complete message. After all the fragments are received, the ESME combines the fragments into one long message and delivers the message to the mobile subscriber.

If a client disconnects from an active connection, the connection to the message center is not closed. It is reused for requests from other clients.

Limitation

Message IDs, from the message center, longer than 59 bytes are not supported. If the message ID length returned by the message center is more than 59 bytes, ancillary operations fail and the Citrix ADC responds with an error message.

Configuring SMPP Load Balancing on the Citrix ADC

Perform the following tasks to configure SMPP load balancing on the ADC:

1. Add an SMPP user. The ADC authenticates the user before it accepts a bind request from the user. The user is typically an ESME.
2. Add a load balancing virtual server, specifying the protocol as SMPP.
3. Add a service, specifying the protocol as SMPP, and a custom server ID that is unique for each server. Bind the service to the load balancing virtual server created earlier.

4. Optionally, create a service group and add services to the service group.
5. Optionally, add a monitor of type SMPP-ECV and bind it to the service. A TCP-default monitor is bound by default.
6. Set the SMPP parameters, such as client mode and message queue.

To configure SMPP load balancing by using the command line

At the command prompt, type:

```
add smpp user <username> -password <password>
add service <name> <IP> SMPP <port>  - customserverID <customserverID>
add lb vserver <name> <IP> SMPP <port>
bind lb vserver <name> <service name>
set smpp param
<!--NeedCopy-->
```

Example

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID ab -CKA NO -TCPB NO -CMP NO
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID xy -CKA NO -TCPB NO -CMP NO
add lb vserver smppvs SMPP 10.102.239.179 2775 -persistenceType NONE -
    cltTimeout 180
bind lb vserver smppvs smmpsvc2
bind lb vserver smppvs smmpsvc
set smpp param -addrrange "d*"
<!--NeedCopy-->
```

To configure SMPP load balancing by using the configuration utility

1. Navigate to **System > User Administration > SMPP Users**, and add an SMPP user.
2. Navigate to **Traffic Management > Load Balancing > Configure SMPP Parameters**, and set the parameters as required by your deployment.
3. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and add a virtual server of type SMPP.

4. Click in the Service section, add a service of type SMPP, and specify a Server ID.

Use case 2: Configure rule based persistence based on a name-value pair in a TCP byte stream

September 14, 2021

Some protocols transmit name-value pairs in a TCP byte stream. The protocol in the TCP byte stream in this example is the Financial Information eXchange (FIX) protocol. In non-XML implementation, the FIX protocol enables two hosts communicating over a network to exchange business or trade-related information as a list of name-value pairs (called "FIX fields"). The field format is <tag>=<value><delimiter>. This traditional tag-value format makes the FIX protocol ideal for the use case.

The tag in a FIX field is a numeric identifier that indicates the meaning of the field. In the example;

- The tag 35 indicates the message type.
- The value after the equal sign holds a specific meaning for the given tag and is associated with a data type. A value of A for the tag 35 indicates that the message is a logon message.
- The delimiter is the nonprinting "Start of Header" (SOH) ASCII character (0x01), which is the caret symbol (^).
- Each field is also assigned a name. The field with tag 35 is the msgType field.

Following is an example of a logon message.

8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52=20000426-12:05:06 98=0 108=30 10=157

Your choice of persistence type for a tag-value list such as the one shown above is determined by the options that are available to you for extracting a particular string from the list. Token-based persistence methods require you to specify the offset and length of the token that you want to extract from the payload. The FIX protocol does not allow you to do that, because the offset of a given field and the length of its value can vary from one message to another. This variation depends on the message type, the preceding fields, and the lengths of the preceding values. It also varies based on the implementation from one to another, depending on whether custom fields have been defined. Such variations make it impossible to predict the exact offset of a given field or to specify the length of the value that is to be extracted as the token. In this case, therefore, rule based persistence is the preferred persistence type.

Assume that a virtual server fixlb1 is load balancing TCP connections to a farm of servers hosting instances of a FIX-enabled application. You want to configure persistence for connections on the basis of the value of the SenderCompID field, which identifies the firm sending the message. The tag for this FIX field is 49 (shown in the earlier logon message example).

To configure rule based persistence for the load balancing virtual server, set the persistence type for the load balancing virtual server to RULE and configure the rule parameter with an expression. The expression must be one that extracts the portion of the TCP payload in which you expect to find the SenderCompID field, typecasts the resulting string to a name-value list based on the delimiters, and then extracts the value of the SenderCompID field (tag 49), as follows:

```
set lb vserver fixlb1 -persistenceType RULE -rule "CLIENT.TCP.PAYLOAD(300).
TYPECAST_NVLIST_T('=','^').VALUE("\49\")"
```

Note: Backslash characters have been used in the expression because this is a CLI command. If you are using the configuration utility, do not enter the backslash characters.

If the client sends a FIX message that contains the name-value list in the earlier logon message example, the expression extracts the value INVMGR, and the Citrix ADC appliance creates a persistence session based on this value.

The argument to the PAYLOAD() function can be as large as you deem is necessary to include the SenderCompID field in the string extracted by the function. Optionally, you can use the SET_TEXT_MODE(IGNORECASE) function if you want the appliance to ignore the case when extracting the value of the field, and the HASH function to create a persistence session based on a hash of the extracted value. The following expression uses the SET_TEXT_MODE(IGNORECASE) and HASH functions:

```
CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=','^').SET_TEXT_MODE(IGNORECASE
).VALUE("49").HASH
```

Following are more examples of rules that you can use to configure persistence for FIX connections (replace <tag> with the tag of the field whose value you want to extract):

- To extract the value of any FIX field in the first 300 bytes of the TCP payload, you can use the expression CLIENT.TCP.PAYLOAD(300).BEFORE_STR("^").AFTER_STR("<tag>=").
- To extract a string that is 20 bytes long at offset 80, cast the string to a name-value list, and then extract the value of the field that you want, use the expression CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPE
  tag>").
- To extract the first 100 bytes of the TCP payload, cast the string to a name-value list, and extract the value of the third occurrence of the field that you want, use the expression CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T('=','^').VALUE("<tag>",2).
  Note: If the second argument that is passed to the
  VALUE() function is
  n, the appliance extracts the value of the
  (n+1)
  th instance of the field because the count starts from zero (
  0).

Following are more examples of rules that you can use to configure persistence. Only the payload-

based expressions can evaluate data being transmitted through the FIX protocol. The other expressions are more general expressions for configuring persistence based on lower networking protocols.

- CLIENT.TCP.PAYLOAD(100)
- CLIENT.TCP.PAYLOAD(100).HASH
- CLIENT.TCP.PAYLOAD(100).SUBSTR(5,10)
- CLIENT.TCP.SRCPORT
- CLIENT.TCP.DSTPORT
- CLIENT.IP.SRC
- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

Use case 3: Configure load balancing in direct server return mode

September 14, 2021

Load balancing in direct server return (DSR) mode allows the server to respond to clients directly by using a return path that does not flow through the Citrix ADC appliance. In DSR mode, however, the appliance can continue to perform health checks on services. In a high-data volume environment, sending server traffic directly to the client in DSR mode increases the overall packet handling capacity of the appliance because the packets do not flow through the appliance.

DSR mode has the following features and limitations:

- It supports one-arm mode and inline mode.
- The appliance ages out sessions based on idle timeout.
- Because the appliance does not proxy TCP connections (that is it does not send SYN-ACK to the client), it does not shut out SYN attacks. By using the SYN packet rate filter, you can control the rate of SYNs to the server. To control the rate of SYNs, set a threshold for the rate of SYNs. To get protection from SYN attacks, you must configure the appliance to proxy TCP connections. However, that requires the reverse traffic to flow through the appliance.
- In a DSR configuration, the Citrix ADC appliance does not replace the load balancing virtual server's IP address with the destination server's IP address. Instead, it forwards packets to a service by using the server's MAC address. The VIP must be configured on the server and ARP must be disabled for the VIP which is configured on the server. Doing so prevents the client request from bypassing the appliance when it is configured in one-arm mode. For example, a user must configure VIP in the loopback interface and disable the ARP for the same VIP.

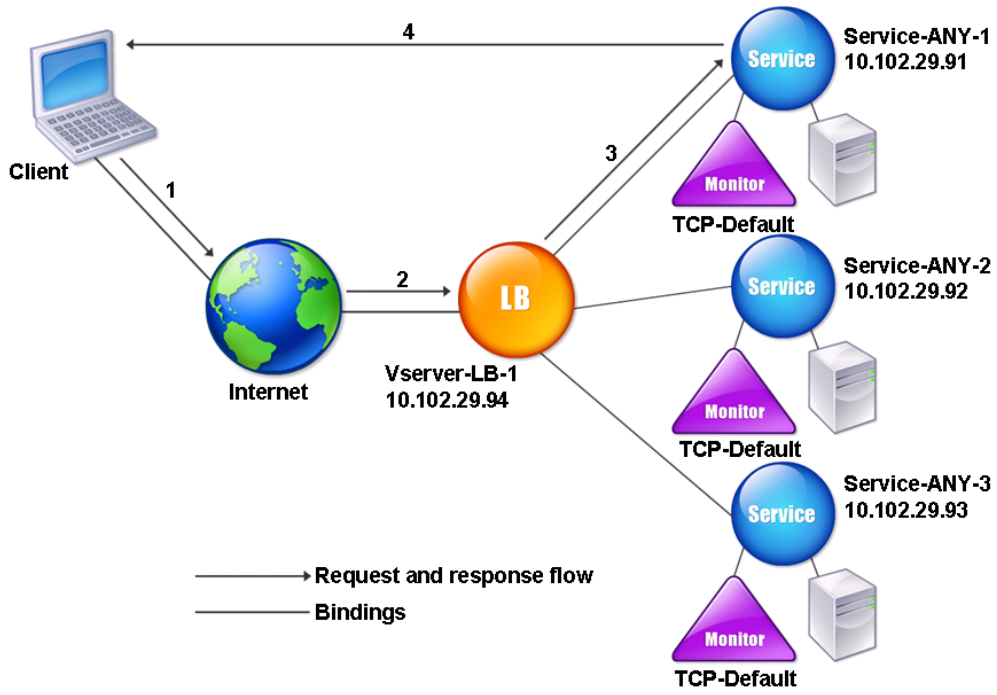
- The appliance obtains the server's MAC address from the monitor bound to the service. However, custom user monitors (monitors of type USER), which use scripts stored on the Citrix ADC appliance, do not learn a server's MAC address. If you use only custom monitors in a DSR configuration, for each request the virtual server receives, the appliance attempts to resolve the destination IP address to a MAC address (by sending ARP requests). Because the destination IP address is a virtual IP address owned by the Citrix ADC appliance, the ARP requests always resolve to the MAC address of the Citrix ADC interface. Therefore, all traffic received by the virtual server is looped back to the appliance. If you use user monitors in a DSR configuration, you must also configure another monitor of a different type (for example, a PING monitor) for the services, ideally with a longer interval between probes, so that the MAC address of the servers can be learned.
- The Citrix ADC appliance learns the server L2 parameters from the monitor that is bound to the service. For UDP-ECV monitors, configure a receive string to enable the appliance to learn the L2 parameters of the server. If the receive string is not configured and the server does not respond, then the appliance does not learn the L2 parameters but the service is set to UP. The traffic for this service is blackholed.

In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service, and the service responds to clients directly, bypassing the Citrix ADC appliance. The following table lists the names and values of the entities configured on the Citrix ADC appliance in DSR mode.

Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
Monitors	TCP	None	None

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

Figure 1. Entity Model for Load Balancing in DSR Model



For the appliance to function correctly in DSR mode, the destination IP in the client request must be unchanged. Instead, the appliance changes the destination MAC to that of the selected server. This setting enables the server to determine the client MAC address for forwarding requests to the client while bypassing the server.

Next, you configure a basic load balancing setup as described in Setting Up Basic Load Balancing, naming the entities and setting the parameters using the values described in the previous table.

After you configure the basic load balancing setup, you must customize it for DSR mode. To do this, you configure a supported load balancing method, such as the Source IP Hash method with a sessionless virtual server. You also need to set the redirection mode to allow the server to determine the client MAC address for forwarding responses and bypass the appliance.

After you configure the load balancing method and redirection mode, you need to enable the USIP mode on each service. The service then uses the source IP address when forwarding responses.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the command line interface

At the command prompt, type:


```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
   RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

Example

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless
   enabled
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, select Redirection Mode as MAC Based, and method as SOURCEIPHASH.
3. In Traffic Settings, select Sessionless Load Balancing.

To configure a service to use source IP address by using the command line interface

At the command prompt, type:

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

To configure a service to use source IP address by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Open a service, and in Traffic Settings, select **Use Source IP Address**.

Some additional steps are required in certain situations, which are described in the succeeding sections.

Use case 4: Configure LINUX servers in DSR mode

September 14, 2021

The LINUX operating system requires that you set up a loopback interface with the Citrix ADC appliance virtual IP address (VIP) on each load balanced server in the DSR cluster.

To configure LINUX server in DSR mode

To create a loop back interface with the Citrix ADC appliance's VIP on each load balanced server, at the Linux OS prompt type the following commands:

```
ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
<!--NeedCopy-->
```

Then, run the software that remaps the TOS id to VIP.

Note: Add the correct mappings to the software before running it. In the preceding commands, the LINUX server uses dummy0 to connect to the network. When you use this command, type the name of the interface that your LINUX server uses to connect to the network.

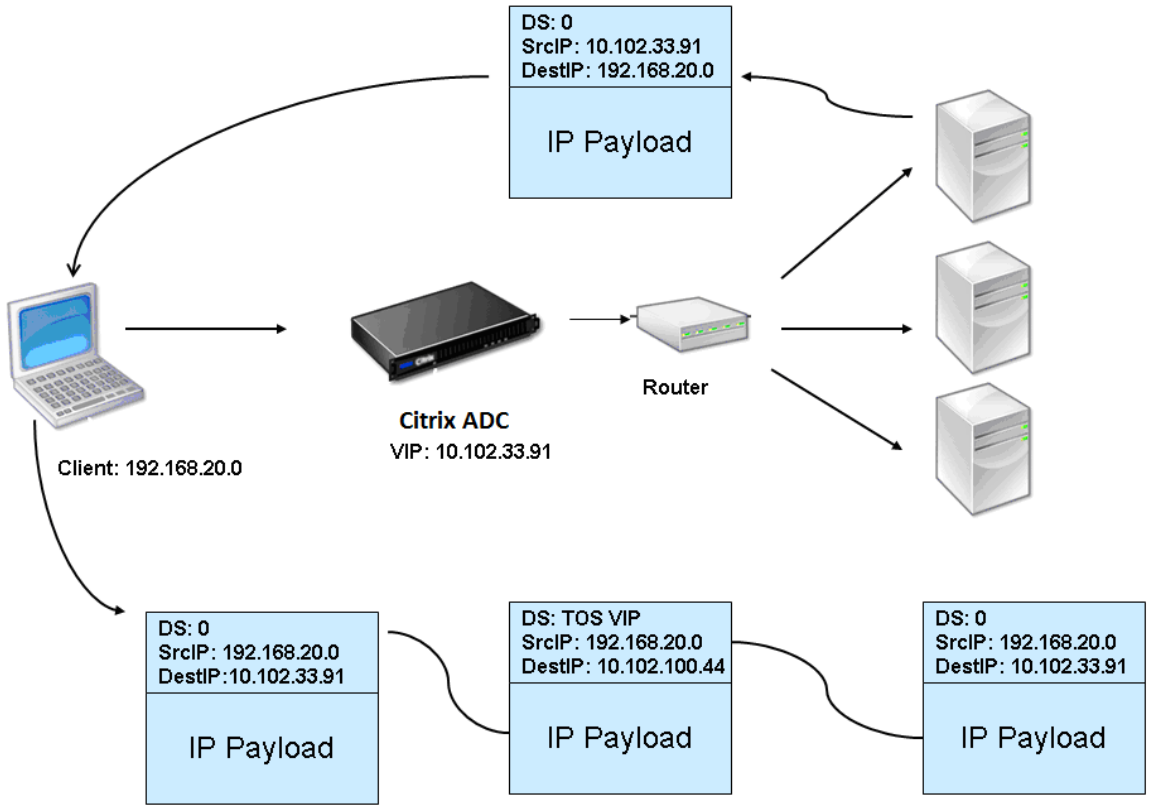
Use case 5: Configure DSR mode when using TOS

September 14, 2021

Differentiated services (DS), also known as TOS (Type of Service), is a field that is part of the IPv4 packet header. The equivalent field in the IPv6 header is Traffic Class. TOS is used by upper layer protocols for optimizing the path for a packet. The TOS information encodes the Citrix ADC appliance virtual IP address (VIP), and the load balanced servers extract the VIP from it.

In the following scenario, the appliance adds the VIP to the **TOS** field in the packet and then forwards the packet to the load balanced server. The load balanced server then responds directly to the client, bypassing the appliance, as illustrated in the following diagram.

Figure 1. The Citrix ADC appliance in DSR mode with TOS



The TOS feature is customized for a controlled environment as follows:

- The environment must not have any stateful devices, such as stateful firewall and TCP gateways, in the path between the appliance and the load balanced servers.
- Routers at all the entry points to the network must remove the TOS field from all incoming packets to make sure that the load balanced server does not confuse another TOS field with that added by the appliance.
- Each server can have only 63 VIPs.
- The intermediate router must not send out ICMP error messages regarding fragmentation. The client does not understand the message, as the source IP address is the IP address of the load balanced server and not the Citrix ADC VIP.
- TOS is valid only for IP-based services. You cannot use domain name based services with TOS.

In the example, Service-ANY-1 is created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to the service, and the service responds to clients directly, bypassing the appliance. The following table lists the names and values of the entities configured on the appliance in DSR mode.

Entity Type	Name	IP Address	Protocol
Virtual server	Vserver-LB-1	10.102.33.91	ANY
Services	Service-ANY-1	10.102.100.44	ANY
Monitors	PING	None	None

DSR with TOS requires that load balancing is set up on layer 3. To configure a basic load balancing setup for Layer 3, see Setting Up Basic Load Balancing. Name the entities and set the parameters using the values described in the previous table.

After you configure the load balancing setup, you must customize the load balancing setup for DSR mode by configuring the redirection mode to allow the server to decapsulate the data packet and then respond directly to the client and bypass the appliance.

After specifying the redirection mode, you can optionally enable the appliance to transparently monitor the server. This enables the appliance to transparently monitor the load balanced servers.

To configure the redirection mode for the virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -m <Value> -tosId <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
<!--NeedCopy-->
```

To configure the redirection mode for the virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and in Redirect Mode, select TOS ID.

To configure the transparent monitor for TOS by using the command line interface

At the command prompt, type:


```
add monitor <MonitorName> <Type> -destip <DestinationIP> -tos <Value> -
    tosId <Value>
<!--NeedCopy-->
```

Example:

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes  -tosId 3
<!--NeedCopy-->
```

To create the transparent monitor for TOS by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Create a monitor, select TOS, and type the TOS ID that you specified for the virtual server.

Wildcard TOS Monitors

In a load balancing configuration in DSR mode using TOS field, monitoring its services requires a TOS monitor to be created and bound to these services. A separate TOS monitor is required for each load balancing configuration in DSR mode using the TOS field, because a TOS monitor requires the VIP address and the TOS ID to create an encoded value of the VIP address. The monitor creates probe packets in which the **TOS** field is set to the encoded value of the VIP address. It then sends the probe packets to the servers represented by the services of a load balancing configuration.

With many load balancing configurations, creating a separate custom TOS monitor for each configuration is a significant, cumbersome task. Managing these TOS monitors is also a significant task. Now, you can create wildcard TOS monitors. Create only one wildcard TOS monitor for all load balancing configurations that use the same protocol (for example, TCP or UDP).

A wildcard TOS monitor has the following mandatory settings:

- Type = <protocol>
- TOS = Yes

The following parameters can be set to a value or can be left blank:

- Destination IP
- Destination Port
- TOS ID

A wildcard TOS monitor (with destination IP, Destination port, and TOS ID not set) bound to a DSR service automatically learns the TOS ID and the VIP address of the load balancing virtual server. The monitor creates probe packets with the TOS field set to the encoded VIP address and then sends the probe packets to the server represented by the DSR service.

To create a wildcard TOS monitor by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <Type> -tos YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

To bind a wildcard TOS monitor to a service by using the CLI

At the command prompt, type:

```
bind lb monitor <monitorName> <serviceName>

show lb monitor <monitorName>
<!--NeedCopy-->
```

To create a wildcard TOS monitor by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**.
2. Add a monitor with the following parameter settings:
   - Type = <protocol>
   - TOS = YES

To bind a wildcard TOS monitor to a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Open a service and bind a wildcard TOS monitor to it.

In the following sample configuration, V1, V2, and V3 are load balancing virtual servers of type ANY and has TOS ID set to 1, 2, and 3 respectively. S1, S2, S3, S4, and S5 are services of type ANY. S1 and S2 are bound to both V1 and V2. S3, S4, and S5 and bound to both V1 and V3. WLCD-TOS-MON is a wildcard TOS monitor with type TCP and is bound to S1, S2, S3, S4, and S5.

WLCD-TOS-MON automatically learns the TOD ID and VIP address of virtual servers bound to S1, S2, S3, S4, and S5.

Because S1 is bound to V1 and V2, WLCD-TOS-MON creates two types of probe packets for S1, one with the **TOS** field set to the encoded VIP address (203.0.113.1) of V1 and the other with the VIP address (203.0.113.2) of V2. The Citrix ADC then sends these probe packets to the server represented by S1. Similarly, WLCD-TOS-MON creates probe packets for S2, S3, S4, and S5.


```
add lb monitor WLCD-TOS-MON  TCP -tos YES

Done

add lb vserver V1 ANY 203.0.113.1 * -m TOS  - tosID 1

Done

add lb vserver V2 ANY 203.0.113.2 * -m TOS  - tosID 2

Done

add lb vserver V3 ANY 203.0.113.3 * -m TOS  - tosID 3

Done

add service S1 198.51.100.1 ANY *

Done

add service S2 198.51.100.2 ANY *

Done

add service S3 198.51.100.3 ANY *

Done

add service S4 198.51.100.4 ANY *

Done

add service S5 198.51.100.5 ANY *

Done

bind lb monitor WLCD-TOS-MON S1

Done

bind lb monitor WLCD-TOS-MON S2

Done

```



```
bind lb monitor WLCD-TOS-MON S3

Done

bind lb monitor WLCD-TOS-MON S4

Done

bind lb monitor WLCD-TOS-MON S5

Done

bind lb vserver V1 S1, S2, S3, S4, S5

Done

bind lb vserver V2, S1, S2

Done

bind lb vserver V3 S3, S4, S5

Done
<!--NeedCopy-->
```

Use case 6: Configure load balancing in DSR mode for IPv6 networks by using the TOS field

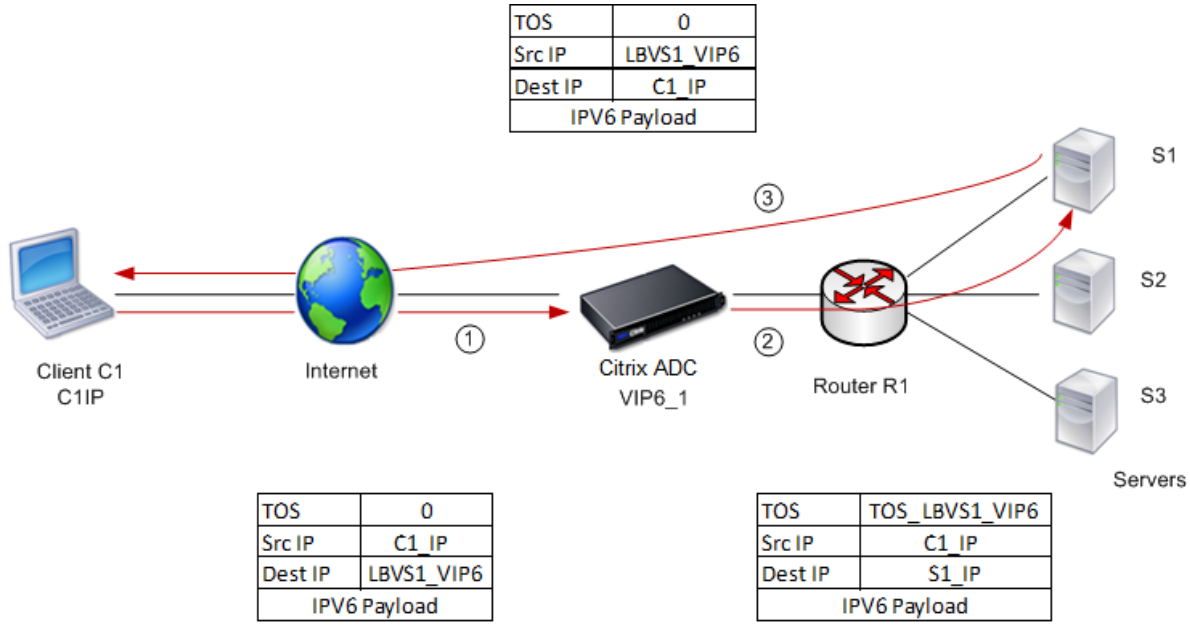
September 14, 2021

You can configure load balancing in Direct Server Return (DSR) mode for IPv6 networks by using the Type of Service (TOS) field when the Citrix ADC appliance and the servers are in different networks.

Note: The TOS field is also called the Traffic Class field.

In DSR mode, when a client sends a request to a VIP6 address on a Citrix ADC appliance, the appliance forwards this request to the server by changing the destination IPv6 address of the packet to the IPv6 address of the server and sets an encoded value of the VIP6 address in the TOS (also called traffic class) field of the IPv6 header. You can configure the server to use the information in the TOS field to derive the VIP6 address from the encoded value, which is then used as source IP address in response packets. Response traffic directly goes to the client, bypassing the appliance.

Consider an example where a load balancing virtual server LBVS1, configured on a Citrix ADC appliance NS1, is used to load balance traffic across servers S1, S2, and S3. The Citrix ADC appliance NS1 and the servers S1, S2, and S3 are in different networks so router R1 is deployed between NS1 and the servers.



The following table lists the settings used in this example.

Entities	Name
IPv6 address of client C1	C1_IP (for reference purposes only)
Load balancing virtual server on NS1	LBVS1
IPv6 address of LBVS1	LBVS1_VIP6 (for references purpose only)
TOS value	TOS_LBVS1_VIP6 (for references purpose only)
Service for server S1 on NS1	SVC_S1
IPv6 address for server S1	S1_IP (for references purpose only)
Service for server S2 on NS1	SVC_S2
IPv6 address for server S1	S2_IP (for references purpose only)
Service for server S3 on NS1	SVC_S3
IPv6 address for server S1	S3_IP (for references purpose only)

Following is the traffic flow in the example scenario:

1. Client C1 sends a request to virtual server LBVS1.

2. LBVS1's load balancing algorithm selects server S1 and the appliance opens a connection to S1. NS1 sends the request to S1 with:
   - TOS field set to TOS_LBVS1_VIP6.
   - Source IP address as C1_IP.
3. The server S1, on receiving the request, uses the information in the TOS field to derive the LBVS1_VIP6 address, which is the IP address of the virtual server LBVS1 on NS1. The server directly sends the response to C1, bypassing the appliance, with:
   - Source IP address set to the derivedLBVS1_VIP6 address so that the client communicates to the virtual server LBVS1 on NS1 and not to server S1.

To configure load balancing in DSR Mode using TOS, perform the following steps on the appliance

1. Enable USIP mode globally.
2. Add the servers as services.
3. Configure a load balancing virtual server with a TOS value.
4. Bind the services to the virtual server.

To configure load balancing in DSR Mode using TOS by using the command line interface

At the command prompt, type:

```
enable ns mode USIP

add service <serviceName> <IP> <serviceType> <port>
<!--NeedCopy-->
```

Repeat the earlier command as many times as necessary to add each server as a service on the Citrix ADC appliance.

```
add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -
    tosId <positive_integer>

bind lb vserver <vserverName> <serviceName>
<!--NeedCopy-->
```

To enable USIP mode by using the configuration utility

Navigate to **System > Settings > Configure Modes**, and select **Use Source IP Address**.

To create services by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services**, and create a service.

To create a load balancing virtual server and bind services by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and create a virtual server.
2. Click in the Service section to bind a service to this virtual server.

Use case 7: Configure load balancing in DSR mode by using IP Over IP

October 26, 2021

You can configure a Citrix ADC appliance to use direct server return (DSR) mode across Layer 3 networks by using IP tunneling, also called *IP over IP* configuration. As with standard load balancing configurations for DSR mode, this allows servers to respond to clients directly instead of using a return path through the Citrix ADC appliance. This improves response time and throughput. As with standard DSR mode, the Citrix ADC appliance monitors the servers and performs health checks on the application ports.

With IP over IP configuration, the Citrix ADC appliance and the servers do not need to be on the same Layer 2 subnet. Instead, the Citrix ADC appliance encapsulates the packets before sending them to the destination server. After the destination server receives the packets, it decapsulates the packets, and then sends its responses directly to the client. This is often referred to as L3DSR.

To configure L3-DSR mode on your Citrix ADC appliance:

- Create a load balancing virtual server. Set the mode to IPTUNNEL and enable sessionless tracking.
- Create services. Create a service for each back-end application and bind the services to the virtual server.
- Configure for decapsulation. Configure either a Citrix ADC appliance or a back-end server to act as a decapsulator.

  > Note:
  >
  > When you use a Citrix ADC appliance, the decapsulation setup is an IP tunnel between the ADC appliances with the back end doing L2DSR to the real servers.

Configure a load balancing virtual server

Configure a virtual server to handle requests to your applications. Assign the service type that matches the service, or use a type of ANY for multiple services.
Set the forwarding method to IPTUNNEL and enable the virtual server to operate in sessionless mode. Configure any load balancing method that you want to use.

To create and configure a load balancing virtual server for IP over IP DSR by using the command line interface

At the command prompt type the following command to configure a load balancing virtual server for IP over IP DSR and verify the configuration:

```
1 add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <
     port> -lbMethod <method> -m <ipTunnelTag> -sessionless [ENABLED |
     DISABLED]
2
3 show lb vserver <name>
4 <!--NeedCopy-->
```

Example:

In the following example, we have selected the load balancing method as sourceIPhash and configured sessionless load balancing.

```
1 add lb vserver Vserver-LB-1 ANY 1.1.1.80 * -lbMethod SourceIPHash -m
     IPTUNNEL -sessionless ENABLED
2 <!--NeedCopy-->
```

To create and configure a load balancing virtual server for IP over IP DSR by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Create a virtual server, and specify Redirection Mode as **IP Tunnel Based**.

Configure services for IP over IP DSR

After creating your load-balanced server, configure one service for each of your applications. The service handles traffic from the Citrix ADC appliance to those applications, and allows the Citrix ADC appliance to monitor the health of each application.

Assign the services to use USIP mode and bind a monitor of type IPTUNNEL to the service for tunnel-based monitoring.

To create and configure a service for IP over IP DSR by using the command line interface

At the command prompt, type the following commands to create a service and optionally, create a monitor and bind it to the service:

```
add service <serviceName> <serverName> <serviceType> <port> -usip <usip
   >

add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <
   iptunnel>

bind service <serviceName> -monitorName <monitorName>
<!--NeedCopy-->
```

Example:

In the following example, a monitor of type IPTUNNEL is created.

```
add monitor mon_DSR PING -destip 1.1.1.80 -iptunnel yes
add service svc_DSR01 2.2.2.100 ANY * -usip yes
bind service svc_DSR01 -monitorName mon_DSR
<!--NeedCopy-->
```

An alternative approach to simplify the routing at both the server and the ADC appliance is to setup both the ADC and server to use an IP from the same subnet. Doing so ensures that any traffic with a destination of a tunnel endpoint is sent over the tunnel. In the example, 10.0.1.0/30 is used.

> **Note:**
>
> The purpose of the monitor is to ensure that the tunnel is active by reaching the loopback of each server through the IP tunnel. If the service is not up, verify whether the outer IP routing between ADC and server is good. Also verify whether the inner IP addresses are reachable through the IP tunnel. Routes might be required on the server, or PBR is added to ADC depending on the chosen implementation.

Example:

```
add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
add netProfile netProfile_DSR -srcIP 10.0.1.2
add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
   YES -netProfile netProfile_DSR
<!--NeedCopy-->
```

To configure a monitor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.

2. Create a monitor, and select **IP Tunnel**.

To create and configure a service for IP over IP DSR by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Create a service and, in **Settings** tab, select **Use Source IP Address**.

To bind a service to a load balancing virtual server by using the command line interface

At the command prompt type the following command:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-DSR-1
<!--NeedCopy-->
```

To bind a service to a load balancing virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and click in the **Services** section to bind a service to the virtual server.

Using the client IP address in the Outer header of tunnel packets

The Citrix ADC supports using the client-source IP address as the source IP address in the outer header of tunnel packets related to direct server return mode using IP tunneling. This feature is supported for DSR with IPv4 and DSR with IPv6 tunneling modes. For enabling this feature, enable the **use client source IP address** parameter for IPv4 or IPv6. This setting is applied globally to all the DSR configurations that use IP tunneling.

To use a client-source IP address as the source IP address by using the CLI

At the command prompt, type:

- `set iptunnelparam -useclientsourceip [YES | NO]`
- `show iptunnelparam`

To use client source IP address as the source IP address by using the GUI

1. Navigate to **System > Network**.
2. In **Settings** tab, click **IPv4 Tunnel Global Settings**.
3. In the **Configure IPv4 Tunnel Global Parameters** page, select **Use Client Source IP** check box.
4. Click **OK**.

To use client source IP address as the source IP address by using the CLI

At the command prompt, type:

- `set ip6tunnelparam -useclientsourceip [YES | NO]`
- `show ip6tunnelparam`

To use client source IP address as the source IP address by using the GUI

1. Navigate to **System > Network**.
2. In **Settings** tab, click **IPv6 Tunnel Global Settings**.
3. In the **Configure IPv6 Tunnel Global Parameters** page, select **Use Client Source IP** check box.
4. Click **OK**.

Decapsulation configuration

You can configure either a Citrix ADC appliance or a back-end server as a decapsulation.

Citrix ADC decapsulation

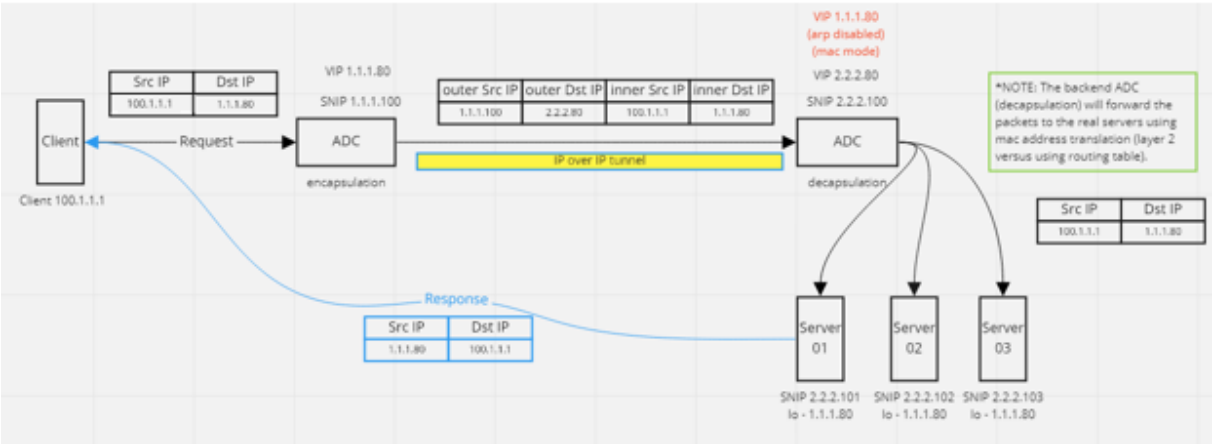
When a Citrix ADC appliance is used as a decapsulation, an IP tunnel must be created in the Citrix ADC appliance. For more details, see Configuring IP Tunnels.

The Citrix ADC decapsulation setup consists of the following two virtual servers:

- The first virtual server receives the encapsulated packet and removes the outer IP encapsulation.
- The second virtual server has the IP of the original service on the front-end ADC and uses MAC translation to forward the packet towards the back end by using the MAC address of the bound services. This setup is typically known as L2DSR. Ensure to disable ARP on this virtual server.

Example setup:

The following illustration shows a decapsulation setup using the ADC appliances.



The complete configuration required for the setup is as follows.

Front-end ADC configuration:

```
add service svc_DSR01 2.2.2.80 ANY * -usip YES -useproxyport NO
add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
<!--NeedCopy-->
```

Back-end ADC configuration:

```
add ipTunnel DSR-IPIP 1.1.1.100 255.255.255.255 *

add service svc_DSR01_01 2.2.2.101 ANY * -usip YES -useproxyport NO
add service svc_DSR01_02 2.2.2.102 ANY * -usip YES -useproxyport NO
add service svc_DSR01_03 2.2.2.103 ANY * -usip YES -useproxyport NO

add lb vserver vs_DSR_DECAP ANY 2.2.2.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED -netProfile netProf_DSR_MBF_noIP

add ns ip 1.1.1.80 255.255.255.255 -type VIP -arp DISABLED -snmp
    DISABLED
add lb vserver vs_DSR_Relay ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    MAC -sessionless ENABLED

bind lb vserver vs_DSR_DECAP svc_DSR01_01
bind lb vserver vs_DSR_DECAP svc_DSR01_02
bind lb vserver vs_DSR_DECAP svc_DSR01_03

bind lb vserver vip_DSR_Relay svc_DSR01_01
bind lb vserver vip_DSR_Relay svc_DSR01_02
bind lb vserver vip_DSR_Relay svc_DSR01_03
```



```
19
20 add netProfile netProf_DSR_MBF_noIP -MBF ENABLED
21 add lb monitor mon_DSR_MAC PING -netProfile netProf_DSR_MBF_noIP
22 bind service svc_DSR01_01 -monitorName mon_DSR_MAC
23 bind service svc_DSR01_02 -monitorName mon_DSR_MAC
24 bind service svc_DSR01_03 -monitorName mon_DSR_MAC
25 <!--NeedCopy-->
```

The following example shows a testing setup using Ubuntu and Red Hat servers running apache2. These commands are set up on each back-end server.

```
1 sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
2 sudo sysctl net.ipv4.conf.all.arp_ignore=1
3 sudo sysctl net.ipv4.conf.all.arp_announce=2
4 sudo sysctl net.ipv4.conf.eth4.rp_filter=2 (The interface has the
     external IP with route towards the ADC)
5 sudo sysctl net.ipv4.conf.all.forwarding=1
6 sudo ip link set dev lo arp on
7 <!--NeedCopy-->
```

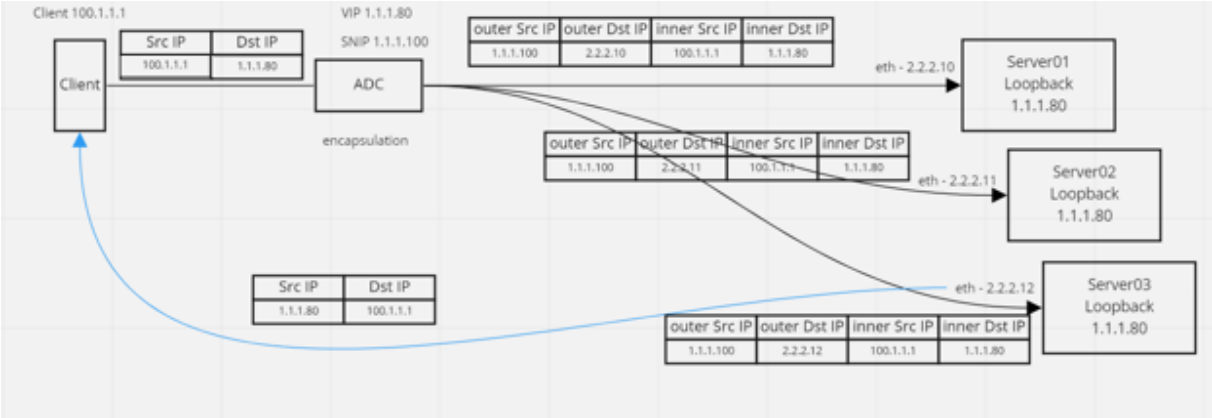
Back-end server decapsulation

When you use the back-end servers as a decapsulation, the back-end configuration varies depending upon the server OS type. You can configure a back-end server as a decapsulation by following these steps:

1. Configure a loop back interface with IP for service IP.
2. Create a tunnel interface.
3. Add a route through tunnel interface.
4. Configure interface settings as required for traffic.

> Note:
>
> Windows OS servers cannot do IP tunneling natively, so the commands are provided as examples for Linux based systems. Third-party plug-ins are available for Windows OS servers, however, that is outside the scope of this example.

The following illustration shows a decapsulation setup using the back-end servers.



Example configuration:

In this example, 1.1.1.80 is the Citrix ADC virtual IP (VIP) address and 2.2.2.10-2.2.2.12 are the back-end server IP addresses. The VIP address is configured in the loopback interface, and a route is added through the tunnel interface. The monitors use the server IP, and tunnel the monitor packets over the IP tunnel using the tunnel endpoints.

The complete configuration required for the setup is as follows.

Front-end ADC configuration:

The following configuration creates a monitor that uses the tunnel endpoint as source. Then, send pings over tunnel to service IP address.

```
add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
add netProfile netProfile_DSR -srcIP 10.0.1.2
add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
    YES -netProfile netProfile_DSR
<!--NeedCopy-->
```

The following configuration creates a VIP for service that uses the original source IP address. Then, forwards traffic over IP tunnel to back-end servers.

```
add service svc_DSR01 2.2.2.10 ANY * -usip YES -useproxyport NO
bind service svc_DSR01 -monitorName mon_DSR

add service svc_DSR02 2.2.2.11 ANY * -usip YES -useproxyport NO
bind service svc_DSR02 -monitorName mon_DSR

add service svc_DSR03 2.2.2.12 ANY * -usip YES -useproxyport NO
bind service svc_DSR03 -monitorName mon_DSR

add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
```



```
bind lb vserver vip_DSR_ENCAP svc_DSR02
bind lb vserver vip_DSR_ENCAP svc_DSR03
<!--NeedCopy-->
```

Back-end server configuration of each server:

The following commands are required for the back-end server to receive the IPIP packet, remove the outer encapsulation, then respond from the loopback to the original client IP. Doing so ensures the IP addresses in the packet received by the client match the IP addresses in the original request.

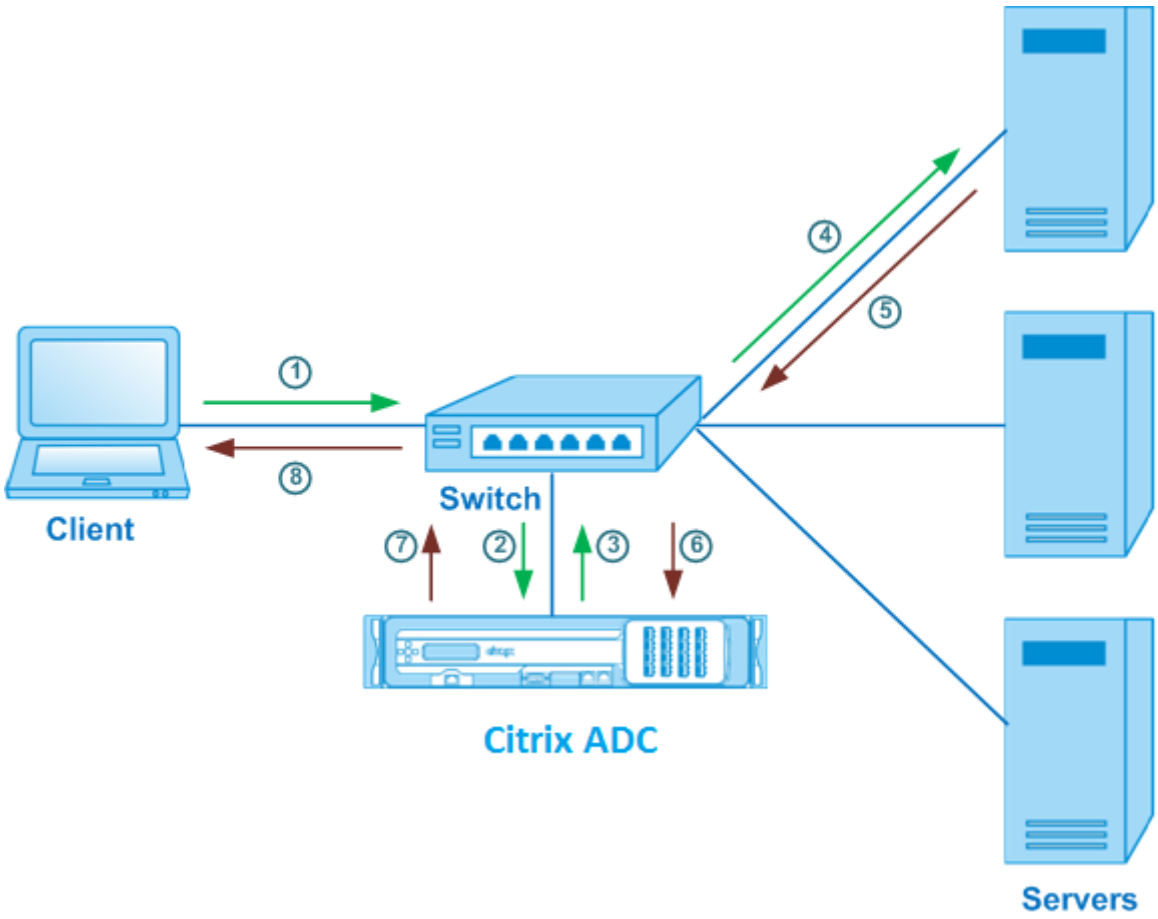
```
modprobe ipip
sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0
ifname tun0 remote 198.51.100.5 local 203.0.113.10
nmcli connection modify tun0 ipv4.addresses '10.0.1.1/30'
nmcli connection up tun0
sudo sysctl net.ipv4.conf.all.arp_ignore=1
sudo sysctl net.ipv4.conf.all.arp_announce=2
sudo sysctl net.ipv4.conf.tun0.rp_filter=2
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo ip link set dev lo arp off
<!--NeedCopy-->
```

Use case 8: Configure load balancing in one-arm mode

September 14, 2021

In a one-arm setup, you connect the Citrix ADC appliance to the network through a single VLAN. The appliance receives the request from the client on a single VLAN and it sends the request to the server on the same VLAN. This is one of the simplest deployment scenarios, where the router, the servers and the appliance are all connected to the same switch. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service.

Figure 1. Load balancing in one-arm mode



In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service. The following table lists the names and values of the entities configured on the appliance in one-arm mode.

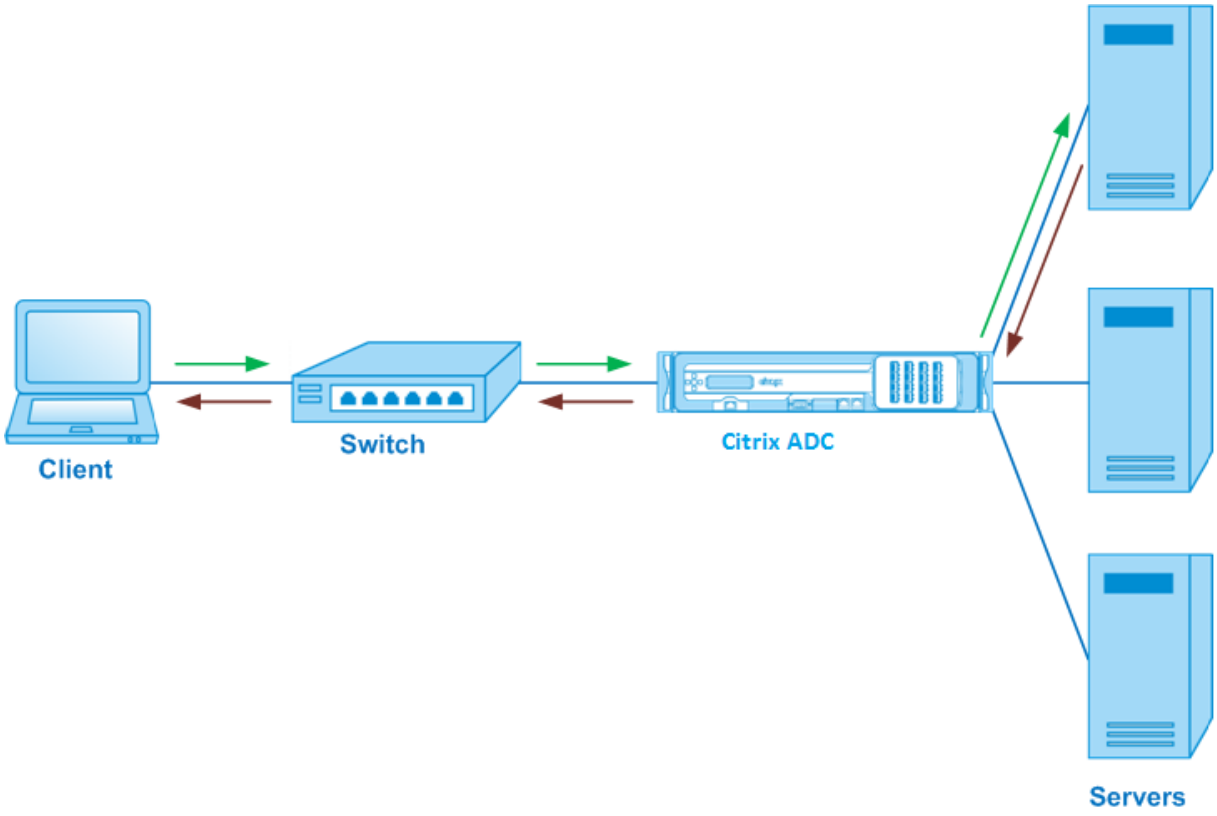
Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
Monitors	TCP	None	None

To configure a load balancing setup in one-arm mode, see Setting Up Basic Load Balancing.

Use case 9: Configure load balancing in the inline mode

September 14, 2021

In an inline mode (also called two-arm mode) setup, you connect the Citrix ADC appliance to the network through multiple VLANs. The appliance receives the request from the client on one VLAN and it sends the request to the server on another VLAN. In the two-arm setup, the appliance is connected between the servers and the client. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service.



The configuration and the entity diagram for inline mode are the same as described in Configuring Load Balancing in One-arm Mode.

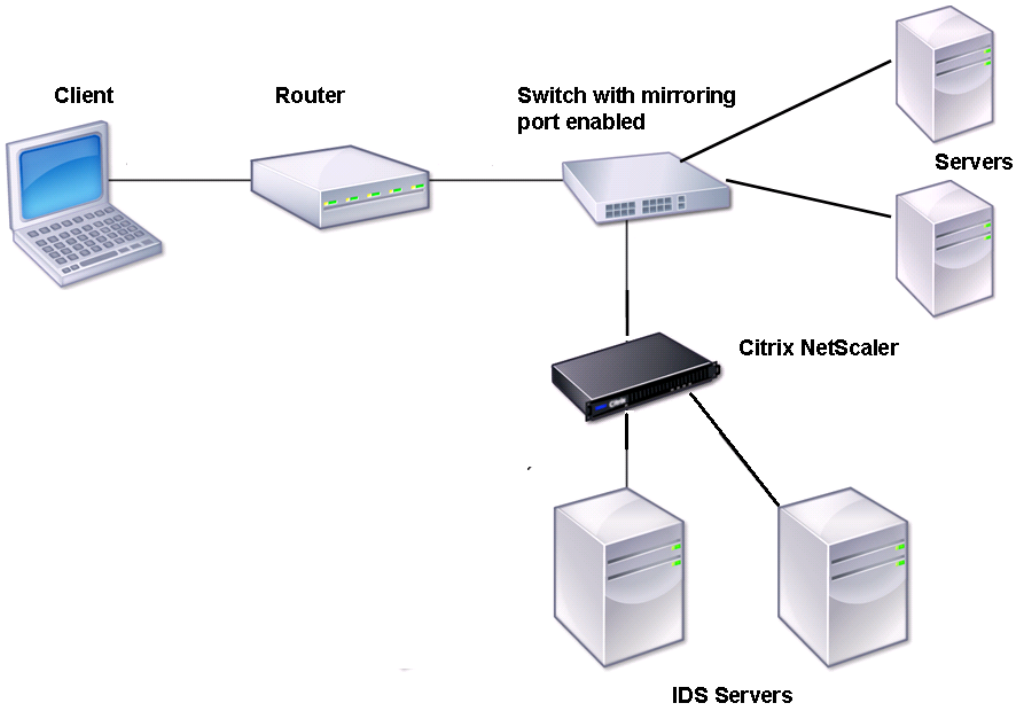
Use case 10: Load balancing of intrusion detection system servers

September 14, 2021

To enable the Citrix ADC appliance to support load balancing of intrusion detection system (IDS) servers, the IDS servers and clients must be connected through a switch that has port mirroring enabled. The client sends a request to the server. Because port mirroring is enabled on the switch,

the request packets are copied or sent to the Citrix ADC appliance virtual server port. The appliance then uses the configured load balancing method to select an IDS server, as shown in the following diagram.

Figure 1. Topology of Load Balanced IDS Servers



Note: Currently, the appliance supports load balancing of passive IDS devices only.

As illustrated in the preceding diagram, the IDS load balancing setup functions as follows:

1. The client request is sent to the IDS server, and a switch with a mirroring port enabled forwards these packets to the IDS server. The source IP address is the IP address of the client, and the destination IP address is the IP address of the server. The source MAC address is the MAC address of the router, and the destination MAC address is the MAC address of the server.
2. The traffic that flows through the switch is mirrored to the appliance. The appliance uses the layer 3 information (source IP address and destination IP address) to forward the packet to the selected IDS server without changing the source IP address or destination IP address. It modifies the source MAC address and the destination MAC address to the MAC address of the selected IDS server.

Note: When load balancing IDS servers, you can configure the SRCIPHASH, DESTIPHASH, or SRCIPDESTIPHASH load balancing methods. The SRCIPDESTIPHASH method is recommended because packets

flowing from the client to a service on the appliance must be sent to a single IDS server.

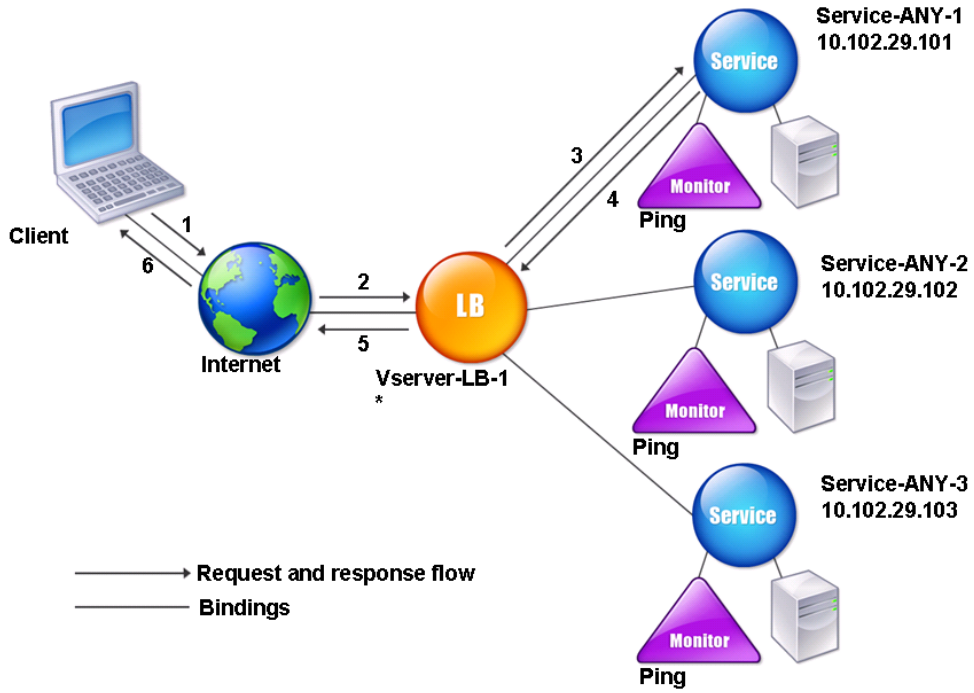
Suppose Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to Vserver-LB-1. The virtual server balances the load on the services. The following table lists the names and values of the entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual server	Vserver-LB-1	*	*	ANY
Services	Service-ANY-1	10.102.29.101	*	ANY
	Service-ANY-2	10.102.29.102	*	ANY
	Service-ANY-3	10.102.29.103	*	ANY
Monitors	Ping	None	None	None

Note: You can use inline mode or one-arm mode for an IDS load balancing setup.

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

Figure 2. Entity Model for Load Balancing IDS Servers



To configure an IDS load balancing setup, you must first enable MAC-based forwarding. Also disable layer 2 and layer 3 modes on the appliance.

To enable MAC-based forwarding by using the command line interface

At the command prompt, type:

```
enable ns mode <ConfigureMode>
<!--NeedCopy-->
```

Example:

```
enable ns mode MAC
<!--NeedCopy-->
```

To enable MAC-based forwarding by using the configuration utility

Navigate to **System > Settings > Configure Modes**, and select **MAC Based Forwarding**.

Next, see "Setting Up Basic Load Balancing", to configure a basic load balancing setup.

After you configure the basic load balancing setup, you must customize it for IDS by configuring a supported load balancing method (such as the SRCIPDESTIP Hash method on a sessionless virtual server) and enabling MAC mode. The appliance does not maintain the state of the connection and only forwards the packets to the IDS servers without processing them. The destination IP address and port remains unchanged because the virtual server is in the MAC mode.

To configure a load balancing method and redirection mode for a sessionless virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
   RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -
   sessionless enabled
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which the -m MAC option is enabled, you must bind a non-user monitor.

To configure a load balancing method and redirection mode for a sessionless virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server and, in Redirection Mode, select MAC Based.
3. In Advanced Settings, click Methods, and select SRCIPDESTIPHASH. Click Traffic Settings, and select Sessionless Load Balancing.

To set a service to use the source IP address by using the command line interface

At the command prompt, type:

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

Example:


```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

To set a service to use the source IP address by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Open a service, and in Settings select **Use Source IP Address**.

For USIP to function correctly, you must set it globally. For more information about configuring USIP globally, see IP Addressing.

Use case 11: Isolating network traffic using listen policies

September 14, 2021

> **Note**
>
> The traffic isolation solution using shadow virtual servers to simulate multitenant isolation is no more recommended. Alternatively, Citrix recommends you to use the Citrix ADC Admin Partitioning feature for such deployments. For more information, see Admin Partitioning.

A common security requirement in a data center is to maintain network path isolation between the traffic of various applications or tenants. One application or tenant's traffic must be isolated from the traffic of other applications or tenants. For example, a financial services company would want to keep the traffic of its insurance department's applications separate from that of its financial services applications. In the past, this was easily achieved through physical separation of network service devices such as firewalls, load balancers, and IdP, and network monitoring and logical separation in the switching fabric.

As data center architectures evolve toward multitenant virtualized data centers, networking services in the aggregation layer of a data center are getting consolidated. This development has made network path isolation a critical component for network service devices and is driving the requirement for ADCs to be able to isolate traffic at the L4 to L7 levels. Furthermore, all the traffic of a particular tenant must go through a firewall before reaching the service layer.

To address the requirement of isolating the network paths, a Citrix ADC appliance identifies the network domains and controls the traffic across the domains. The Citrix ADC solution has two main components: listen policies and shadow virtual servers.

Each network path to be isolated is assigned a virtual server on which a listen policy is defined so that the virtual server listens to traffic only from a specified network domain.

To isolate the traffic, listen policies can be based on several client parameters or their combinations, and the policies can be assigned priorities. The following table lists the parameters that can be used in listen policies for identifying the traffic.

Category	Parameters
Ethernet protocol	Source MAC address, destination MAC address
Network interface	Network ID, receiving throughput, sending throughput, transmission throughput
IP protocol	Source IP address, destination IP address
IPv6 protocol	Source IPv6 address, destination IPv6 address
TCP protocol	Source port, destination port, maximum segment size, payload, and other options
UDP protocol	Source port, destination port
VLAN	ID

Table 1. Client Parameters Used to Define Listen Policies

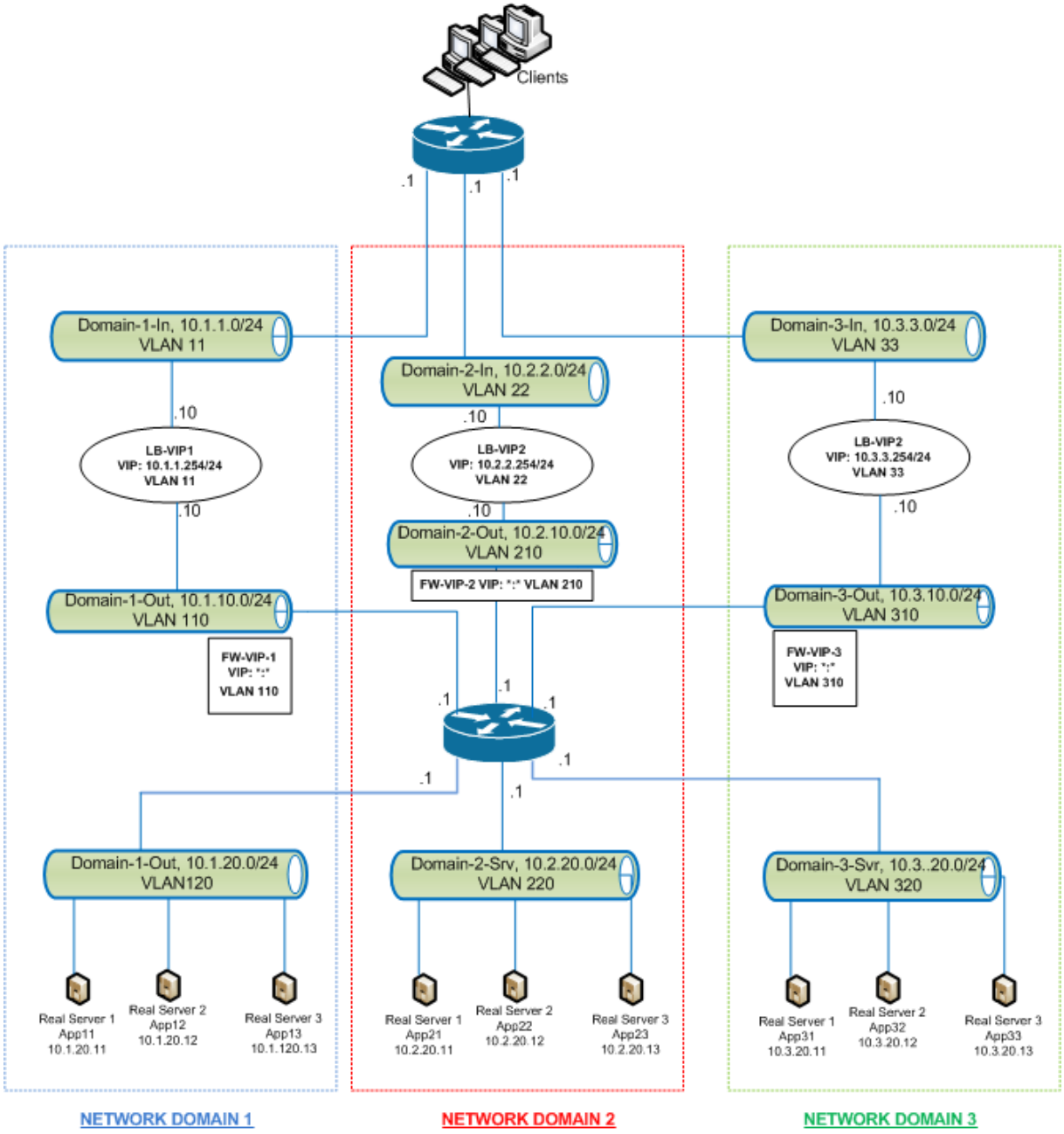
On the Citrix ADC appliance, a virtual server is configured for each domain, with a listen policy specifying that the virtual server is to listen only to traffic for that domain. Also configured for each domain is a shadow load balancing virtual server, which listens to traffic destined for any domain. Each of the shadow load balancing virtual servers has a wildcard (*) IP address and port, and its service type is set to ANY.

In each domain, a firewall for the domain is bound as a service to the shadow load balancing virtual server, which forwards all traffic through the firewall. Local traffic is forwarded to its destination, and traffic destined for another domain is forwarded to the firewall for that domain. The shadow load balancing virtual servers are configured for MAC mode redirection.

How network paths are isolated

The following figure shows a typical traffic flow across domains. Consider the traffic flow within Network Domain 1, and between Network Domain 1 and Network Domain 2.

Figure 1. Isolating Network Path



Traffic within network domain 1

Network Domain 1 has three VLANs: VLAN 11, VLAN110, and VLAN120. The following steps describe the traffic flow.

- A client from VLAN 11 sends a request for a service available from the service pool in VLAN 120.
- The load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110. The virtual server in VLAN 110 forwards the request to shadow load balancing virtual server FW-VIP-1.

- FW-VIP-1, which is configured to listen to traffic from VLAN 110, receives the request and forwards it to VLAN 120.
- The load balancing virtual server in VLAN 120 load balances the request to one of the physical servers, App11, App12, or App13.
- The response sent by the physical server returns by the same path to the client in VLAN 11.

This configuration ensures that traffic is always segregated inside the Citrix ADC for all the traffic that originates from a client.

Traffic between network domain 1 and network domain 2

Network Domain 1 has three VLANs: VLAN 11, VLAN 110, and VLAN 120. Network Domain 2 also has three VLANS: VLAN 22, VLAN 210, and VLAN 220. The following steps describe the traffic flow from VALN 11 to VLAN 22.

- A client from VLAN 11, which belongs to Network Domain 1, sends a request for a service available from the service pool in VLAN 220, which belongs to the Network Domain 2.
- In Network Domain 1, the load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110.
- Shadow load balancing virtual server FW-VIP-1, which is configured to listen to VLAN 110 traffic destined to any other domain, receives the request and forwards it to firewall virtual server FW-VIP-2 because the request is destined to a physical server in Network Domain 2.
- In Network Domain 2, FW-VIP-2 forwards the request to VLAN 220.
- The load balancing virtual server in VLAN 220 load balances the request to one of the physical servers, App21, App22, or App23.
- The response sent by the physical server returns by the same path through the firewall in Network Domain 2 and then to Network Domain 1 to reach the client in VLAN 11.

Configuration Steps

To configure network path isolation by using listen policies, do the following:

- Add listen policy expressions. Each expression specifies a domain to which traffic is destined. You can use the VLAN ID or other parameters to identify the traffic.
- For each network domain, configure two virtual servers as follows:
  - Create a load balancing virtual server for which you specify a listen policy that identifies the traffic destined for this domain. You can specify the name of an expression created earlier, or you can create an expression while creating the virtual server.
  - Create another load balancing virtual server, referred to as shadow virtual server, for which you specify a listen policy expression that applies to traffic destined for any domain. On this virtual server, set the service type to ANY and the IP address and port to an asterisk (*). Enable MAC-based forwarding on this virtual server.

  - Enable the L2 Connection option on both the virtual servers.
    Generally, to identify a connection, the Citrix ADC appliance uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.
- Add services representing the server pools in the domain, and bind them to the virtual server.
- Configure the firewall for each domain as a service, and bind all the firewall services to the shadow virtual server.

To isolate network traffic by using the command line interface

At the command prompt, type the following commands:

```
add policy expression <expressionName> <listenPolicyExpression>

add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -
   listenPolicy <expressionName>
<!--NeedCopy-->
```

Add a load balancing virtual server for each domain. This virtual server is for traffic of the same domain.

```
add lb vserver <name> ANY * * -l2conn ON -m MAC -listenPolicy <
   expressionName>
<!--NeedCopy-->
```

Add a shadow load balancing virtual server for each domain. This virtual server is for traffic of other domains.

Example:

```
add policy expression e110 client.vlan.id==110
add policy expression e210 client.vlan.id==210
add policy expression e310 client.vlan.id==310
add policy expression e11 client.vlan.id==11
add policy expression e22 client.vlan.id==22
add policy expression e33 client.vlan.id==33

add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -
   listenPolicy e11
-cltTimeout 180 -l2Conn ON

add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE -
   listenPolicy e22
```



```
-cltTimeout 180 -l2Conn ON

add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE -
    listenPolicy e33
-cltTimeout 180 -l2Conn ON


add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod
    ROUNDROBIN - listenPolicy e110 -Listenpriority 1 -m MAC -cltTimeout
    120

add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod
    ROUNDROBIN - listenPolicy e210 -Listenpriority 2 -m MAC -cltTimeout
    120

add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod
    ROUNDROBIN - listenPolicy e310 -Listenpriority 3 -m MAC -cltTimeout
    120


add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
    DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
    NO -TCPB NO -CMP NO

add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
    DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
    NO -TCPB NO -CMP NO

add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
    DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
    NO -TCPB NO -CMP NO


bind lb vserver FW-VIP-1 RD-1

bind lb vserver FW-VIP-2 RD-2

bind lb vserver FW-VIP-3 RD-3
<!--NeedCopy-->
```


To isolate network traffic by using the configuration utility

1. Add services representing the servers, as described in Creating a Service.
2. Add each firewall as a service:
   a) Navigate to **Traffic Management > Load Balancing > Services**.
   b) Create a service, specifying the protocol as ANY, server as firewall's IP address, and port as 80.
3. Configure a load balancing virtual server.
4. Configure the shadow load balancing virtual server.
5. For each network domain, repeat steps 3 and 4.
6. From the Load Balancing Virtual Servers pane, open the virtual servers that you created and verify the settings.

Use case 12: Configure XenDesktop for load balancing

September 14, 2021

For an improved performance in the delivery of virtual desktop applications, you can integrate the Citrix ADC appliance with Citrix XenDesktop and use the Citrix ADC load balancing feature to distribute the load across the Web Interface servers and the Desktop Delivery Controller (DDC) servers.

Generally, you use XenDesktop in situations where applications are not compatible with running on a terminal server or XenApp, or if each virtual desktop has unique requirements. In such cases, you need one desktop host for each user that connects. However, the hosts can be pooled so that you need only one host for each currently connected user.

The core application service deployed for XenDesktop is the Desktop Delivery Controller (DDC). The DDC is installed on a server, and its main function is to register desktop hosts and broker client connections to them.

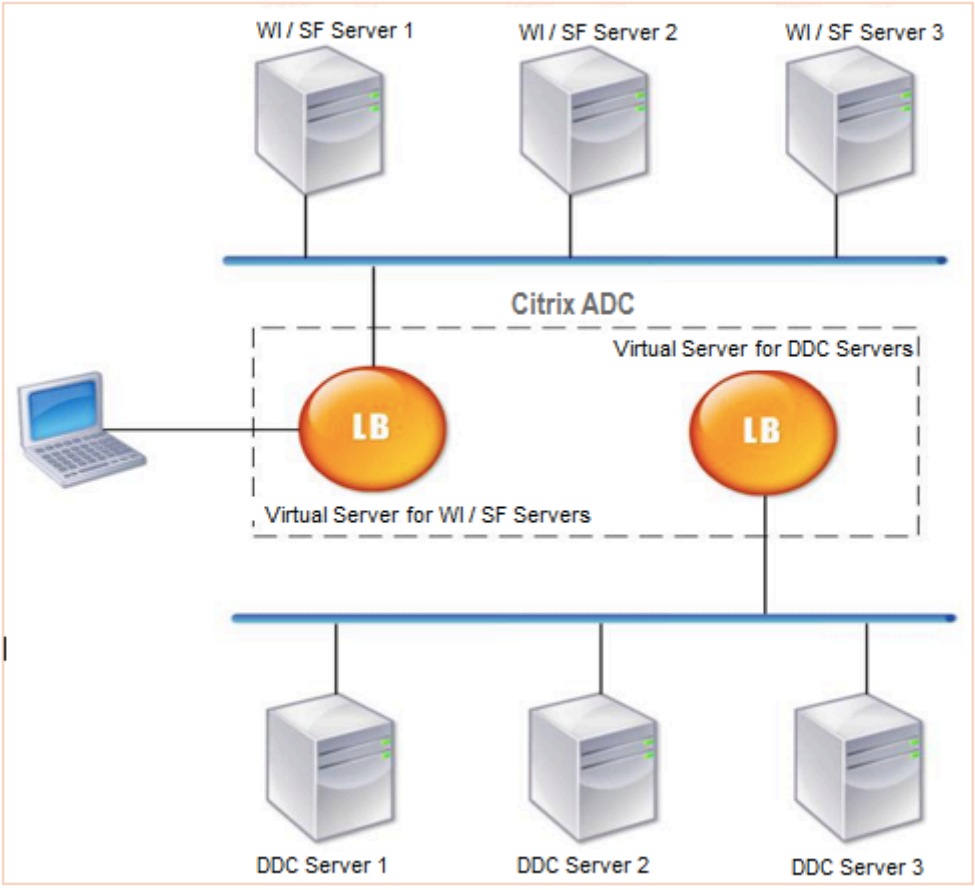
The DDC also authenticates users and manages the assembly of the users' virtual desktop environments by controlling the state of the desktops, and starting and stopping the desktops.

Generally, multiple DDCs are installed to enhance availability.

The Web Interface servers provide secure access to virtual desktops. The Web Interface is the initial connection portal to the Desktop Delivery Controller (DDC). The Web browser on the user's device sends information to the Web server, which communicates with the server farm to provide the user with access to the virtual desktop.

The following figure shows the topology of a Citrix ADC appliance working with XenDesktop.

Figure 1. Load Balancing of XenDesktop



> **Note**
>
> Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the Citrix ADC appliance. You can use the HTTP protocol for communication between the Citrix ADC and the DDC servers even though you use the SSL protocol for communication with the client.

To configure load balancing for XenDesktop by using the GUI

1. Create a service.
   a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
   b) Create a service by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
2. Create a load balancing virtual server.
   a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and click **Add**.
   b) Create a virtual server by specifying a name, an IP address, a port, and a protocol type and

then click **OK**.

3. Bind the service to the load balancing virtual server.
4. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and select a server.
   a) Click **Edit**.
   b) In the **Services and Service Groups**, click **>** and click **Add Binding**.
   c) Select the service you want to bind and enter the weight value.
   d) Click **Bind**.

To configure load balancing for XenDesktop by using the command line interface

- To create a service, at the command prompt, type:

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- To create a virtual server, at the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

Example:

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- To bind a service to a load balancing virtual server, at the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

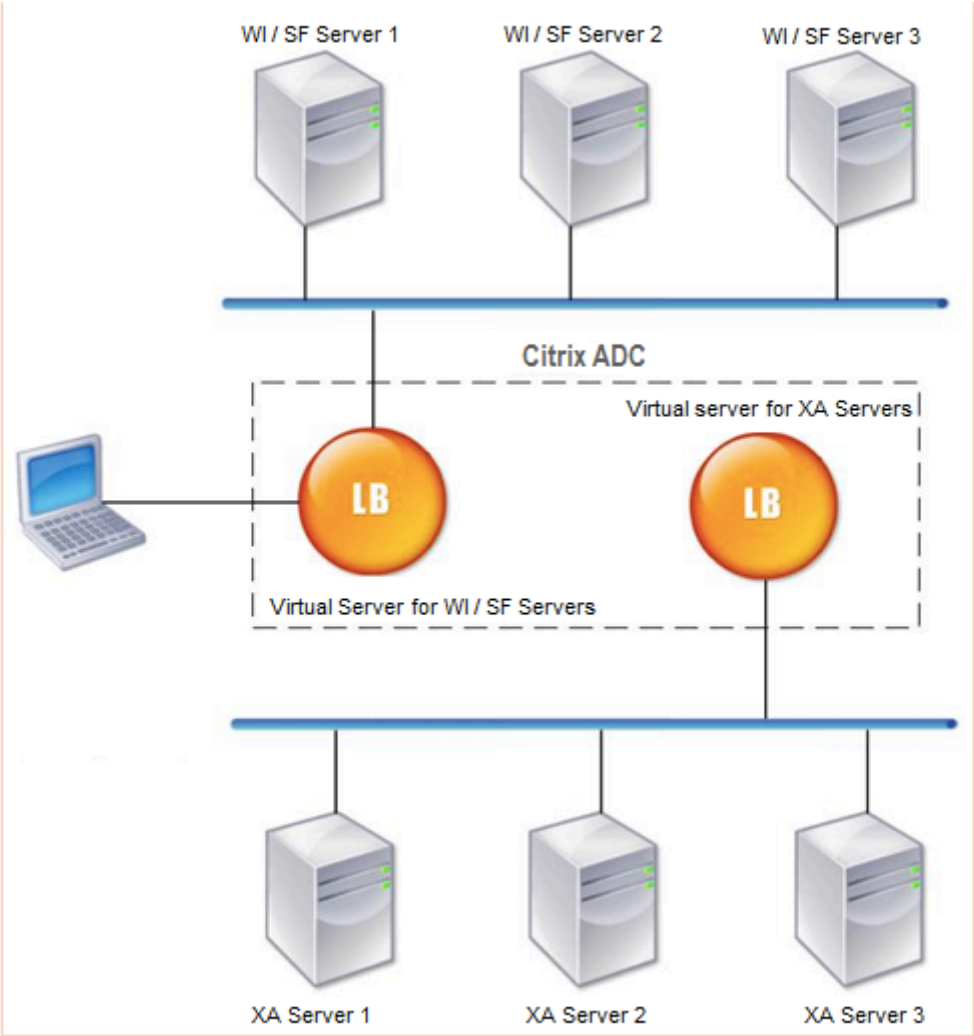
```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

Use case 13: Configure XenApp for load balancing

September 14, 2021

For efficient delivery of applications, you can integrate the Citrix ADC appliance with Citrix XenApp and use the Citrix ADC load balancing feature to distribute the load across the XenApp server farms. The following figure is a topology diagram of such a setup.

Figure 1. Load Balancing of XenApp



The Web Interface servers provide secure access to XenApp application resources through the user's Web browser. The Web Interface client presents to the users all the resources, such as applications, content, and desktops that are made available in the XenApp server farms. Users can access the published resources through a standard Web browser or through the Citrix online plug-in.

The Web browser on the user's device sends information to the Web server, which communicates with the servers on the server farm to provide the user with access to the resources.

The Web Interface and the XML Broker are complementary services. The Web Interface provides users

with access to applications, and the XML Broker evaluates the user's permissions to determine which applications appear in the Web Interface.

The XML service is installed on all the servers in the server farm. The XML service specified in the Web Interface functions as an XML broker. Based on the user credentials passed by the Web Interface server, the XML Broker server sends a list of applications accessible to the user.

In large enterprises where multiple Web Interface servers and XML Broker servers are deployed, Citrix recommends load balancing these servers by using the Citrix ADC appliance. Configure one virtual server to load balance the Web Interface servers and another for the XML Broker servers. The load balancing method and other features can be configured on the virtual server as required.

> **Note**
>
> Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the Citrix ADC. You can use the HTTP protocol for communication between the Citrix ADC and the WI servers even though you use the SSL protocol for communication with the client.

To configure load balancing for XenApp by using the GUI

1. Create a service.
   a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
   b) Create a service by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
2. Create a load balancing virtual server.
   a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and click **Add**.
   b) Create a virtual server by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
3. Bind the service to the load balancing virtual server.
4. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and select a server.
   a) Click **Edit**.
   b) In the **Services and Service Groups**, click **>** and click **Add Binding**.
   c) Select the service you want to bind and enter the weight value.
   d) Click **Bind**.

To configure load balancing for XenApp by using the command line interface

- To create a service, at the command prompt, type:


```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- To create a virtual server, at the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

- To bind a service to a load balancing virtual server, at the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

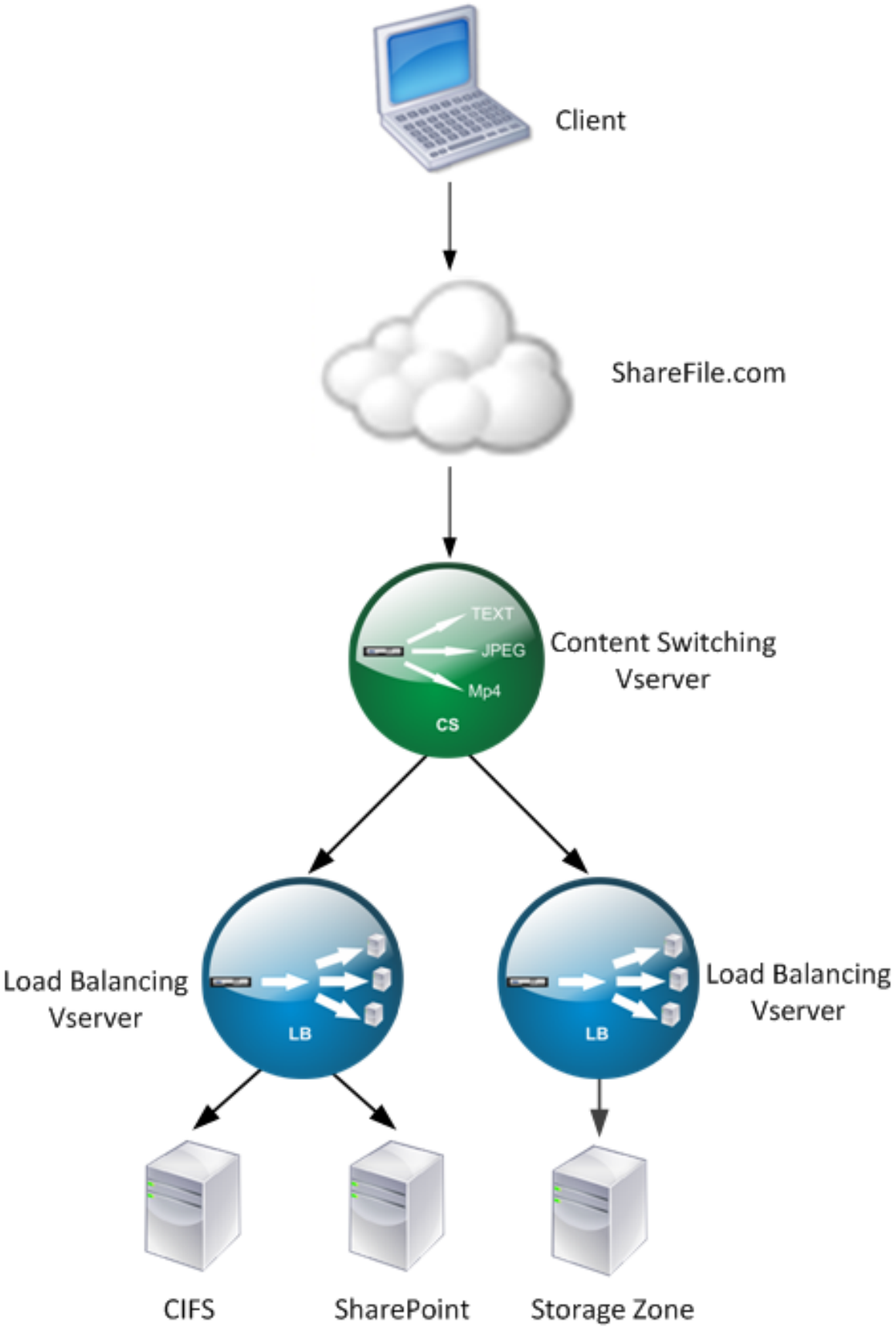
Use case 14: ShareFile wizard for load balancing Citrix ShareFile

September 14, 2021

You can configure load balancing for Citrix ShareFile using the wizard. The Citrix ShareFile wizard helps in setting up the load balancing configuration for the ShareFile site based on the type of content requested. The content switching server directs the request based on whether it is a StorageZone, CIFS, or a SharePoint request. The content switching is based on policies. The wizard auto generates the policies to identify whether the request is for StorageZone, CIFS, or SharePoint. The content switching virtual server uses these policies to direct the request to the correct load balancing server.

A typical data flow can be depicted as shown in the following diagram.

Figure 1. ShareFile Data Load Balancing

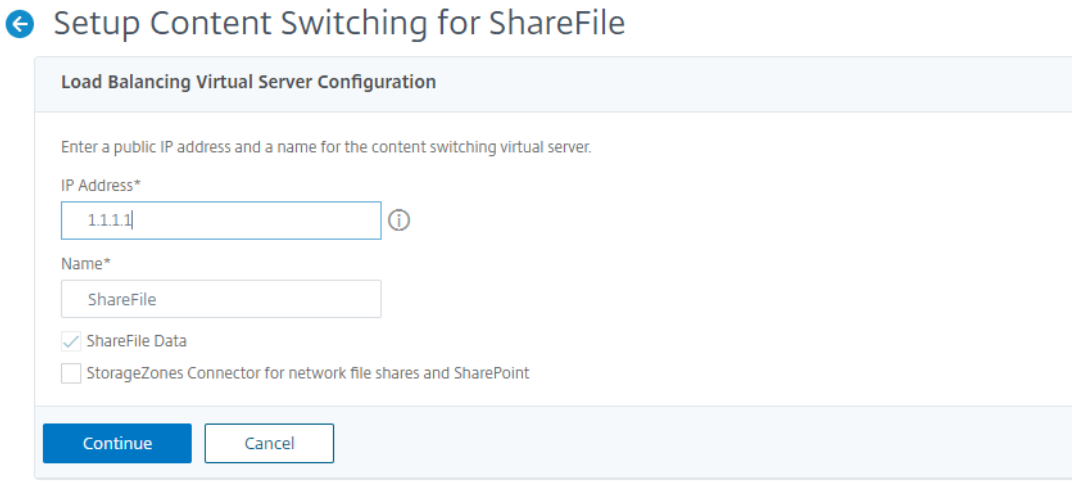


You can view the load balancing virtual servers that the ShareFile wizard creates by navigating to **Traffic Management > Virtual Servers** and **Services > Virtual Servers**. You cannot manually remove the virtual servers created using the ShareFile wizard. Use the wizard to remove the virtual servers.

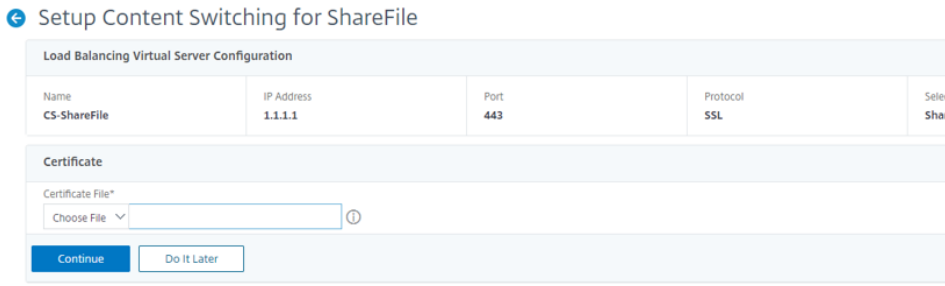
Citrix ADC uses the LDAP authentication for SharePoint or CIFS request. Hash authentication is used for authenticating requests for StorageZones.

To configure a Citrix ADC appliance for load balancing Citrix ShareFile

1. In the navigation pane, click **Traffic Management**.
2. Under **Citrix ShareFile** section, click **Setup Citrix ADC for ShareFile**.
3. On the **Setup Content Switching for ShareFile** page, provide the following information:
   - IP Address: IP address of the content switching virtual server.
   - Name: Name of the content switching virtual server.
   - If you want to set up load balancing for CIFS or SharePoint, click the **StorageZone Connector for Network File Shares/SharePoint** check box and then click **Continue**. By default, the **ShareFile Data** check box is selected.



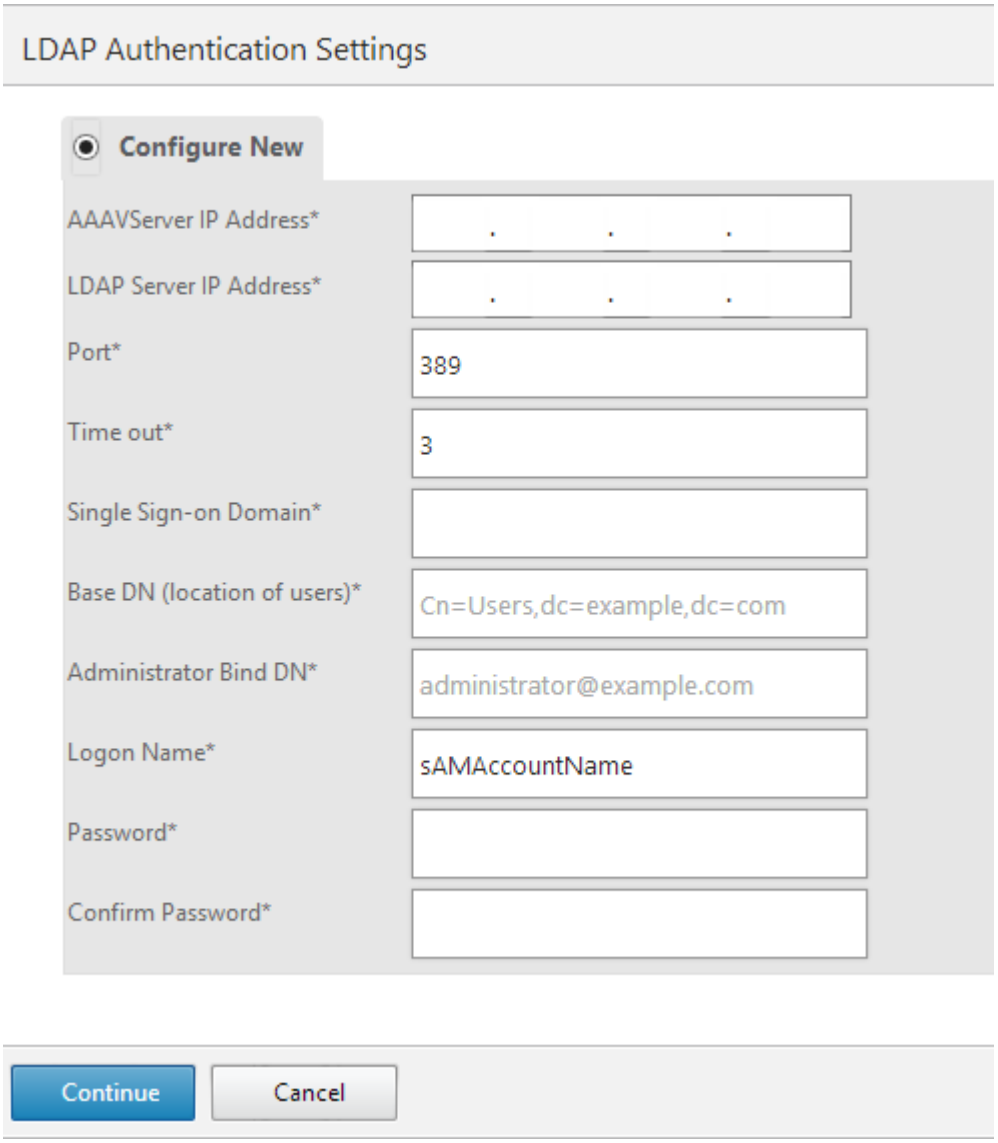
4. Provide a valid certificate. If you have a certificate, click **Choose Certificate** and from the drop-down list select the certificate. If you have to install a certificate, click **Install Certificate** and provide the Certificate-Key pair.



5. Click **Continue**.
6. In the **Add New StorageZone Controller** dialog box, specify the values of the following parameters:

- StorageZone Controller IP Address— IP address
- Port— Port number. The default value is 443.
- Protocol— Select from HTTPS or HTTP

7. Click **Create** and then click **Done**. The wizard automatically creates a service and autogenerates the name of the service.
8. If you chose load balancing for CIFS or SharePoint in step 4.c, then specify the values for LDAP Authentication Settings:
   - Citrix ADC AAA virtual server IP Address— IP address of Citrix ADC AAA virtual server
   - LDAP Server IP Address— IP Address of the LDAP server
   - Port— Port number. The default value is 389
   - Timeout— The time-out value in minutes
   - Single sign-on Domain— Single sign-on domain name
   - Base DN— Base domain name
   - Administrator Bind DN— LDAP account name with the domain name, for example, administrator@domainname.com
   - Logon Name— Logon name is the sAMAccountName
   - Password and Confirm Password— Enter the password and confirm the password



9. Click **Continue** and then click **Done**.

To remove load balancing configuration for ShareFile

1. In the navigation pane, click **Traffic Management**.
2. Under **Citrix ShareFile** section, click **Remove ShareFile Configuration**.

Use case 15: Configure layer 4 load balancing on the Citrix ADC appliance

September 14, 2021

The layer 4 load balancer (TCP and UDP ports) uses information provided in the networking transport layer for routing client requests across the server groups.

When a layer 4 connection is established between a client and a server, it has a packet view of traffic exchanged between them. The layer 4 load balancer makes its routing decisions based on the address information extracted from the first few packets in the TCP stream, and doesn't inspect the packet content. Therefore, the layer 4 load balancing is also called as connection-based load balancing.

The layer 4 load balancer monitors the health of a server. Traffic is not routed to the server if it is DOWN.

The layer 4 load balancing is useful for various applications that uses TCP or UDP payloads. Such protocols exchange data as TCP payload and don't have a specific structure to follow.

To configure layer 4 load balancing using the command line interface

At the command prompt, type:

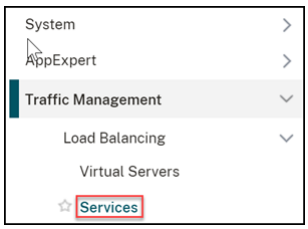
```
add service <name> <serverName> <serviceType> <port>
add lb vserver <name> <serviceType> <ip> <port>
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
add service TCPservice 192.0.2.3 TCP 1
add lb vserver TCPserver TCP 192.0.2.4 1
bind lb vserver TCPserver TCPservice
<!--NeedCopy-->
```

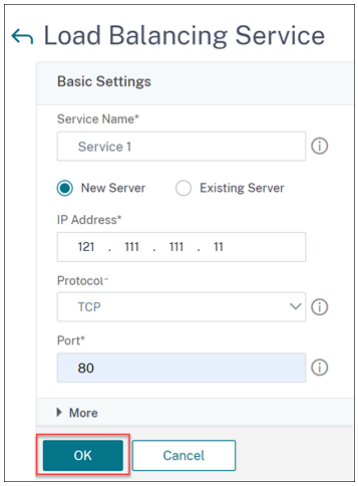
To configure layer 4 load balancing using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.



2. Click **Add** to a create a service.
3. Specify the required details in **Service Name** and **IP Address**.
4. Select either **TCP** or **UDP** in **Protocol**.

5. Click **OK**.



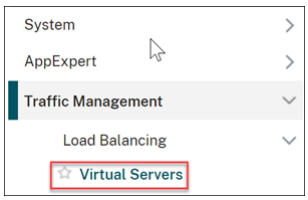
6. Click **Done**.

   A service is created.

   When you create a service using UDP as the transport layer protocol, a ping monitor (built-in monitor) is automatically bound to the service. When you create a service using TCP as the transport layer protocol, a **tcp_default** monitor is automatically bound to the service.

   For the load balancing setup, you can bind your service to a different type of monitor or multiple monitors. For advance monitoring requirements you can use the **tcp-ecv** monitor and configure the request and response messages.

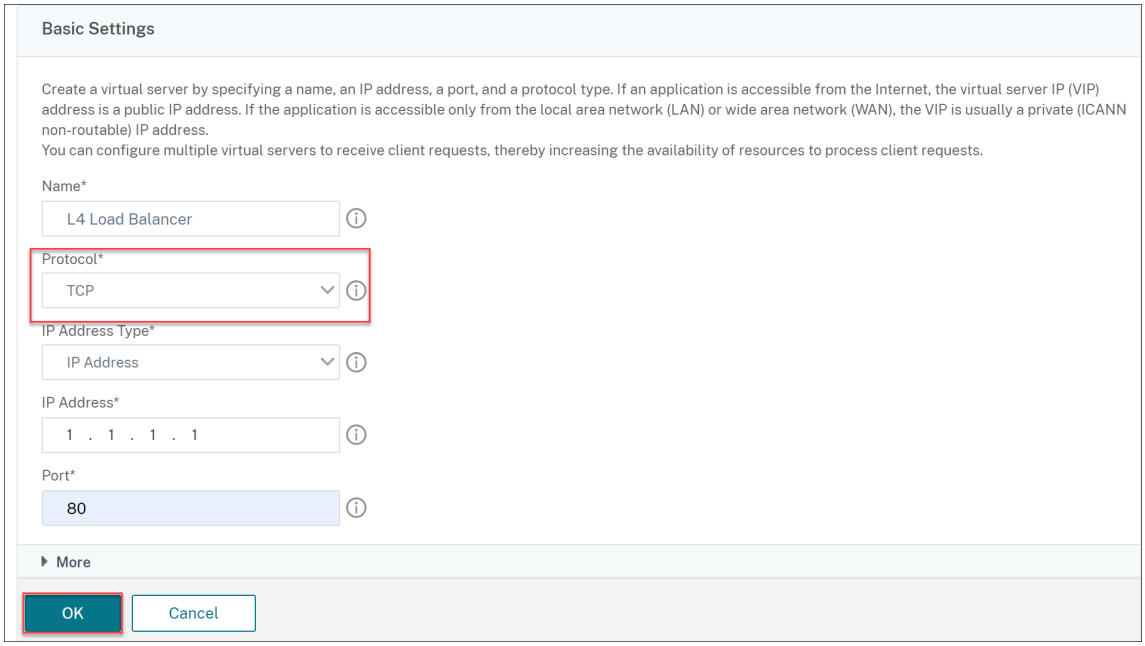
7. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.



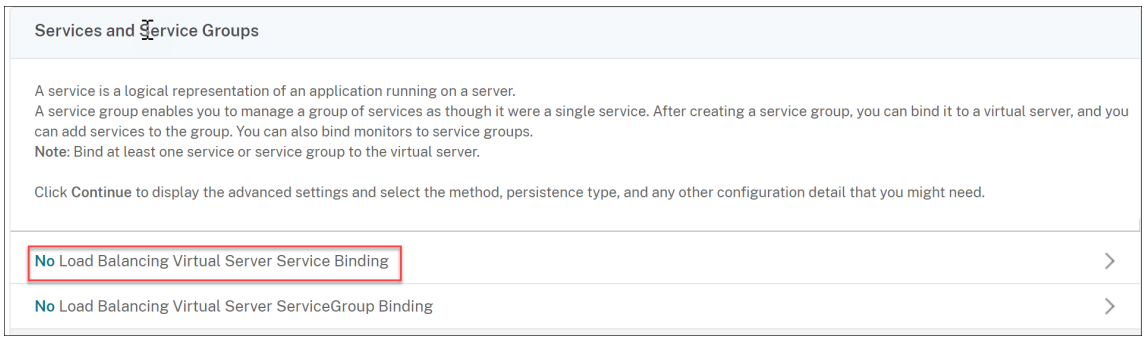
8. Click **Add** to create a new virtual server.

   When the load balancing is configured, you can connect to the load-balanced website, application, or server through the virtual server's IP address or FQDN.

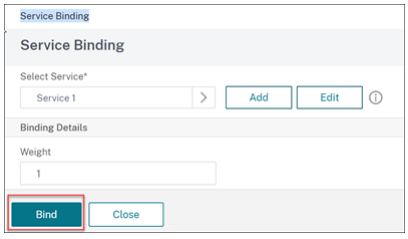
9. Specify the required details in **Name**, **IP Address Type**, and **IP Address**.
10. Select either **TCP** or **UDP** in **Protocol**.
11. Type a port number (0–1023 based on the type of service) in **Port**.
12. Click **OK**.



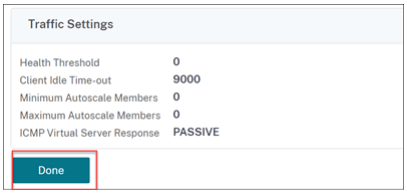
13. Click **No Load Balancing Virtual Server Service Binding** in **Services and Service Groups**.



14. In the **Service Binding** page, select **Click to Select** in **Select Service**.
15. Select the service to be bound and click **Select**.
16. Click **Bind** to bind the service to the virtual server.



17. Click **Continue**.
18. Click **Done**.



The layer 4 load balancing virtual server configuration is completed.

Troubleshooting

September 14, 2021

If the load balancing does not work as expected after you have configured it, you can use some common tools to access Citrix ADC resources and diagnose the problem.

Resources for Troubleshooting Load Balancing

For best results, use the following resources to troubleshoot a content switching issue on a Citrix ADC appliance:

- Latest ns.conf file
- Relevant `newnslog` files
- Ethereal packet traces recorded on the appliance and relevant client, if possible
- The ns.log file

In addition to the above resources, the following tools expedite troubleshooting:

- A browser add-on tool that can display HTTP headers. This can be used to troubleshoot persistency related issues.
- The Wireshark application customized for the Citrix ADC trace files.

Troubleshooting Load Balancing Issues

- **Issue**

CPU usage reaches 100% when a user monitor is bound to a service that is bound to a virtual server on which the -m MAC option is enabled.

- **Resolution**

Bind a non-user monitor to the service.

- **Issue**

  I created a user script for monitoring, but it is not working.

Resolution

Check the number of arguments in the script. The limit is 512. A script with more than 512 arguments might not work properly. Use the nsumon-debug.pl script from the CLI to debug the script.

- **Issue**

  I see a lot of monitor probes, and they seem to be increasing the network traffic unnecessarily. Is there a way to off the monitor probes?

  Resolution

  You can set off the monitor probe connections, by disabling the monitor or setting the value of the healthMonitor parameter in the set service command to NO. With the NO option, the appliance shows the service as UP at all times.

- **Issue**

  I have set up monitors for services, but connections are still directed to servers that are DOWN.

  Resolution

  You probably need to decrease the monitor probe intervals. The Citrix ADC appliance does not detect the DOWN state until the monitor sends a probe.

- **Issue**

  A metric bound to the monitor is present in the local and custom metric tables.

  Resolution

  Add the local prefix to the metric name if the metric is chosen from the local metric table. However, if the metric is chosen from the custom table, you don't need to add any prefix.

- **Issue**

  The monitor probes to a service are not reaching the service.

  Resolution

  Check whether you have set a limit on the number of connections for a service. If yes, exempt monitor-probe connections from this limit by setting the monitorSkipMaxClient parameter to ENABLED.

- **Issue**

  I am able to ping the servers, but the state of the services is always shown as DOWN.

  Resolution

  Check the type of monitors configured. For example, if a server is not configured for SSL and you use an HTTPS monitor, the state of the service is marked as DOWN. In this case using a TCP monitor must change the state of the service to UP.

- **Issue**

  Setting a weight for load monitors does not help in deciding the state of the service.

  Resolution

  Load monitors cannot decide the state of the service. Therefore, setting a weight on the load monitors is inappropriate.

- **Issue**

  A service is not stable.

  Resolution

  Consider troubleshooting the following components:

  - Verify that a correct server is bound to the service.
  - Verify the type of monitor bound to the service.
  - Verify the reasons for the monitor failures. You can open a service from the Services page and verify the details for the number of probes, failures, and last response status for the monitor in the Monitors tab of the Configure service dialog box. To display the details, click the monitor configured.
  - If it is a custom monitor, bind a TCP or ping monitor to the service and verify the status of the monitor. If this resolves the issue, there is some problem with the custom monitor and the monitor requires further investigation.
  - You can record packet traces on the Citrix ADC appliance and verify the monitor probes and server response for further investigation.

- **Issue**

  The virtual IP (VIP) address is not stable or its status is displayed as DOWN.

  Resolution

  Consider troubleshooting the following components:

  - Verify that the load balancing feature is licensed.
  - Verify that the feature is enabled.
  - Verify that an appropriate service is bound to the virtual server.
  - If the status of the VIP address is displayed as DOWN, verify that an administrator has enabled the service. If it is not, the status of the service must be Out-Of-Service. In such a case, you must enable the service and verify if the issue is resolved.
  - Verify the service(s) bound to the virtual server and complete the troubleshooting steps mentioned for service not stable issue.
  - If the VIP address is not stable, all the services bound to the virtual server must fail. Therefore, verify if all the services are failing at the same time. If it is so, there is a network issue between the Citrix ADC appliance and the servers.

- **Issue**

  The site is experiencing uneven load balancing.

  Resolution

  Consider troubleshooting the following components:

  - Verify the load balancing method configured on the appliance.
  - Verify weights associated with the services are as expected.
  - If the load balancing method is other than round robin, verify the number of connections to the server logged in the `newnslog` file. You can run the following command to verify the number on the `newnslog` file:

    ```
    ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

    Verify the services for the specific virtual server and check for the Response time, Open Established connections (OE), number of requests, Persistent requests and persistent rate (P) to troubleshoot the issue further.
  - If the load balancing method is round robin, verify the persistent requests as mentioned in the preceding step. Additionally, verify if the service is not stable. If it is not, complete the troubleshooting steps mentioned for service not stable issue
  - Verify if persistency is configured on the appliance.
  - Verify if any service is not stable. If yes, complete the troubleshooting steps mentioned for service not stable issue.

- **Issue**

  The service status is displayed as DOWN.

  Resolution

  Consider troubleshooting the following components:

  - Verify whether a SNIP address is configured.
  - Verify that appropriate monitors are bound to the service.
  - If custom monitors are bound to the service, bind a TCP or ping monitor to the service and verify the status of the monitor. If this resolves the issue, there is some problem with the custom monitor and the monitor requires further investigation.
  - Verify if the status of service is displayed as DOWN for the server that is in another subnet. If yes, verify if Use Subnet IP (USNIP) resolves the issue because this can be due to the MIP address being unable to communicate to the server.

- **Issue**

  There is an issue with the response time.

Resolution

Consider troubleshooting the following components:

- Verify the server response time from the service stats either by running the following command:

```
## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
```

- Check for service not stable and service status being displayed as DOWN issues.

- **Issue**

One of the servers is serving more requests than the other load balanced servers.

Resolution

Consider troubleshooting the following components:

- Verify the load balancing method. Use the round robin method to distribute the client request equally regardless of the load on the servers.
- Determine whether persistence is enabled for the load balancing configuration. If persistence is enabled, a given server might be carrying a heavier load to maintain its session, especially If the persistence sessions are long.
- Verify whether weights are assigned to each service. Assigning proper weights helps in proper load distribution.

- **Issue**

Connections to a specific load balanced server are stalled. For example, all connections to one Outlook server might be stalled.

Resolution

Consider troubleshooting the following components:

- Verify the load balance method. If it is round robin, consider changing the method to least connections.
- Consider reducing the monitor time-out period. A shorter timeout period helps in marking a service as DOWN sooner, which would help in directing the traffic to the server which is functional.
- If the connections are stalled for a long period, a surge queue might build. Consider flushing the surge queue to avoid a sudden spike in load on the server.
- If the servers are working at their maximum level, consider adding a new server for better performance.

- **Issue**

A majority of the connections are directed to a particular server, even when the least connections method for load balancing is configured.

Resolution

Determine whether persistence is configured and is of type source IP. If source IP persistence is configured even with the least connections method, the requests go to a specific server. The server's IP address is required for maintaining the session information. Consider using HTTP Cookies based persistence.

- **Troubleshooting Tips**
  For other issues, consider the following tips to troubleshoot an issue not listed above:
  - If multiple load monitors are bound to a service, the load on the service is the sum of all the values on the load monitors bound to it. For load balancing to work properly, you must bind the same set of monitors to all the services.
  - If you disable a load monitor bound to the service and the service is bound to a virtual server, the virtual server uses the round robin method for load balancing.
  - When you bind a service to a virtual server where the load balancing method is CUSTOMLOAD and the service status is UP, the virtual server uses the initial round robin method for load balancing. It continues to be in round robin if the service has no custom load monitors, or if the status of at least one of the custom load monitors is not UP.
  - All the services that are bound to a virtual server where the load balancing method is CUSTOMLOAD, the services must have load monitors bound to them.
  - The CUSTOMLOAD load balancing method also follows the startup round robin.
  - If you disable a metric-based binding and this is the last active metric, the specific virtual server uses the round robin method for load balancing. A metric is disabled by setting the metric threshold to zero.
  - When a metric bound to a monitor crosses the threshold value, that particular service is not considered for load balancing. If all the services have reached the threshold, the virtual server uses the round robin method for load balancing and an error message "5xx - server busy error" is displayed.
  - A maximum of 10 metrics from a custom table can be bound to the monitor.
  - The OIDs must be scalar variables.
  - For successful load balancing, the interval must be as low as possible. If the interval is high, the time period for retrieving the load value increases. As a result, load balancing takes place using improper values.
  - A user cannot modify the local table.

Load balancing FAQs

September 14, 2021

What are the various load balancing policies I can create on the Citrix ADC appliance

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

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

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

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

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

What are the various devices that I can use to load balance with a Citrix ADC appliance?

You can load balance the following devices with a Citrix ADC appliance:

- Server farms
- Caches or Reverse Proxies
- Firewall devices
- Intrusion detection systems

- SSL offload devices
- Compression devices
- Content Inspection servers

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

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

- Reduce the response time: When you implement the load balancing feature for the website, one of the major benefits is the boost you can look forward to in load time. With two or more servers sharing the load of the web traffic, each of the servers runs less traffic load than a single server alone. This means there are more resources available to fulfill the client requests. This results in a faster website.
- Redundancy: Implementing the load balancing feature introduces a bit of redundancy. For example, if the website is balanced across three servers and one of them does not respond at all, the other two can keep running and the website visitors do not even notice any downtime. Any load balancing solution immediately stops sending traffic to the back-end server that is not available.

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

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

Networking

September 14, 2021

The following topics provide a conceptual reference and instructions for configuring the various networking components on the Citrix ADC appliance.

IP Addressing	Learn the various types of Citrix ADC owned IP addresses and how to create, customize, and remove them.
Interfaces	Configure some of the basic network configurations that must be done to get started.
Access Control Lists (ACLs)	Configure the different types of Access Control Lists and how to create, customize, and remove them.
IP Routing	Learn and configure the routing functionality of the Citrix ADC appliance, both static and dynamic.
Internet Protocol version 6 (IPv6)	Learn how the Citrix ADC appliance supports IPv6.
Traffic Domains	Learn and configure traffic domains to segment network traffic for different applications.
VXLAN	Learn and configure VXLANs to meet the scalability needs in your datacenter.

IP Addressing

September 14, 2021

Before you can configure the Citrix ADC appliance, you must assign the NSIP address, also known as the Management IP address. You can also create other Citrix ADC-owned IP addresses for abstracting servers and establishing connections with the servers. In this type of configuration, the appliance serves as a proxy for the abstracted servers. You can also proxy connections by using network address translations (INAT and RNAT). When proxying connections, the appliance can behave either as a bridging (Layer 2) device or as a packet forwarding (Layer 3) device. To make packet forwarding more efficient, you can configure static ARP entries. For IPv6, you can configure neighbor discovery (ND).

Configuring Citrix ADC-owned IP addresses

September 14, 2021

The Citrix ADC-owned IP addresses—NSIP address, Virtual IP Addresses (VIPs), Subnet IP Addresses (SNIPs), and Global Server Load Balancing Site IP Addresses (GSLBIPs)—exist only on the Citrix ADC appliance. The NSIP uniquely identifies the Citrix ADC on your network, and it provides access to the appliance. A VIP is a public IP address to which a client sends requests. The Citrix ADC terminates the client connection at the VIP and initiates a connection with a server. This new connection uses a SNIP or a MIP as the source IP address for packets forwarded to the server. If you have multiple data centers that are geographically distributed, each data center can be identified by a unique GSLBIP. You can configure some Citrix ADC-owned IP addresses to provide access for management applications.

Configuring the NSIP address

September 14, 2021

The NSIP address is the IP address at which you access the Citrix ADC appliance for management purposes. The appliance can have only one NSIP, which is also called the management IP address. You must add this IP address when you configure the Citrix ADC for the first time. You cannot remove an NSIP address. For security reasons, the NSIP should be a non-routable IP address on your organization's LAN.

If you modify this address, you must reboot the Citrix ADC appliance. If the subnet address of the new NSIP address is different from the previous one, you must add a default route for this subnet so that the new NSIP address becomes reachable from other networks on the LAN.

> **Important**
>
> Configuring the NSIP address is mandatory.

Changing the NSIP address of a Citrix ADC appliance consists of the following tasks:

- Change the NSIP address.
- Add a default route for the subnet address of the NSIP address, if one is not present.
- Save the configuration.
- Restart the appliance.

Command Line Procedures

To change the NSIP address by using the CLI:

At the command prompt, type:

- **set ns config -IPAddress** <ip_addr> **-netmask** <netmask>
- **show ns config**

To add a default route by using the CLI:

At the command prompt, type:

- **add route 0 0** <gateway IP address>
- **show route**

To save the configuration by using the CLI:

At the command prompt, type:

- **save config**

To restart the Citrix ADC appliance by using the CLI:

At the command prompt, type:

- **reboot**

GUI Procedures

To configure the NSIP address by using the GUI:

1. Click the gear icon in the top-right corner of the **Configuration** page.
2. Click the **NSIP address** pane.
3. On the **NSIP address** page, set the following parameters, and then click **Done**:
    - NSIP address
    - Netmask

To add a default route by using the GUI:

Navigate to **System** > **Network** > **Routes** and, on the **Basic** tab, add a default route with the following parameter settings, and then click **Create**.

- Network (set to zero)
- Netmask (set to zero)
- Gateway (IP address of the gateway)

To restart the Citrix ADC by using the GUI:

1. On the **System Information** tab page of the **System** node, click **Reboot**.
2. When prompted to reboot, select **Save Configuration** to make sure that you do not lose any configurations.

Sample configuration

In the following example, the NSIP address of a Citrix ADC appliance is changed to 192.0.2.90, which has a different subnet address (192.0.2.0/24) than the previous NSIP address. Therefore, a default route is added for this subnet, so that the new NSIP address becomes reachable from other networks.

```
> set nsconfig -ipAddress 192.0.2.90 -netmask 255.255.255.0

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> add route 0 0 192.0.2.1

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> save config

 Done
> reboot
```

Configuring and Managing Virtual IP (VIP) Addresses

September 14, 2021

Configuration of a virtual server IP (VIP) address is not mandatory during initial configuration of the Citrix ADC. When you configure load balancing, you assign VIP addresses to virtual servers.

For more information about configuring a load balancing setup, see Load Balancing.

In some situations, you need to customize VIP attributes or enable or disable a VIP address. A VIP address is usually associated with a virtual server, and some of the VIP attributes are customized to meet the requirements of the virtual server. You can host the same virtual server on multiple Citrix ADC appliances residing on the same broadcast domain, by using ARP and ICMP attributes. After you add a VIP (or any IP address), the appliance sends, and then responds to, ARP requests. VIPs are the only Citrix ADC-owned IP addresses that can be disabled. When a VIP address is disabled, the virtual server using it goes down and does not respond to ARP, ICMP, or L4 service requests. As an alternative to creating VIP addresses one at a time, you can specify a consecutive range of VIP addresses.

To create a VIP address by using the CLI:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.59 255.255.255.0 -type VIP
 Done
<!--NeedCopy-->
```

To create a range of VIP addresses by using the CLI:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
ip "10.102.29.60" added
ip "10.102.29.61" added
ip "10.102.29.62" added
ip "10.102.29.63" added
ip "10.102.29.64" added
 Done
<!--NeedCopy-->
```

To enable or disable an IPv4 VIP address by using the CLI:

At the command prompt, type one of the following sets of commands to enable or disable a VIP and verify the configuration:

- enable ns ip <IPAddress>
- show ns ip <IPAddress>
- disable ns ip <IPAddress>
- show ns ip <IPAddress>

Example:

```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Enabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
```



```
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Disabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled

 Done
<!--NeedCopy-->
```

To configure a VIP address by using the GUI:

Navigate to **System** > **Network** > **IPs** > **IPV4s**, and add a new IP address or edit an existing address.

To create a range of VIP addresses by using the GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. In the **Action** list, select **Add Range**.

To enable or disable a VIP address by using the GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Do one of the following:

- Select a VIP address.
- Hold down the **Ctrl** key and select multiple server address entries.
- Hold down the **Shift** key and select a range of server address entries.
- Select all the addresses by selecting the checkbox on the left side of the header row.

3. From the **Action** list, select **Disable** or **Enable**.

Detecting a Citrix ADC Appliance in a UDP Load Balancing Setup through TTL Updates

The following table displays how a Citrix ADC appliance handles the TTL value of received packets in different functionalities.

Functionality	TTL value
Virtual Server	TTL is set to 255 when forwarding the request to the backend servers. TTL is decremented by 1 when forwarding the response to the client.
L2 Mode	TTL is not changed.
L3 Mode	TTL is set to 255.
INAT	TTL is set to 255 when forwarding the request to the backend server. TTL is decremented by 1 when forwarding the response to the client.

Some enterprises/scenarios running a monitoring application requires the Citrix ADC appliance of a load balancing setup to be detected as one of the hop in a traceroute. A Citrix ADC appliance of a load balancing setup is not detected in a traceroute because the appliance, by default, sets the TTL value to 255 instead of decrementing it when forwarding the request to a backend server.

To meet this requirement, **Decrement TTL** parameter of a VIP address can be used. This parameter applies to all UDP virtual servers using this VIP.

When you enable the **Decrement TTL** parameter of a VIP, the Citrix ADC appliance decrements the TTL value by 1 instead of setting it to 255 when forwarding requests, which are received on the UDP virtual servers that uses this VIP.

Monitoring applications using traceroute data can now detect the presence of a Citrix ADC appliance of a UDP load balancing setup.

Before You Begin

Before you begin configuring a Citrix ADC appliance to be detected in a traceroute of a load balancing setup, note the following points:

- Decrement TTL parameter is supported only for UDP load balancing virtual servers.
- Decrement TTL parameter is supported for IPv4 VIP as well as IPv6 VIP (VIP6) addresses.
- Decrement TTL parameter is supported for standalone Citrix ADC appliances as well as for high availability (HA) and cluster setups.

Configuration Steps

Configuring a Citrix ADC appliance to be detected in a traceroute of a UDP load balancing setup consists of the following tasks:

- Create a UDP load balancing configuration
- Enable the Decrement TTL parameter for the VIP address

CLI Procedures

To enable the decrement TTL option for a VIP address by using the CLI:

- To enable the decrement TTL option for a VIP address while adding the VIP address, at the command prompt, type:
  - **add ns ip** <ip> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>
- To enable the decrement TTL option for an existing VIP address, at the command prompt, type:
  - **set ns ip** <ip> <mask> -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>

To enable the decrement TTL option for a VIP6 address by using the CLI:

- To enable the decrement TTL option for a VIP6 address while adding the VIP6 address, at the command prompt, type:
  - **add ns ip6** <IP6/prefix> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6/prefix>
- To enable the decrement TTL option for an existing VIP6 address, at the command prompt, type:
  - **set ns ip6** <ip6/prefix> <mask> -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6 address>

```
1     > add ns ip 203.0.113.30 -type VIP -decrementTTL ENABLED
2      Done
3
4     > add ns ip6  2001:DB8:5001::30 -type VIP -decrementTTL ENABLED
5      Done
6 <!--NeedCopy-->
```


GUI Procedures

To enable the decrement TTL option for a VIP address by using the GUI:

Navigate to **System** > **Network** > **IPs** > **IPv4s**, and enable the **Decrement TTL** parameter while adding a new VIP address or editing an existing address.

To enable the decrement TTL option for a VIP6 address by using the GUI:

Navigate to **System** > **Network** > **IPs** > **IPv6s**, and enable the **Decrement TTL** parameter while adding a new VIP6 address or editing an existing address.

Configuring ARP response Suppression for Virtual IP addresses (VIPs)

September 14, 2021

You can configure the Citrix ADC appliance to respond or not respond to ARP requests for a Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP.

For example, if virtual servers V1, of type HTTP, and V2, of type HTTPs, share VIP address 10.102.29.45 on a Citrix ADC appliance, you can configure the appliance to not respond to any ARP request for VIP 10.102.29.45 if both V1 and V2 are in the DOWN state.

The following three options are available for configuring ARP-response suppression for a virtual IP address.

- **NONE**. The Citrix ADC appliance responds to any ARP request for the VIP address, irrespective of the state of the virtual servers associated with the address.
- **ONE VSERVER**. The Citrix ADC appliance responds to any ARP request for the VIP address if at least one of the associated virtual servers is in UP state.
- **ALL VSERVER**. The Citrix ADC appliance responds to any ARP request for the VIP address if all of the associated virtual servers are in UP state.

Following table shows the sample behavior of Citrix ADC appliance for a VIP configured with two virtual servers:

Associated virtual servers for a VIP	STATE 1	STATE 2	STATE 3	STATE 4
NONE				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN

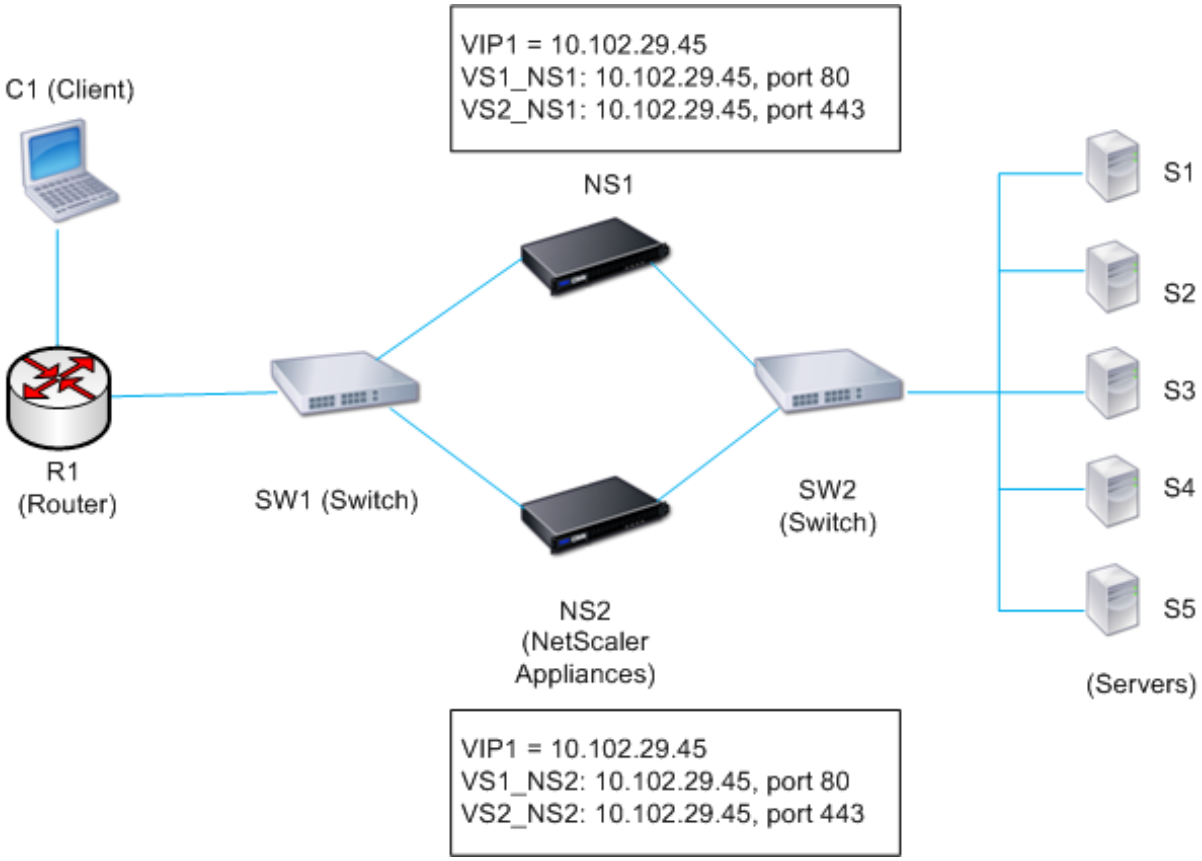
Associated virtual servers for a VIP	STATE 1	STATE 2	STATE 3	STATE 4
Respond to an ARP request for this VIP?	Yes	Yes	Yes	Yes
ONE VSERVER				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	Yes	Yes	No
ALL VSERVER				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	No	No	No

Consider an example where you want to test the performance of two virtual servers, V1 and V2, which have the same VIP address but are of different types and are each configured on Citrix ADC appliances NS1 and NS2. Let's call the shared VIP address *VIP1*.

V1 load balances servers S1, S2, and S3. V2 load balances servers S4 and S5.

On both NS1 and NS2, for VIP1, the ARP suppression parameter is set to ALL_VSERVER. If you want to test the performance of V1 and V2 on NS1, you must manually disable V1 and V2 on NS2, so that NS2 does not respond to any ARP request for VIP1.

Figure 1.



The execution flow is as follows:

1. Client C1 sends a request to V1. The request reaches R1.
2. R1 does not have an APR entry for the IP address (VIP1) of V1, so R1 broadcasts an ARP request for VIP1.
3. NS1 replies with source MAC address MAC1 and source IP address VIP1. NS2 does not reply to the ARP request.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table, and R1 updates the ARP entry with MAC1 and VIP1.
5. R1 forwards the packet to address VIP1 on NS1.
6. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP addresses and S2. When S2 sends a response to the client, the response returns by the same path.
7. Now you want to test the performance of V1 and V2 on NS2, so you enable V1 and V2 on NS2 and disable them on NS1. NS2 now broadcasts an ARP message for VIP1. In the message, MAC2 is the source MAC address and VIP1 is the source IP address.
8. SW1 learns the port number for reaching MAC2 from the ARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS2. R1 updates its ARP table.
9. Now suppose the ARP entry for VIP1 times out in the ARP table of R1, and client C1 sends a request for V1. Because R1 does not have an APR entry for VIP1, it broadcasts an ARP request

for VIP1.

10. NS2 replies with a source MAC address and VIP1 as the source IP address. NS1 does not reply to the ARP request.

To configure ARP response suppression by using the CLI:

At the command prompt, type:

- **set ns ip** -**arpResponse** <arpResponse>]
- **sh ns ip** <IPAddress>

Example:

```
> set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
 Done
<!--NeedCopy-->
```

To configure ARP response suppression by using the GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Open an IP address entry and select the type of ARP Response.

Configuring Subnet IP Addresses (SNIPs)

September 14, 2021

A subnet IP address (SNIP) is a Citrix ADC owned IP address that is used by the Citrix ADC to communicate with the servers.

The Citrix ADC uses the subnet IP address as a source IP address to proxy client connections to servers. It also uses the subnet IP address when generating its own packets, such as packets related to dynamic routing protocols, or to send monitor probes to check the health of the servers. Depending on your network topology, you might have to configure one or more SNIPs for different scenarios.

To configure a SNIP address on a Citrix ADC, you add the SNIP address and then enable global Use Subnet IP (USNIP) mode. As an alternative to creating SNIPs one at a time, you can specify a consecutive range of SNIPs.

To configure a SNIP address by using the CLI:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

Example:


```
> add ns ip 10.102.29.203 255.255.255.0 -type SNIP
 Done
<!--NeedCopy-->
```

To create a range of SNIP addresses by using the CLI:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done
<!--NeedCopy-->
```

To enable or disable USNIP mode by using the CLI:

At the command prompt, type one of the following commands:

- enable ns modeUSNIP
- disable ns modeUSNIP

To configure a SNIP address by using the GUI:

Navigate to System > Network > IPs > IPV4s, and add a new SNIP address or edit an existing address.

To create a range of SNIP addresses by using the GUI:

1. Navigate to System > Network > IPs > IPV4s.
2. In the Action list, select Add Range.

To enable or disable USNIP mode by using the CLI:

At the command prompt, type one of the following commands:

- enable ns mode USNIP
- disable ns mode USNIP

To enable or disable USNIP mode by using the GUI:

1. Navigate to System > Settings, in Modes and Features group, click Change modes.
2. Select or clear the Use Subnet IP option.

Using SNIPs for a Directly Connected Server Subnet

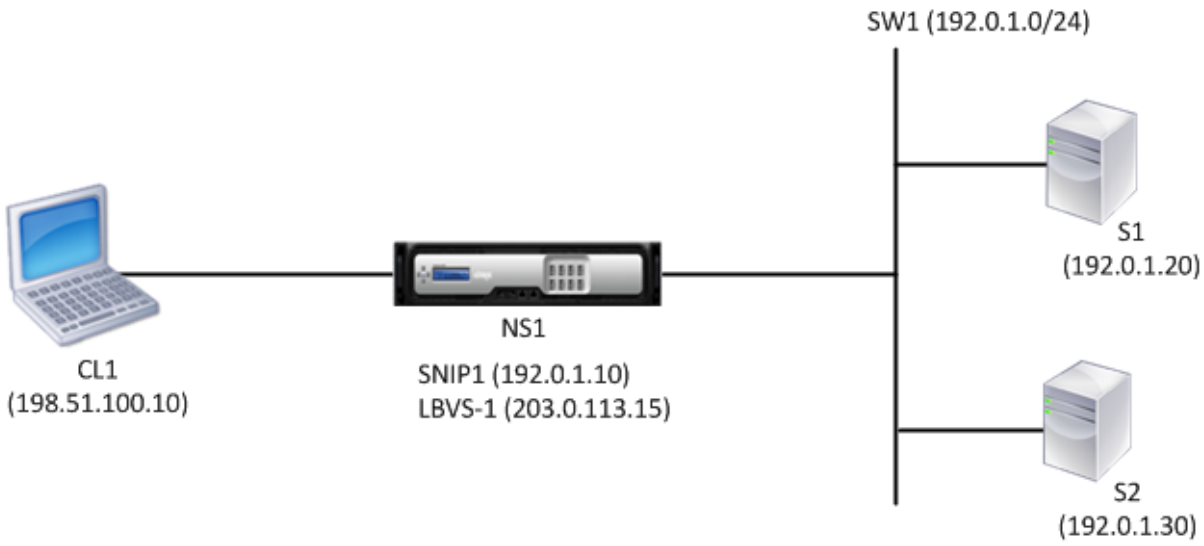
To enable communication between the Citrix ADC and a server that is either connected directly to the Citrix ADC or connected through only an L2 switch, you must configure a subnet IP address that belongs to the subnet of the server. You must configure at least one subnet IP address for each directly connected subnet, except for the directly connected management subnet that is connected through NSIP.

Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on Citrix ADC NS1 is used to load balance servers S1 and S2, which are connected to NS1 through L2 switch SW1. S1 and S2 belong to the same subnet.

SNIP address SNIP1, which belongs to the same subnet as S1 and S2, is configured on NS1. As soon as SNIP1 is configured, NS1 broadcasts ARP packets for SNIP1.

Services SVC-S1 and SVC-S2 on NS1 represent S1 and S2. As soon as these services are configured, NS1 broadcasts ARP requests for S1 and S2 to resolve IP-to-MAC mapping. After S1 and S2 respond, NS1 sends them monitoring probes at regular intervals, from address SNIP1, to check their health.

For more information about configuring load balancing on a Citrix ADC, see Load Balancing.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1's load balancing algorithm selects server S2.
4. Because S2 is directly connected to NS1, and SNIP1 (192.0.1.10) is the only IP address on NS1 that belongs to the same subnet as S2, NS1 opens a connection between SNIP1 and S2.
5. NS1 sends the request packet to S2 from SNIP1. The request packet has:

- Source IP = SNIP1 (192.0.1.10)
- Destination IP = IP address of S2 (192.0.1.30)

6. S2's response returns by the same path.

Using SNIPs for Server Subnets Connected through a Router

To enable communication between the Citrix ADC and servers in subnets connected through a router, you must configure at least one subnet IP address that belongs to the subnet of the directly connected interface to the router. The ADC uses this subnet IP address to communicate with servers in subnets that can be reached through the router.

Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on Citrix ADC NS1 is used to load balance servers S1, S2, S3, and S4, which are connected to NS1 through router R1.

S1 and S2 belong to same subnet, 192.0.2.0/24, and are connected to R1 through L2 switch SW1. S3 and S4 belong to a different subnet, 192.0.3.0/24, and are connected to R1 through L2 switch SW2.

Citrix ADC NS1 is connected to router R1 through subnet 192.0.1.0/24. SNIP address SNIP1, which belongs to the same subnet as the directly connected interface to the router (192.0.1.0/24), is configured on NS1. NS1 uses this address to communicate with servers S1 and S2, and with servers S3 and S4.

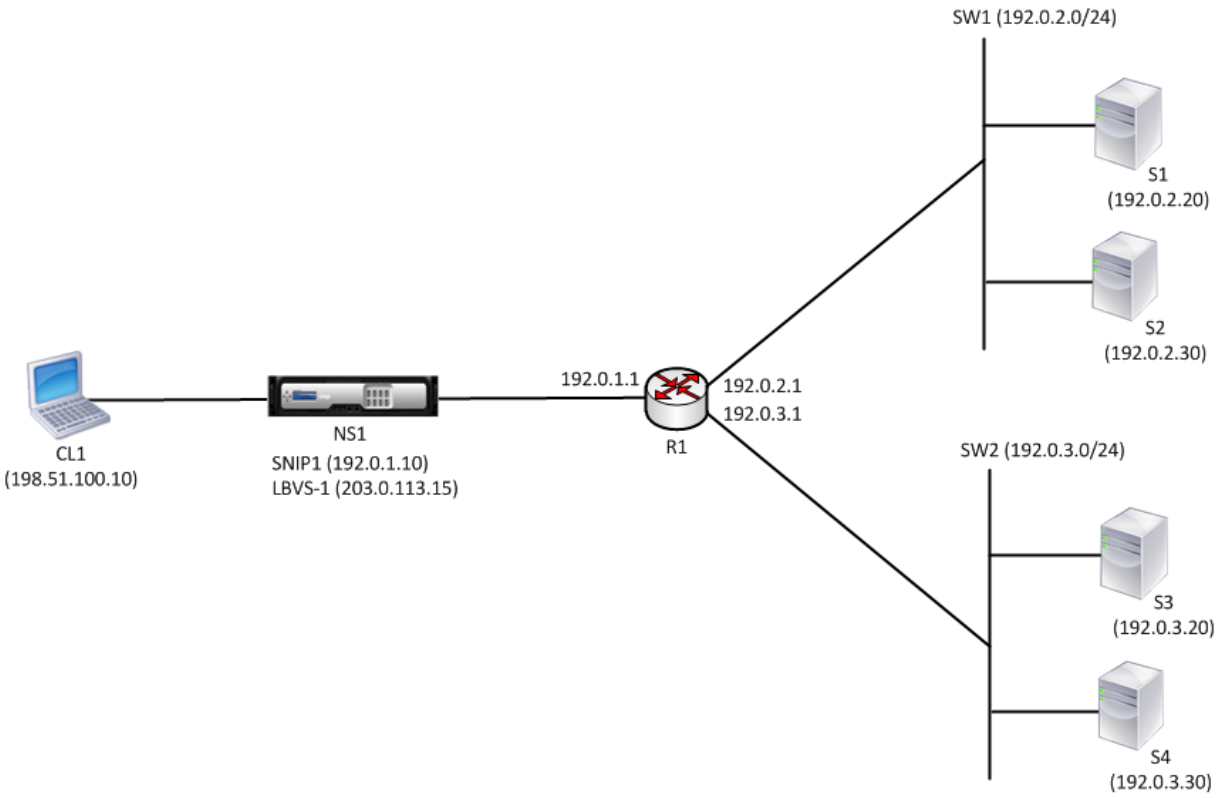
For more information about configuring load balancing on a Citrix ADC, see Load Balancing.

As soon as address SNIP1 is configured, NS1 broadcasts ARP announcement packets for SNIP1.

NS1's routing table consists of route entries for S1, S2, S3, and S4 through R1. These route entries are either static route entries or advertised by R1 to NS1, using dynamic routing protocols.

Services SVC-S1, SVC-S2, SVC-S3, and SVC-S4 on NS1 represent servers S1, S2, S3, and S4. NS1 finds, in its routing tables, that these servers are reachable through R1. NS1 sends them monitoring probes at regular intervals, from address SNIP1, to check their health.

For more information about IP routing on a Citrix ADC, see IP Routing.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1's load balancing algorithm selects server S3.
4. NS1 checks its routing table and finds that S3 is reachable through R1. SNIP1 (192.0.1.10) is the only IP address on NS1 that belongs to the same subnet as router R1, NS1 opens a connection between SNIP1 and S3 through R1.
5. NS1 sends the request packet to R1 from SNIP1. The request packet has:
   - Source IP address = SNIP1 (192.0.1.10)
   - Destination IP address = IP address of S3 (192.0.3.20)
6. The request reaches R1, which checks its routing table and forwards the request packet to S3.
7. S3's response returns by the same path.

Using SNIPs for Multiple Server Subnets (VLANs) on an L2 Switch

When you have multiple server subnets (VLANs) on an L2 switch that is connected to a Citrix ADC, you must configure at least one SNIP address for each of the server subnets, so that the Citrix ADC can communicate with these server subnets.

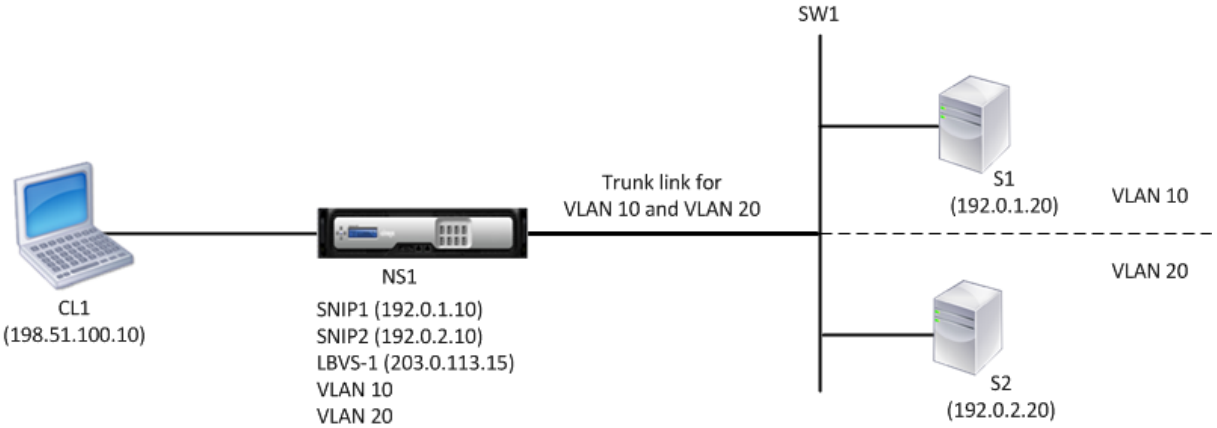
Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on Citrix ADC NS1 is used to load balance servers S1 and S2, which are connected to NS1 through L2 switch SW1. S1 and S2 belong to different subnets and are part of VLAN 10 and VLAN20, respectively. The link between NS1 and SW1 is a trunk link and is shared by VLAN10 and VLAN20.

For more information about configuring load balancing on a Citrix ADC, see Load Balancing.

Subnet IP addresses SNIP1 (for reference purposes only) and SNIP2 (for reference purposes only) are configured on NS1. NS1 uses SNIP1 (on VLAN 10) to communicate with server S1, and SNIP2 (on VLAN 20) to communicate with S2. As soon as SNIP1 and SNIP2 are configured, NS1 broadcasts ARP announcement packets for SNIP1 and SNIP2.

For more information about configuring VLANs on a Citrix ADC, see Configuring a VLAN.

Services SVC-S1 and SVC-S2 on NS1 represent servers S1 and S2. As soon as these services are configured, NS1 broadcasts ARP requests for them. After S1 and S2 respond, NS1 sends them monitoring probes at regular intervals to check their health. NS1 sends monitoring probes to S1 from address SNIP1, and to S2 from address SNIP2.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1's load balancing algorithm selects server S2.
4. Because S2 is directly connected to NS1, and SNIP2 (192.0.2.10) is the only IP address on NS1 that belongs to the same subnet as S2, NS1 opens a connection between SNIP2 and S2.
   Note: If S1 is selected, NS1 opens a connection between SNIP1 and S1.
5. NS1 sends the request packet to S2 from SNIP2. The request packet has:
   - Source IP = SNIP1 (192.0.2.10)
   - Destination IP = IP address of S2 (192.0.2.20)
6. S2's response returns by the same path.

Configuring GSLB Site IP Addresses (GSLBIP)

September 14, 2021

A GSLB site IP (GSLBIP) address is an IP address associated with a GSLB site. It is not mandatory to specify a GSLBIP address when you initially configure the Citrix ADC appliance. A GSLBIP address is used only when you create a GSLB site.

For more information about creating a GSLB site IP address, see Global Server Load Balancing.

Removing a Citrix ADC-owned IP address

September 14, 2021

You can remove any IP address except the NSIP. The following table provides information about the processes you must follow to remove the various types of IP addresses. Before removing a VIP, remove the associated virtual server.

IP address type	Implications
Subnet IP address (SNIP)	If IP address being removed is the last IP address in the subnet, the associated route is deleted from the route table. If the IP address being removed is the gateway in the corresponding route entry, the gateway for that subnet route is changed to another Citrix ADC-owned IP address.
Virtual Server IP address (VIP)	Before removing a VIP, you must first remove the virtual server associated with it. For information about removing the virtual server, see Load Balancing.
GSLB-Site-IP address	Before removing a GSLB site IP address, you must remove the site associated with it. For information about removing the site, see Global Server Load Balancing.

To remove an IP address by using the CLI:

At the command prompt, type:

rm ns ip <IPaddress>

Example:

```
> rm ns ip 10.102.29.54
Done
<!--NeedCopy-->
```

To remove an IP address by using the GUI:

Navigate to **System** > **Network** > **IPs** > **IPV4s**, delete the IP address.

Configuring Application Access Controls

September 14, 2021

Application access controls, also known as management access controls, form a unified mechanism for managing user authentication and implementing rules that determine user access to applications and data. You can configure SNIPs to provide access for management applications. Management access for the NSIP is enabled by default and cannot be disabled. You can, however, control it by using ACLs.

For information about using ACLs, see Access Control Lists (ACLs).

The Citrix ADC appliance does not support management access to VIPs.

The following table provides a summary of the interaction between management access and specific service settings for Telnet.

Management Access	Telnet (State Configured on the Citrix ADC)	Telnet (Effective State at the IP Level)
Enable	Enable	Enable
Enable	Disable	Disable
Disable	Enable	Disable
Disable	Disable	Disable

The following table provides an overview of the IP addresses used as source IP addresses in outbound traffic.

Application/ IP	NSIP	SNIP	VIP
ARP	Yes	Yes	No
Server side traffic	No	Yes	No
RNAT	No	Yes	Yes
ICMP PING	Yes	Yes	No
Dynamic routing	Yes	Yes	Yes

The following table provides an overview of the applications available on these IP addresses.

Application/ IP	NSIP	SNIP	VIP
SNMP	Yes	Yes	Yes
System access	Yes	Yes	No

You can access and manage the Citrix ADC by using applications such as Telnet, SSH, GUI, and FTP.

Note: Telnet and FTP are disabled on the Citrix ADC for security reasons. To enable them, contact the customer support. After the applications are enabled, you can apply the controls at the IP level.

To configure the Citrix ADC to respond to these applications, you need to enable the specific management applications. If you disable management access for an IP address, existing connections that use the IP address are not terminated, but no new connections can be initiated.

Also, the non-management applications running on the underlying FreeBSD operating system are open to protocol attacks, and these applications do not take advantage of the Citrix ADC appliance's attack prevention capabilities.

You can block access to these non-management applications on a SNIP or NSIP. When access is blocked, a user connecting to a Citrix ADC by using the SNIP or NSIP is not be able to access the non-management applications running on the underlying operating system.

To configure management access for an IP address by using the CLI:

At the command prompt, type:

set ns ip <IPAddress> -mgmtAccess <value> -**telnet** <value> -**ftp** <value> -**gui** <value> -**ssh** <value> -**snmp** <value> -**restrictAccess** (**ENABLED** | **DISABLED**)

Example:

```
1  > set ns ip 10.102.29.54 -mgmtAccess enabled -restrictAccess ENABLED
2   Done
```



```
<!--NeedCopy-->
```

To enable management access for an IP address by using the GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Open an IP address entry, and select the **Enable Management Access control** to support the listed applications option.

Enable secure access to Citrix ADC GUI using a subnet IP address(SNIP)

Secure access to Citrix ADC GUI is enabled by default for the Citrix ADC IP (NSIP). You can also enable secure access to the Citrix ADC appliance by using a subnet IP address of the appliance.

After configuring an SNIP address for secure access to a high availability pair, the secure access is available to the primary appliance, if you access the SNIP address.

Citrix ADC CLI procedure

To enable secure access to Citrix ADC GUI using a subnet IP address(SNIP) by using the CLI:

At the command prompt, type:

set ns ip <SNIP_Address> **-type SNIP -gui SECUREONLY -mgmtAccess ENABLED**

Example:

```
> set ns ip 203.0.113.99 -mgmtAccess enabled -restrictAccess ENABLED

 Done
<!--NeedCopy-->
```

How the Citrix ADC Proxies Connections

September 14, 2021

When a client initiates a connection, the Citrix ADC appliance terminates the client connection, initiates a connection to an appropriate server, and sends the packet to the server. The appliance does not perform this action for service type UDP or ANY.

For more information about service types, see Load Balancing.

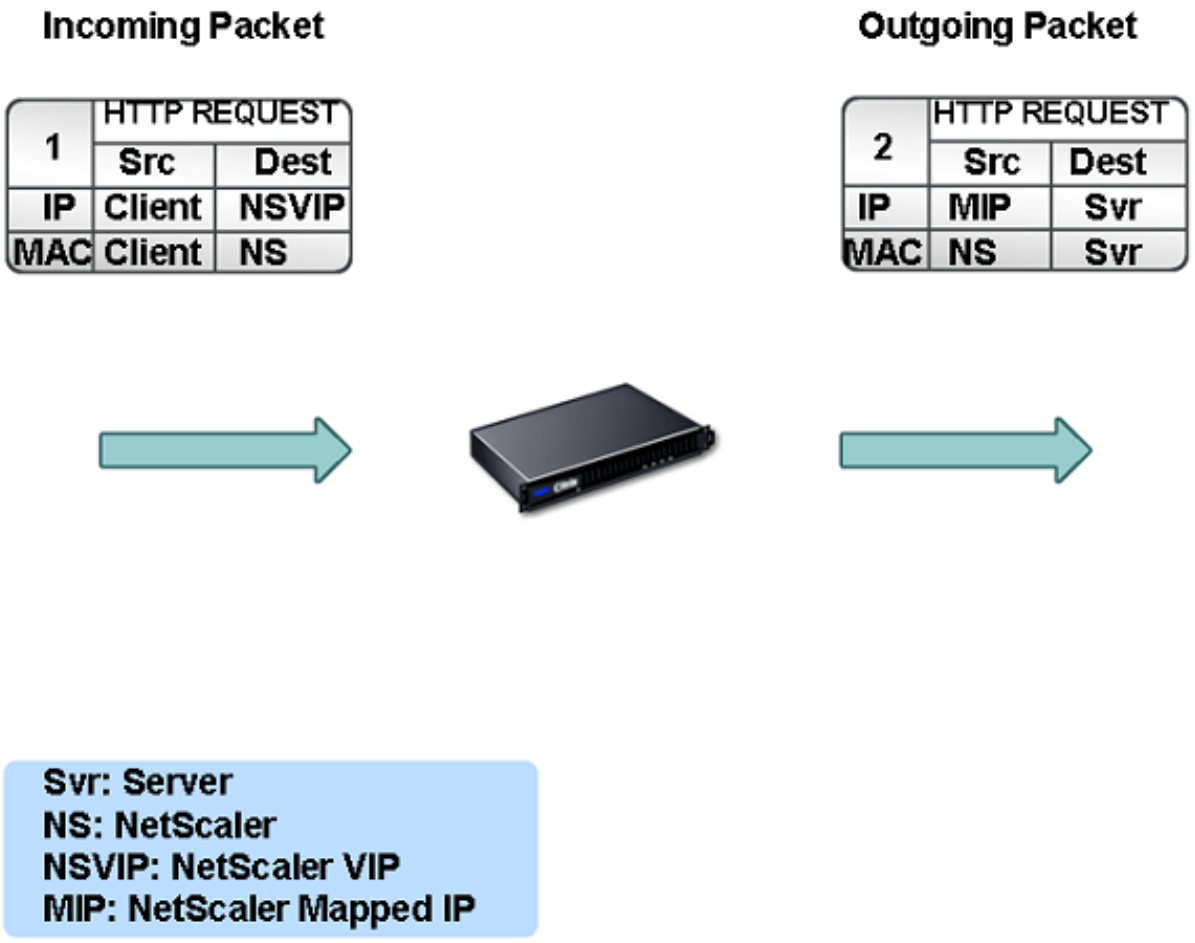
You can configure the Citrix ADC to process the packet before initiating the connection with a server. The default behavior is to change the source and destination IP addresses of a packet before sending

the packet to the server. You can configure the Citrix ADC to retain the source IP address of the packets by enabling Use Source IP mode.

How the Destination IP Address Is Selected

Traffic sent to the Citrix ADC appliance can be sent to a virtual server or to a service. The appliance handles traffic to virtual servers and services differently. The Citrix ADC terminates traffic received at a virtual server IP (VIP) address and changes the destination IP address to the IP address of the server before forwarding the traffic to the server, as shown in the following diagram.

Figure 1. Proxying Connections to VIPs



Packets destined for a service are sent directly to the appropriate server, and the Citrix ADC does not modify the destination IP addresses. In this case, the Citrix ADC functions as a proxy.

How the Source IP Address Is Selected

When the Citrix ADC appliance communicates with the physical servers or peer devices, by default, it does not use the IP address of the client. Citrix ADC maintains a pool of subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address of a connection to the physical server. Depending on the subnet in which the physical server is placed, Citrix ADC selects a specific SNIP address.

Note: If the Use Source IP (USIP) option is enabled, appliance uses the IP address of the client.

Enable Use Source IP Mode

September 14, 2021

When the Citrix ADC appliance communicates with the physical servers or peer devices, by default, it uses one of its own IP addresses as the source IP. The appliance maintains a pool of subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address for a connection to the physical server. The decision of selecting a SNIP address depends on the subnet in which the physical server resides.

If necessary, you can configure the Citrix ADC appliance to use the client's IP address as source IP. Some applications need the actual IP address of the client. The following use cases are a few examples:

- Client's IP address in the web access log is used for billing purposes or usage analysis.
- Client's IP address is used to determine the country of origin of the client or the originating ISP of the client. For example, many search engines such as Google provide content relevant to the location to which the user belongs.
- The application must know the client's IP address to verify that the request is from a trustworthy source.
- Sometimes, even though an application server does not need the client's IP address, a firewall placed between the application server and the Citrix ADC may need the client's IP address for filtering the traffic.

Enable Use Source IP mode (USIP) mode if you want the Citrix ADC to use the client's IP address for communication with the servers.

The following figure shows how the appliance uses IP addresses in USIP mode.

Before you begin

Before you enable USIP mode, note the following points:

- Enable USIP in the following situations:
  - Load balancing of Intrusion Detection System (IDS) servers
  - SMTP load balancing
  - Stateless connection failover
  - Sessionless load balancing
  - If you use the Direct Server Return (DSR) mode
- The USIP global setting applies only to services that are created after the USIP global setting is made. In other words, the USIP global setting does not apply to the existing services when the USIP global setting is made. For example, Disabling USIP globally does not disable USIP on the existing services. But it stops the subsequently created services from having USIP enabled automatically.

  To enable or disable USIP on a set of existing services, you need to enable or disable USIP on each of these services.
- When USIP is enabled, you must set server's gateway to one of the Citrix ADC owned IP addresses (of type Subnet IP (SNIP) so that server's response always go through the Citrix ADC appliance.
- If you enable USIP, set the idle timeout for server connections to a value lower than the default value, so that idle connections are cleared quickly on the server side.
- For transparent cache redirection, if you enable USIP, enable L2CONN also.
- Because HTTP connections are not reused when USIP is enabled, a large number of server-side connections may accumulate. Idle server connections can block connections for other clients. Therefore, set limits on maximum number of connections to a service. Citrix also recommends

setting the HTTP server time-out value, for a service on which USIP is enabled, to a value lower than the default, so that idle connections are cleared quickly on the server side.

- As an alternative to USIP mode, you have the option of inserting the client's IP address (CIP) in the request header of the server-side connection for an application server that needs the client's IP address.
- In earlier Citrix ADC releases, USIP mode had the following source-port options for server-side connections:
  - **Use the client's port**. With this option, connections cannot be reused. For every request from the client, a new connection is made with the physical server.
  - **Use proxy port**. With this option, connection reuse is possible for all requests from the same client.

  In the later Citrix ADC releases, if USIP is enabled, the default is to use a proxy port for server-side connections and not reuse connections. Not reusing connections may not affect the speed of establishing connections.

  By default, the Use Proxy Port option is enabled if the USIP mode is enabled.

  Note: If you enable the USIP mode, it is recommended to enable the Use Proxy Port option.

  For more information about the Use Proxy Port option, see Configure the source port for server-side connections.

Configuration Steps

Enable Use Source IP mode (USIP) mode if you want Citrix ADC to use the client's IP address for communication with the servers. By default, USIP mode is disabled. USIP mode can be enabled globally on the Citrix ADC or on a specific service. If you enable it globally, USIP is enabled by default for all subsequently created services. If you enable USIP for a specific service, the client's IP address is used only for the traffic directed to that service.

CLI procedures

To globally enable or disable USIP mode by using the CLI:

At the command prompt, type one of the following commands:

- **enable ns mode USIP**
- **disable ns mode USIP**

To enable USIP mode for a service by using the CLI:

At the command prompt, type:

set service <name>@ -**usip** (**YES** | **NO**)

Example:

```
> set service Service-HTTP-1 -usip YES
Done
<!--NeedCopy-->
```

GUI procedures

To globally enable or disable USIP mode by using the GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features** group, click **Change Modes**.
2. Select or clear the **Use Source IP** option.

To enable USIP mode for a service by using the GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and edit a service.
2. In **Advanced Settings**, select **Service Settings**, and select **Use Source IP Address**.

Configuring Network Address Translation

September 14, 2021

Network address translation (NAT) involves modification of the source and/or destination IP addresses and/or the TCP/UDP port numbers of IP packets that pass through the Citrix ADC appliance. Enabling NAT on the appliance enhances the security of your private network, and protects it from a public network such as the Internet, by modifying your networks source IP addresses when data passes through the Citrix ADC. Also, with the help of NAT entries, your entire private network can be represented by a few shared public IP addresses. The Citrix ADC supports the following types of network address translation:

- **Inbound NAT (INAT).** The Citrix ADC replaces the destination IP address in the packets generated by the client with the private IP address of the server.
- **Reverse NAT (RNAT).** The Citrix ADC replaces the source IP address in the packets generated by the servers with the public NAT IP addresses.

Inbound Network Address Translation

September 14, 2021

When a client sends a packet to a Citrix ADC appliance that is configured for Inbound Network Address Translation (INAT), the appliance translates the packet's public destination IP address to a private destination IP address and forwards the packet to the server at that address.

The following configurations are supported:

- **IPv4-IPv4 Mapping**: A public IPv4 address on the Citrix ADC appliance listens to connection requests on behalf of a private IPv4 server. The Citrix ADC appliance translates the packet's public destination IP address to the destination IP address of the server. Then the appliance forwards the packet to the server at that address.
- **IPv4-IPv6 Mapping**: A public IPv4 address on the Citrix ADC appliance listens to connection requests on behalf of a private IPv6 server. The Citrix ADC appliance creates an IPv6 request packet with the IP address of the IPv6 server as the destination IP address.
- **IPv6-IPv4 Mapping**: A public IPv6 address on the Citrix ADC appliance listens to connection requests on behalf of a private IPv4 server. The Citrix ADC appliance creates an IPv4 request packet with the IP address of the IPv4 server as the destination IP address.
- **IPv6-IPv6 Mapping**: A public IPv6 address on the Citrix ADC appliance listens to connection requests on behalf of a private IPv6 server. The Citrix ADC appliance translates the packet's public destination IP address to the destination IP address of the server. Then the appliance forwards the packet to the server at that address.

When the appliance forwards a packet to a server, the source IP address assigned to the packet is determined as follows:

- If use subnet IP (USNIP) mode is enabled and use source IP (USIP) mode is disabled, the appliance uses a subnet IP address (SNIP) as the source IP address.
- If USIP mode is enabled, and USNIP mode is disabled the appliance uses the client IP (CIP) address as the source IP address.
- If both USIP and USNIP modes are enabled, USIP mode takes precedence.
- You can also configure the Citrix ADC to use a unique IP address as the source IP address, by setting the proxyIP parameter.
- If none of the above modes are enabled and a unique IP address has not been specified, the Citrix ADC attempts to use a MIP as the source IP address.
- If both USIP and USNIP modes are enabled and a unique IP address has been specified, the order of precedence is as follows: USIP-unique IP-USNIP-MIP-Error.

To protect the Citrix ADC from DoS attacks, you can enable TCP proxy. However, if other protection mechanisms are used in your network, you can disable them.

Configure INAT rules

You can create, modify, or remove an INAT entry.

CLI procedures

To create an INAT entry by using the CLI:

At the command prompt, type the following commands to create an INAT entry and verify its configuration:

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <**ip_addr > ipv6_addr>**]
- **show inat** [<name>]

Example:

```
> add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
 Done
<!--NeedCopy-->
```

To modify an INAT entry by using the CLI:

To modify an INAT entry, type the `set inat` command, the name of the entry, and the parameters to be changed, with their new values.

To remove an INAT configuration by using the CLI:

At the command prompt, type:

- **rm inat** <name>

Example:

```
> rm inat ip4-ip4
 Done
<!--NeedCopy-->
```

GUI procedures

To configure an INAT entry by using the GUI:

Navigate to **System** > **Network** > **Routes** > **INAT**, and add an INAT entry or edit an existing INAT entry.

To remove an INAT configuration by using the GUI:

Navigate to **System** > **Network** > **Routes** > **INAT**, delete the INAT configuration.

Connection failover for INAT rules

Connection failover or connection mirroring enables the primary node to duplicate connection and persistence information to the secondary node in a high availability. The state information of the connection is shared with the secondary node regularly when connection mirroring is enabled.

Enabling connection failover provides more reliability but it comes at the cost of some system time being used up for sharing the state information. The connection data is synchronized to the standby unit with every packet or flow state update. Hence, it must be used only at places where connection level reliability is of prime importance.

Citrix ADC appliance high availability setups support connection failover for INAT connections. The primary node sends INAT mappings and other INAT related connection information to the secondary node at regular intervals. The secondary appliance uses the mapping and connection information only in the event of a failover.

When a failover occurs, the new primary node has information about the INAT connections established before the failover. Hence, it continues to serve those connections even after the failover.

From the client's perspective the failover is transparent. During the transition period, the client and server might experience a brief disruption and retransmissions. Connection failover can be enabled per INAT rule.

For enabling connection failover on an INAT rule, you enable the `connFailover` parameter of that specific RNAT rule by using CLI.

CLI procedure

To enable connection failover for an INAT rule by using the CLI:

To enable connection failover while adding an INAT rule, at the command prompt, type:

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <ip_addr|ipv6_addr>] -**connfailover** (**ENABLED** | **DISABLED**)
- **show inat** <name>

To enable connection failover while modifying an existing INAT rule, at the command prompt, type:

set inat -connfailover (**ENABLED** **DISABLED**)

-
- **show inat** <name>

Coexistence of INAT and Virtual Servers

September 14, 2021

If both INAT and RNAT are configured, the INAT rule takes precedence over the RNAT rule. If RNAT is configured with a network address translation IP (NAT IP) address, the NAT IP address is selected as the source IP address for that RNAT client.

The default public destination IP in an INAT configuration is the virtual IP (VIP) address of the Citrix ADC device. Virtual servers also use VIPs. When both INAT and a virtual server use the same IP address, the Vserver configuration overrides the INAT configuration.

Following are a few sample configuration setup scenarios and their effects.

Case	Result
You have configured a virtual server and a service to send all data packets received on a specific Citrix ADC port to the server directly. You have also configured INAT and enabled TCP. Configuring INAT in this manner sends all data packets received through a TCP engine before sending them to the server.	All packets received on the Citrix ADC, except those received on the specified port, pass through the TCP engine.
You have configured a virtual server and a service to send all data packets of service type TCP, that are received on a specific port on the Citrix ADC, to the server after passing through the TCP engine. You have also configured INAT and disabled TCP. Configuring INAT in this manner sends the data packets received directly to the server.	Only packets received on the specified port pass through the TCP engine.
You have configured a virtual server and a service to send all data packets received to either of two servers. You are attempting to configure INAT to send all data packets received to a different server.	The INAT configuration is not allowed.
You have configured INAT to send all received data packets directly to a server. You are attempting to configure a virtual server and a service to send all data packets received to two different servers.	The vserver configuration is not allowed.

Stateless NAT46

September 14, 2021

The stateless NAT46 feature enables communication between IPv4 and IPv6 networks through IPv4 to IPv6 packet translation, and vice versa, without maintaining any session information on the Citrix ADC appliance.

For a stateless NAT46 configuration, the appliance translates an IPv4 packet to IPv6 or an IPv6 packet to IPv4 as defined in RFCs 6145 and 2765.

A stateless NAT46 configuration on the Citrix ADC appliance has the following components:

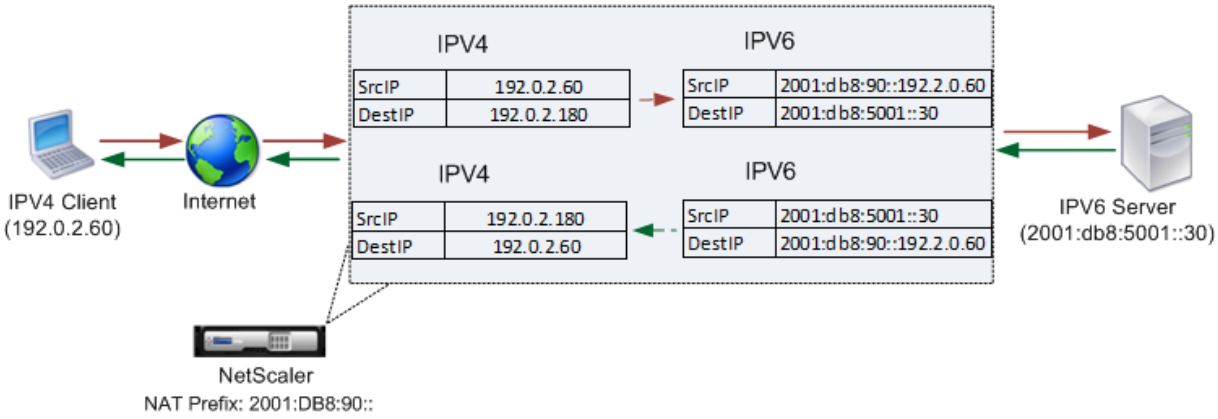
- **IPv4-IPv6 INAT entry.** An INAT entry defining a 1:1 relationship between an IPv4 address and an IPv6 address. In other words, an IPv4 address on the appliance listens to connection requests on behalf of an IPv6 server. An IPv4 request packet for this IPv4 address is translated into an IPv6 packet, and then the IPv6 packet is sent to the IPv6 server.

  The appliance translates an IPv6 response packet into an IPv4 response packet with its source IP address field set as the IPv4 address specified in the INAT entry. The translated packet is then sent to the client.

- **NAT46 IPv6 prefix.** A global IPv6 prefix of length 96 bits (128-32=96) configured on the appliance. During IPv4 packet to IPv6 packet translation, the appliance sets the source IP address of the translated IPv6 packet to a concatenation of the NAT46 IPv6 prefix [96 bits] and the IPv4 source address [32 bits] that was received in the request packet.

  During IPv6 packet to IPv4 packet translation, the appliance sets the destination IP address of the translated IPv4 packet to the last 32 bits of the destination IP address of the IPv6 packet.

Consider an example in which an enterprise hosts site www.example.com on server S1, which has an IPv6 address. To enable communication between IPv4 clients and IPv6 server S1, Citrix ADC appliance NS1 is deployed with a stateless NAT46 configuration that includes an IPv4-IPv6 INAT entry for server S1, and a NAT46 Prefix. The INAT entry includes an IPv4 address at which the appliance listens to connection requests from IPv4 clients on behalf of the IPv6 server S1.



The following table lists the settings used in this example:

Entities	Name	Value
IP address of the client	Client_IPv4 (for reference purposes only)	192.0.2.60
IPv6 address of the server	Sevr_IPv6 (for reference purposes only)	2001:DB8:5001::30
IPv4 address defined in the INAT entry for IPv6 server S1	Map-Sevr-IPv4 (for reference purposes only)	192.0.2.180
IPv6 prefix for NAT 46 translation	NAT46_Prefix (for reference purposes only)	2001:DB8:90::

Following is the traffic flow in this example:

1. IPv4 Client CL1 sends a request packet to the Map-Sevr-IPv4 (192.0.2.180) address on the Citrix ADC appliance.
2. The appliance receives the request packet and searches the NAT46 INAT entries for the IPv6 address mapped to the Map-sevr-IPv4 (192.0.2.180) address. It finds the Sevr-IPv6 (2001:DB8:5001::30) address.
3. The appliance creates a translated IPv6 request packet with:
   - Destination IP address field = Sevr-IPv6 = 2001:DB8:5001::30
   - Source IP address field = Concatenation of NAT Prefix (First 96 bits) and Client_IPv4 (last 32 bits) = 2001:DB8:90::192.0.2.60
4. The appliance sends the translated IPv6 request to Sevr-IPv6.
5. The IPv6 server S1 responds by sending an IPv6 packet to the Citrix ADC appliance with:
   - Destination IP address field = Concatenation of NAT Prefix (First 96 bits) and Client_IPv4 (last 32 bits)= 2001:DB8:90::192.0.2.60
   - Source IP address field = Sevr-IPv6 = 2001:DB8:5001::30

6. The appliance receives the IPv6 response packet and verifies that its destination IP address matches the NAT46 prefix configured on the appliance. Because the destination address matches the NAT46 prefix, the appliance searches the NAT46 INAT entries for the IPv4 address associated with the Sevr-IPv6 address (2001:DB8:5001::30). It finds the Map-Sevr-IPv4 address (192.0.2.180).
7. The appliance creates an IPv4 response packet with:
   - Destination IP address field = The NAT46 prefix stripped from the destination address of the IPv6 response = Client_IPv4 (192.0.2.60)
   - Source IP address field = Map-Sevr-IPv4 address (192.0.2.180)
8. The appliance sends the translated IPv4 response to client CL1.

Limitations of Stateless NAT46

The following limitations apply to stateless NAT46:

- Translation of IPv4 options is not supported.
- Translation of IPv6 routing headers is not supported.
- Translation of hop-by-hop extension headers of IPv6 packets is not supported.
- Translation of ESP and EH headers of IPv4 packets is not supported.
- Translation of multicast packets is not supported.
- Translation of destination option headers and source routing headers is not supported.
- Translation of fragmented IPv4 UDP packets that do not contain UDP checksum is not supported.

Configure Stateless NAT46

Creating the required entities for stateless NAT46 configuration on the Citrix ADC appliance involves the following procedures:

1. Create an IPv4-IPv6 mapping INAT entry with stateless mode enabled.
2. Create a NAT46 IPv6 prefix.

CLI procedures

To configure an INAT mapping entry by using the CLI:

At the command prompt, type:

- add inat <name> <publicIPv4> <privateIPv6> -mode STATELESS
- show inat <name>

To create an NAT46 prefix by using the CLI:

At the command prompt, type:


```
set inatparam -nat46v6Prefix <ipv6_addr            *>
```

-
- show inatparam

Example:

```
 > add inat exmpl-com-stls-nat46 192.0.2.180
2001:DB8:5001::30 -mode stateless
Done

> set inatparam -nat46v6Prefix 2001:DB8:90::/96
Done
<!--NeedCopy-->
```

GUI procedures

To create an INAT mapping entry by using the GUI:

1. Navigate to System > Network > Routes > INAT.
2. Add a new INAT entry, or edit an existing INAT entry.
3. Set the following parameters:
   - Name*
   - Public IP Address*
   - Private IP Address* (Select the IPv6 check box and enter the address in IPv6 format.)
   - Mode (Select Stateless from the drop down list.)

   * A required parameter

To create a NAT46 prefix by using the GUI:

Navigate to **System** > **Network**, in the **Settings** group, click **Configure INAT Parameters**, and set the **Prefix** parameter.

Setting Global Parameters for Stateless NAT46

The appliance provides some optional global parameters for stateless NAT46 configurations.

To set global parameters for stateless NAT46 by using the CLI:

At the command prompt, type:

set inatparam [-**nat46IgnoreTOS** (**YES**	**NO**)] [-**nat46ZeroCheckSum** (**ENABLED**	**DISABLED**)] [-**nat46v6Mtu** <positive_integer>] [-**nat46FragHeader** (**ENABLED**	**DISABLED**)]

-
- **show inatparam**

Example:

```
> set inatparam -nat46IgnoreTOS YES -nat46ZeroCheckSum DISABLED -
    nat46v6Mtu 1400 -nat46FragHeader DISABLED
 Done
<!--NeedCopy-->
```

To set global parameters for stateless NAT46 by using the GUI:

Navigate to **System** > **Network**, in the **Settings** group, click **Configure INAT Parameters**.

DNS64

September 14, 2021

The Citrix ADC DNS64 feature responds with a synthesized DNS AAAA record to an IPv6 client sending an AAAA request for an IPv4-only domain. The DNS64 feature is used with the NAT64 feature to enable seamless communication between IPv6-only clients and IPv4-only servers. DNS64 enables discovery of the IPv4domain by the IPV6 only clients, and NAT64 enables communication between the clients and servers.

For synthesizing an AAAA record, the Citrix ADC appliance fetches a DNS A record from a DNS server. The DNS64 prefix is a 96-bit IPv6 prefix configured on the Citrix ADC appliance. The Citrix ADC appliance synthesizes the AAAA record by concatenation of the DNS64 Prefix (96 bits) and the IPv4 address (32 bits).

For enabling communication between IPv6 clients and IPv4 servers, a Citrix ADC appliance with DNS64 and NAT64 configuration can deployed either on the IPv6 client side or on the IPv4 server side. In both cases, the DNS64 configuration on the Citrix ADC appliance is similar and includes a load balancing virtual server acting as a proxy server for DNS servers. If the Citrix ADC appliance is deployed on the client side, the load balancing virtual server must be specified, on the IPv6 client, as the nameserver for a domain.

Consider an example where a Citrix ADC appliance with DNS64 and NAT64 configuration is configured on the IPv4 side. In this example, an enterprise hosts site www.example.com on server S1, which has an IPv4 address. To enable communication between IPv6 clients and IPv4 server S1, Citrix ADC appliance NS1 is deployed with a DNS64 and stateful NAT64 configuration.

The DNS64 configuration includes DNS load balancing virtual server LBVS-DNS64-1, on which the DNS64 option is enabled. A DNS64 policy named DNS64-Policy-1, and an associated DNS64 action named DNS64-Action-1, are also configured on NS1, andDNS64-Policy-1 is bound to LBVS-DNS64-1. LBVS-DNS64-1 acts as a DNS proxy server for DNS servers DNS-1 and DNS-2.

When traffic arriving at LBVS-DNS64-1 matches the conditions specified in DNS64-Policy-1, the traffic is processed according to the settings in DNS64-Action-1. DNS64-Action-1 specifies the DNS64 prefix used, with the A record received from a DNS server, to synthesize an AAAA record.

The global DNS parameter cacherecords is enabled on the Citrix ADC appliance, so the appliance caches DNS records. This setting is necessary for the DNS64 to work properly.

The following table lists the settings used in the above example: DNS64 example settings.

Following is the traffic flow in this example:

1. IPv6 client CL1 sends a DNS AAAA request for the IPv6 address of the site www.example.com.
2. The request is received by the DNS load balancing virtual server LBVS-DNS64-1 on Citrix ADC appliance NS1.
3. NS1 checks its DNS cache records for the requested AAAA record and finds that AAAA record for the site www.example.com does not exist in the DNS cache.
4. LBVS-DNS64-1's load balancing algorithm selects DNS server DNS-1 and forwards the AAAA request to it.
5. Because the site www.example.com is hosted on an IPv4 server, the DNS server DNS-1 does not have any AAAA record for the site www.example.com.
6. DNS-1 sends either an empty DNS AAAA response or an error message to LBVS-DNS64-1.
7. Because DNS64 option is enabled on LBVS-DNS64-1 and the AAAA request from CL1 matches the condition specified in DNS64-Policy-1, NS1 sends a DNS A request to DNS-1 for the IPv4 address of www.example.com.
8. DNS-1 responds by sending the DNS A record for www.example.com to LBVS-DNS64-1. The A record includes the IPv4 address for www.example.com.
9. NS1 synthesizes an AAAA record for the site www.example.com with:
    - IPv6 address for site www.example.com = Concatenation of DNS64 Prefix (96 bits) specified in the associated DNS64action, and IPv4 address of DNS A record (32 bits) = 2001:DB8:300::192.0.2.60
10. NS1 sends the synthesized AAAA record to IPv6 client CL1. NS1 also caches the A record into its memory. NS1 uses the cached A record to synthesize AAAA records for subsequent AAAA requests.

Points to Consider for a DNS64 Configuration

Before configuring DNS64 on a Citrix ADC appliance, consider the following points:

- The DNS64 feature of the Citrix ADC appliance is compliant with RFC 6174.
- The DNS64 feature of the Citrix ADC appliance does not support DNSSEC. The Citrix ADC appliance does not synthesize an AAAA record from a DNSSEC response received from a DNS server. A response is classified as a DNSSEC response, only if it contains RRSIG records.
- The Citrix ADC appliance supports DNS64 prefix of length of only 96 bits.
- Though the DNS64 feature is used with the NAT64 feature, the DNS64 and NAT64 configurations are independent on the Citrix ADC appliance. For a particular flow, you must specify the same IPv6 prefix value for the DNS64 prefix and the NAT64 prefix parameters, so that the synthesized IPv6 addresses received by the client are routed to the particular NAT64 configuration. For more information on configuring NAT64 on a Citrix ADC appliance, see Stateful NAT64.
- The following are the different cases of DN64 processing by the Citrix ADC appliance:
  - If the AAAA response from the DNS server includes AAAA records, then each record in the response is checked for the set of exclusion rule configured on the Citrix ADC appliance for the particular DNS64 configuration. The Citrix ADC removes the IPv6 addresses, whose prefix matches the exclusion rule, from the response. If the resulting response includes at least one IPv6 record, the Citrix ADC appliance forwards this response to the client, else, the appliance synthesizes a AAAA response from the A record of the domain and sends it to the IPv6 client.
  - If the AAAA response from the DNS server is an empty answer response, the appliance requests for A resource records with the same domain name or searches in its own records if the appliance is an authentic domain name server for the domain. If the request results in an empty answer or error, the same is forwarded to the client.
  - If the response from the DNS server includes RCODE=1 (format error), the Citrix ADC appliance forwards the same to the client. If there is no response before the timeout, the Citrix ADC appliance sends a response with RCODE=2 (server failure) to the client.
  - If the response from the DNS server includes a CNAME, the chain is followed until the terminating A or AAAA record is reached. If the CNAME does not have any AAAA resource records, the Citrix ADC appliance fetches the DNS A record to be used for synthesizing AAAA record. The CNAME chain is added to the answer section along with the synthesized AAAA record and then sent to the client.
- The DNS64 feature of the Citrix ADC appliance also supports responding to PTR request. When a PTR request for a domain of an IPv6 address is received on the appliance and the IPv6 address matches any of the configured DNS64 prefix, the appliance creates a CNAME record mapping

the IP6-ARPA domain into the corresponding IN-ADDR. ARPA domain and the newly formed IN-ADDR.ARPA domain is used for resolution. The appliance searches the local PTR records and if the records are not present, the appliance sends a PTR request for IN-ADDR.ARPA domain to the DNS server. The Citrix ADC appliance uses the response from the DNS server to synthesize response for the initial PTR request.

Configuration Steps

Creating the required entities for stateful NAT64 configuration on the Citrix ADC appliance involves the following procedures:

- **Add DNS services.** DNS services are logical representation of DNS servers for which the Citrix ADC appliance acts as a DNS proxy server. For more information on setting optional parameters of a service, see Load Balancing.
- **Add DNS64 action and DNS64 policy and then bind the DNS64 action to the DNS64 policy.** A DNS64 policy specifies conditions to be matched against traffic for DNS64 processing according to the settings in the associated DNS64 action. The DNS64 action specifies the mandatory DNS64 prefix and the optional exclude rule and mapped rule settings.
- **Create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it.** The DNS load balancing virtual server acts as a DNS proxy server for DNS servers represented by the bound DNS services. Traffic arriving at the virtual server is matched against the bound DNS64 policy for DNS64 processing. For more information on setting optional parameters of a load balancing virtual server, see Load Balancing.

  Note: The CLI has separate commands for these two tasks, but the GUI combines them in a single dialog box.

  Enable caching of DNS records. Enable the global parameter for the Citrix ADC appliance to cache DNS records, which are obtained through DNS proxy operations. For more information on enabling caching of DNS records, see Domain Name System.

CLI procedures

To create a service of type DNS by using the CLI:

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> …

To create a DNS64 action by using the CLI:

At the command prompt, type:

- add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <expression>] [-excludeRule <expression>]

To create a DNS64 policy by using the CLI:

At the command prompt, type:

- add dns policy64 <name> -rule <expression> -action <string>

To create a DNS load balancing virtual server by using the CLI:

At the command prompt, type:

- add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED) [-bypassAAAA (YES | NO)] …

To bind the DNS services and the DNS64 policy to the DNS load balancing virtual server by using the CLI:

At the command prompt, type:

- bind lb vserver <name> <serviceName> …
- bind lb vserver <name> -policyName <string> -priority <positive_integer> …

GUI procedures

To create a service of type DNS by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Services, and add a new service.
2. Set the following parameters:
    - Service Name*
    - Server*
    - Protocol* (Select DNS from the drop down list.)
    - Port*

To create a DNS64 action by using the GUI:

Navigate to Traffic Management > DNS > Actions, on the DNS Actions64 tab, add a new DNS64 action.

To create a DNS64 policy by using the GUI:

Navigate to Traffic Management > DNS > Policies, on the DNS Policies64 tab, add a new DNS64 policy.

To create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a new virtual server.
2. Set the following parameters:
    - Name*
    - IP Address*
    - Protocol* (Select DNS from the drop down list.)
    - Port*

3. Select the Enable DNS64 option.
4. In the Services pane, bind the service to the virtual server.
5. In the Policies pane, bind the policy to the virtual server.

Sample Configuration

```
 > add service SVC-DNS-1 203.0.113.50 DNS 53
 Done

> add service SVC-DNS-2 203.0.113.60 DNS 53
 Done

> add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
 Done

> add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET
   (2001:DB8:5001::/64)"
-action DNS64-Action-1
 Done

> add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-1
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-2
 Done

> bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
 Done

<!--NeedCopy-->
```

Stateful NAT64 Translation

September 14, 2021

The stateful NAT64 feature enables communication between IPv6 clients and IPv4 servers through IPv6 to IPv4 packet translation, and vice versa, while maintaining session information on the Citrix ADC appliance.

A stateful NAT64 configuration on the Citrix ADC appliance has the following components:

- **NAT64 rule**— An entry consisting of an ACL6 rule and a netprofile, which consists of a pool of Citrix ADC owned SNIP Addresses.
- **NAT64 IPv6 Prefix**— A global IPv6 prefix of length 96 bits (128-32=96) configured on the appliance.
  Note: Currently the Citrix ADC appliance supports only one prefix to be used commonly with all NAT 64 rules.

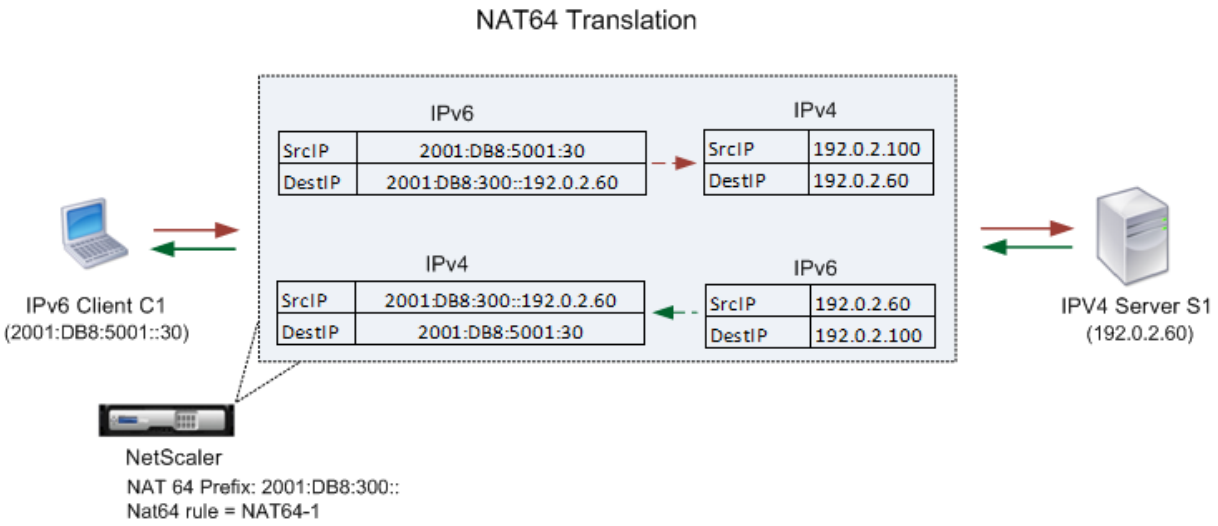
The Citrix ADC appliance considers an incoming IPv6 packet for NAT64 translation when all of the following conditions are met:

- The incoming IPv6 packet matches the ACL6 rule bound to a NAT64 rule.
- The destination IP address of the IPv6 packet matches the NAT64 IPv6 prefix.

When an IPv6 request packet received by the Citrix ADC appliance matches an ACL6 defined in a NAT64 rule and the destination IP of the packet matches the NAT64 IPv6 prefix, the Citrix ADC appliance considers the IPv6 packet for translation.

The appliance translates this IPv6 packet to an IPv4 packet with a source IP address matching one of the IP address bound to the netprofile defined in the NAT64 rule, and a destination IP address consisting of the last 32 bits of the destination IPv6 address of the IPv6 request packet. The Citrix ADC appliance creates a NAT64 session for this particular flow and forwards the packet to the IPv4 server. Subsequent responses from the IPv4 server and requests from the IPv6 client are translated accordingly by the appliance, on the basis of information in the particular NAT64 session.

Consider an example in which an enterprise hosts site www.example.com on server S1, which has an IPv4 address. To enable communication between IPv6 clients and IPv4 server S1, Citrix ADC appliance NS1 is deployed with a stateful NAT64 configuration that includes a NAT64 rule and a NAT64 prefix. A mapped IPv6 address of server S1 is formed by concatenating the NAT64 IPv6 prefix [96 bits] and the IPv4 source address [32 bits]. This mapped IPv6 address is then manually configured in the DNS servers. The IPv6 clients get the mapped IPv6 address from the DNS servers to communicate withIPv4 server S1.



The following table lists the settings used in this example: Stateful NAT64 translation example settings.

Following is the traffic flow in this example:

1. IPv6 client CL1 sends a request packet to Map-Sevr-IPv6 (2001:DB8:300::192.0.2.60) address.
2. The Citrix ADC appliance receives the request packet. If the request packet matches the ACL6 defined in the NAT64 rule, and the destination IP address of the packet matches the NAT64 IPv6 prefix, the Citrix ADC considers the IPv6 packet for translation.
3. The appliance creates a translated IPv4 request packet with:
   - Destination IP address field containing the NAT64 prefix stripped from the destination address of the IPv6 request (Sevr_IPv4 = 192.0.2.60)
   - Source IP address field containing one of the IPv4 address bound to Netprofile-1(in this case, 192.0.2.100)
4. The Citrix ADC appliance creates a NAT64 session for this flow and sends the translated IPv4 request to server S1.
5. IPv64 server S1 responds by sending an IPv4 packet to the Citrix ADC appliance with:
   - Destination IP address field containing 192.0.2.100
   - Source IP address field containing the address ofSevr_IPv4(192.0.2.60)
6. The appliance receives the IPv4 response packet, searches all the session entries, and finds that the IPv6 response packet matches the NAT64 session entry created in step 4. The appliance considers the IPv4 packet for translation.
7. The appliance creates a translated IPv6 response packet with:
   - Destination IP address field=Client_IPv6=2001:DB8:5001::30
   - Source IP address field = Concatenation of NAT64 Prefix (First 96 bits) and Sevr_IPv4 (last 32 bits) =2001:DB8:300::192.0.2.60

8. The appliance sends the translated IPv6 response to client CL1.

Limitations of Statelful NAT64

The following limitations apply to stateful NAT64:

- Translation of IPv4 options is not supported.
- Translation of IPv6 routing headers is not supported.
- Translation of hop-by-hop extension headers of IPv6 packets is not supported.
- Translation of ESP and EH headers of IPv6 packets is not supported.
- Translation of multicast packets is not supported.
- Packets of Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP), and IPSec, are not translated.

Configuring Stateful NAT64

Creating the required entities for stateful NAT64 configuration on the Citrix ADC appliance involves the following procedures:

1. Add an ACL6 rule with action ALLOW.
2. Add an ipset, which binds multiple IP addresses.
3. Add a netprofile and bind the ipset to it. If you want to bind only one IP address, you need not create an ipset entity. In that case, bind the IP address directly to the netprofile.
4. Add a NAT64 rule, which includes binding the ACl6 rule and the netprofile to the NAT 64 rule.
5. Add a NAT64 IPv6 prefix.

CLI procedures

To add an ACL6 rule by using the CLI:

At the command prompt, type:

- add ns acl6 <acl6name> <acl6action> …

To add an IPset and bind multiple IPs to it by using the CLI:

At the command prompt, type:

- add ipset <name>
- bind ipset <name> <IPaddress …>

To add a netprofile by using the CLI:

At the command prompt, type:

- add netprofile <name> -srcIP <IPaddress or IPset>

To add a NAT64 rule by using the CLI:

At the command prompt, type:

- add nat64 <name> <acl6name> -netProfile <string>

To add a NAT64 prefix by using the CLI:

At the command prompt, type:

```
set ipv6 -natprefix <ipv6_addr                    *>
```

-

Example:

```
 > add acl6 ACL6-1 ALLOW -srcIPv6 2001:DB8:5001::30
 Done

> apply acls6
 Done

> add ip 192.0.2.100 255.255.255.0  - type SNIP
 Done

> add ip 192.0.2.102 255.255.255.0   - type SNIP
 Done

> add ipset IPset-1
 Done

> bind ipset IPset-1 192.0.2.100 192.0.2.102
IPAddress "192.0.2.100" bound
IPAddress "192.0.2.102" bound
 Done

> add netprofile Netprofile-1 -srcIP IPset-1
 Done

> add nat64 NAT64-1 ACL6-1 -netprofile Netprofile-1
 Done

> set ipv6 -natprefix 2001:DB8:300::/96
 Done
<!--NeedCopy-->
```


GUI procedures

To add a NAT64 rule by using the GUI:

Navigate to System > Network > Routes > NAT64, and a new NAT64 rule, or edit an existing rule.

To add a NAT64 prefix by using the GUI:

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

RNAT

September 14, 2021

In Reverse Network Address Translation (RNAT), the Citrix ADC appliance replaces the source IP addresses in the packets generated by the servers with public NAT IP addresses. By default, the appliance uses a SNIP address as the NAT IP address. You can also configure the appliance to use a unique NAT IP address for each subnet. You can also configure RNAT by using Access Control Lists (ACLs). Use Source IP (USIP), Use Subnet IP (USNIP), and Link Load Balancing (LLB) modes affect the operation of RNAT. You can display statistics to monitor RNAT.

Note: The ephemeral port range for RNAT on the Citrix ADC appliance is 1024-65535.

You can use either a network address or an extended ACL as the condition for an RNAT entry:

- **Using a Network address.** When you use a network address, RNAT processing is performed on all of the packets coming from the specified network.
- **Using Extended ACLs.** When you use ACLs, RNAT processing is performed on all packets that match the ACLs. To configure the Citrix ADC appliance to use a unique IP address for traffic that matches an ACL, you must perform the following three tasks:

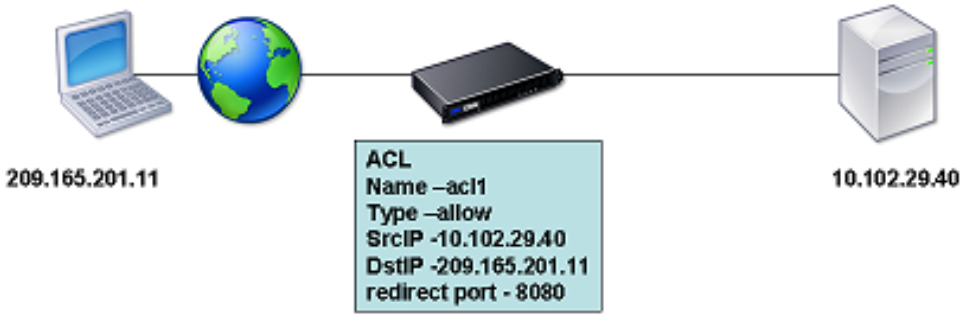
  1. Configure the ACL.
  2. Configure RNAT to change the source IP address and Destination Port.
  3. Apply the ACL.

  The following diagram illustrates RNAT configured with an ACL.

  Figure 1. RNAT with an ACL

Source IP Address	Destination IP Address	Destination Port
209.165.202.129	209.165.201.11	8080

Source IP Address	Destination IP Address
10.102.29.40	209.165.201.11

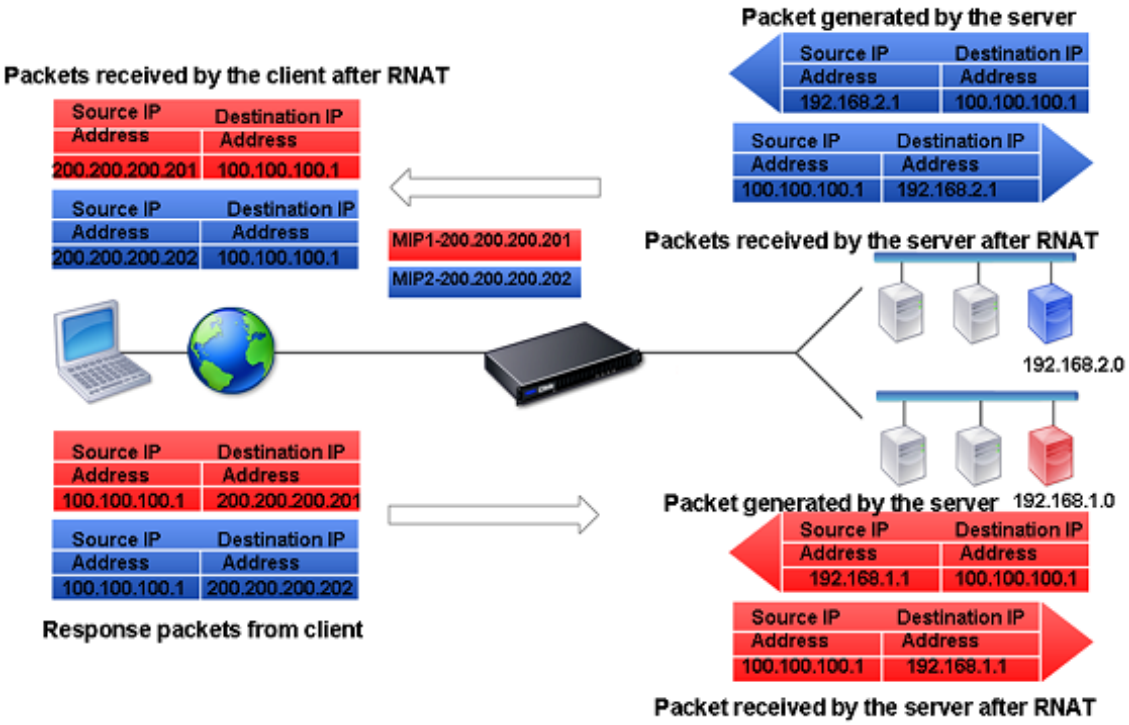


You have the following basic choices for the type of NAT IP address:

- **Using a SNIP as the NAT IP Address.** When using a SNIP as the NAT IP address, the Citrix ADC appliance replaces the source IP addresses of server-generated packets with the a SNIP. Therefore, the SNIP address must be a public IP address. If Use Subnet IP (USNIP) mode is enabled, the Citrix ADC can use a subnet IP address (SNIP) as the NAT IP address.
- **Using a Unique IP Address as the NAT IP Address.** When using a unique IP address as the NAT IP address, the Citrix ADC appliance replaces the source IP addresses of server-generated packets with the unique IP address specified. The unique IP address must be a public Citrix ADC-owned IP address. If multiple NAT IP addresses are configured for a subnet, NAT IP selection uses the round robin algorithm.

  This configuration is illustrated in the following diagram.

  Figure 2. Using a Unique IP Address as the NAT IP Address



Before you begin

Before configuring a RNAT rule, consider the following points:

- When RNAT and Use Source IP (USIP) are both configured on the Citrix ADC appliance, RNAT takes precedence. In other words, the source IP address of the packets, which matches a RNAT rule, is replaced according to the setting in the RNAT rule.
- In a topology where the Citrix ADC appliance performs both Link Load Balancing (LLB) and RNAT for traffic originating from the server, the appliance selects the source IP address based on the router. The LLB configuration determines selection of the router. For more information about LLB, see Link Load Balancing.

Configure RNAT

The following instructions provide separate command-line procedures for creating RNAT entries that use different conditions and different types of NAT IP addresses. In the GUI, all of the variations can be configured in the same dialog box, so there is only one procedure for GUI users.

CLI procedures

To create an RNAT rule by using the CLI:

At the command prompt, to create the rule and verify the configuration, type:

- `add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))`
- `bind rnat <name> <natIP>@ …`
- `show rnat`

To modify or remove an RNAT rule by using the CLI:

- To modify an RNAT rule:

  `set rnat <name> (<aclname> [-redirectPort <port>])`

- To remove an RNAT rule, type the command.

  `rm rnat <name>`

Use the following command to verify the configuration:

- `show rnat`

Examples:

```
A network address as the condition and a SNIP address as the NAT IP
    address:

> add rnat RNAT-1 192.168.1.0 255.255.255.0
Done

A network address as the condition and a unique IP address as the NAT
    IP address:

> add rnat RNAT-2 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-2 -natip 10.102.29.50
Done

If instead of a single NAT IP address you specify a range, RNAT entries
     are created with all the Citrix ADC-owned IP addresses, except the
    NSIP, that fall within the range specified:

> add rnat RNAT-3 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-3 -natip 10.102.29.[50-110]
Done


An ACL as the condition and a SNIP address as the NAT IP address:

> add rnat RNAT-4 acl1
```



```
Done

An ACL as a condition and a unique IP address as the NAT IP address:

> add rnat RNAT-4 acl1
Done

> bind rnat RNAT-4 -natip 10.102.29.50
Done

If instead of a single NAT IP address you specify a range, RNAT entries
     are created with all the Citrix ADC-owned IP addresses, except the
    NSIP, that fall within the range specified:

> add rnat RNAT-5 acl1
Done

> bind rnat RNAT-5 -natip 10.102.29.[50-70]
Done

<!--NeedCopy-->
```

GUI procedures

To create an RNAT entry by using the GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT** tab, and add a new RNAT rule, or edit an existing rule.

Monitor RNAT

You can display RNAT statistics to troubleshoot issues related to IP address translation.

The following table describes the statistics associated with RNAT and RNAT IP.

Statistic	Description
Bytes received	Bytes received during RNAT sessions
Bytes sent	Bytes sent during RNAT sessions
Packets received	Packets received during RNAT sessions
Packets sent	Packets sent during RNAT sessions

Statistic	Description
Syn sent	Requests for connections sent during RNAT sessions
Current sessions	Currently active RNAT sessions

To view RNAT statistics by using the CLI:

At the command prompt, type:

- **stat rnat**

Example:

```
> stat rnat

RNAT summary
                                Rate (/s)            Total
Bytes Received                    0                    0
Bytes Sent                        0                    0
Packets Received                  0                    0
Packets Sent                      0                    0
Syn Sent                          0                    0
Current RNAT sessions            --                    0
 Done
>
<!--NeedCopy-->
```

To monitor RNAT by using the GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT** tab, and then click **Statistics**.

Configure RNAT6

Reverse Network Address Translation (RNAT) rules for IPv6 packets are called RNAT6s. When an IPv6 packet generated by a server matches the conditions specified in the RNAT6 rule, the appliance replaces the source IPv6 address of the IPv6 packet with a configured NAT IPv6 address before forwarding it to the destination. The NAT IPv6 address is one of the Citrix ADC owned SNIP6 or VIP6 addresses.

When configuring an RNAT6 rule, you can specify either an IPv6 prefix or an ACL6 as the condition:

- **Using a IPv6 network address**. When you use an IPv6 prefix, the appliance performs RNAT processing on those IPv6 packets whose IPv6 address matches the prefix.
- **Using ACL6s**. When you use an ACL6, the appliance performs RNAT processing on those IPv6 packets that match the conditions specified in the ACL6.

You have one of the following options to set the NAT IP address:

- Specify a set of Citrix ADC owned SNIP6 and VIP6 addresses for an RNAT6 rule. The Citrix ADC appliance uses any one of the IPv6 addresses from this set as a NAT IP address for each session. The selection is based on the round robin algorithm and is done for each session.
- Do not specify any Citrix ADC owned SNIP6 or VIP6 address for an RNAT6 rule. The Citrix ADC appliance uses any one of the Citrix ADC owned SNIP6 or VIP6 addresses as a NAT IP address. The selection is based on the next hop network to which an IPv6 packet that matches the RNAT rule is destined.

CLI procedures

To create an RNAT6 rule by using the CLI:

At the command prompt, to create the rule and verify the configuration, type:

- **add rnat6** <name> (<network> | (<acl6name> [-**redirectPort** <port>]))
- **bind rnat6** <name> <natIP6>@ …
- **show rnat6**

To modify or remove an RNAT6 rule by using the CLI:

- To modify an RNAT6 rule whose condition is an ACL6, type the **set rnat6** <name> command, followed by a new value for the **redirectPort** parameter.
- To remove an RNAT6 rule, type the **clear rnat6** <name> command.

GUI procedures

To configure an RNAT6 rule by using the GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT6** tab, and add a new RNAT6 rule, or edit an existing rule.

Monitor RNAT6

You can display statistics related to the RNAT6 feature to monitor the performance or to troubleshoot problems related to RNAT6 feature. You can display a summary of statistics of the RNAT6 rules or of a particular RNAT6 rule. The statistical counters reflect events since the Citrix ADC appliance was last restarted. All these counters are reset to 0 when the Citrix ADC appliance is restarted.

The following lists some of the statistics counters associated with the RNAT6 feature:

- **Bytes received** - Total bytes received during RNAT6 sessions.
- **Bytes sent** - Total number of bytes sent during RNAT6 sessions.
- **Packets received** - Total number of packets received during RNAT6 sessions.

- **Packets sent** - Total number of packets sent during RNAT6 sessions.
- **Syn sent** - Total number of requests for connections sent during RNAT6 sessions
- **Current sessions** - Currently active RNAT6 sessions

To display a summarized statistics of all RNAT6 rules by using the CLI:

At the command prompt, type:

- **stat rnat6**

To display statistics for a specified RNAT6 rule by using the CLI:

At the command prompt, type:

- **stat rnat6** [<rnat6 rule name>]

To display RNAT6 statistics by using the GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT6** tab, and then click **Statistics**.

```
> stat rnat6

RNAT6 summary

                          Rate (/s)               Total

Bytes Received             178                20644

Bytes Sent                 178                20644

Packets Received            5                  401

Packets Sent                5                  401

Syn Sent                    0                    2

Current RNAT6 sessions      --                   1

Done

<!--NeedCopy-->
```

Log Start Time and Connection Closure Reasons in RNAT Log Entries

For diagnosing or troubleshooting problems related to RNAT, the Citrix ADC appliance logs RNAT sessions whenever they are closed.

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

- Citrix ADC owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp of log creation
- Protocol of the RNAT session
- Source IP address
- RNAT IP address
- Destination IP address
- Start time of the RNAT session
- Closing time of the RNAT session
- Total bytes sent by the Citrix ADC appliance for this RNAT session
- Total bytes received by the Citrix ADC appliance for this RNAT session
- Reason for closure of the RNAT session. The Citrix ADC appliance logs closure reason for TCP RNAT sessions that do not use the TCP proxy (TCP proxy disabled) of the appliance. The following are the type of closure reasons that are logged for TCP RNAT sessions:
  - **TCP FIN**. The RNAT session was closed because of a TCP FIN sent by either the source or destination device.
  - **TCP RST**. The RNAT session was closed because of a TCP Reset that was sent by either the source or destination device.
  - **TIMEOUT**. The RNAT session timed out.

The following table shows some sample log entries for RNAT sessions.

Type of Entry	Sample Log Entry
Sample log entry for UDP RNAT session	Dec 1 15:28:12 10.102.53.114 12/01/2015:15:28:12 GMT 0-PPE-0 : default UDP NAT_OTHERCONN_DELINK 154 0 : Source 1.2.2.5:23431 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:4045 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:26:58 GMT - Delink Time 12/01/2015:15:28:12 GMT - Total_bytes_send 2511 - Total_bytes_recv 3725

Type of Entry	Sample Log Entry
Sample log entry for TCP RNAT session. The log entry shows that the session closed because of TCP Reset	Dec 1 15:29:59 10.102.53.114 12/01/2015:15:27:59 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 152 0 : Source 1.2.2.5:33826 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:2384 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:27:40 GMT - Delink Time 12/01/2015:15:27:59 GMT - Total_bytes_send 2147 - Total_bytes_recv 3257 - Closure Reason TCP RST
Sample log entry for TCP RNAT session. The log entry shows that the session timed out	Dec 1 15:30:12 10.102.53.114 12/01/2015:15:30:12 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 155 0 : Source 1.2.2.5:64976 - Destination 192.168.123.115:22 - NatIP 192.168.123.1:19636 - Destination 192.168.123.115:22 - Start Time 12/01/2015:15:27:25 GMT - Delink Time 12/01/2015:15:30:12 GMT - Total_bytes_send 0 - Total_bytes_recv 0 - Closure Reason TIMEOUT

Stateful Connection Failover for RNAT

Connection failover helps prevent disruption of access to applications deployed in a distributed environment. The Citrix ADC appliance now supports stateful connection failover for connections related to RNAT rules in a Citrix ADC High Availability (HA) setup. In an HA setup, connection failover (or connection mirroring) refers to the process of keeping an established TCP or UDP connection active when a failover occurs.

The primary appliance sends messages to the secondary appliance to synchronize current information about the RNAT connections. The secondary appliance uses this connection information only in the event of a failover. When a failover occurs, the new primary Citrix ADC appliance has information about the connections established before the failover and hence continues to serve those connections even after the failover. From the client's perspective this failover is transparent. During the transition period, the client and server may experience a brief disruption and retransmissions.

Connection failover can be enabled per RNAT rule. For enabling connection failover on an RNAT rule, you enable the connFailover (Connection Failover) parameter of that specific RNAT rule by using either CLI or GUI.

To enable connection failover for a RNAT rule by using the CLI:

At the command prompt, type:

- `set rnat <name> -connfailover (ENABLED | DISABLED)`
- `show rnat`

To enable connection failover for a RNAT rule by using the GUI:

1. Navigate to **System** > **Network** > **NATs**, and then click the **RNAT** tab.
2. Select **Connection Failover** while adding a new RNAT rule, or while editing an existing rule.

Reserving the source port for RNAT connections to servers

For a request hitting an RNAT configuration that has one or more RNAT IP addresses and Use Proxy port parameter disabled, the Citrix ADC appliance uses one of the RNAT IP address and the source port of the RNAT request for connecting to servers. Prior to the 13.0 47.x build, RNAT connection (using the RNAT client's source port) to the server fails if the same source port is already been used in some other connections.

- **Source port less than 1024**. By default, the Citrix ADC appliance reserves the first 1024 ports of any Citrix ADC owned IP address (including RNAT IP addresses). Prior to 13.0 47.x build, RNAT connection (using the RNAT client's source port) to the server fails if the source port of the RNAT request is less than or equal to 1024. With the 13.0 47.x build, RNAT connection (using the RNAT client's source port) to the server succeeds even if the source port of the RNAT request is less than or equal to 1024.
- **Source port greater than 1024**. Prior to the 13.0 47.x build, RNAT connection (using the RNAT client's source port) to the server fails if the same source port is already been used in some other connections. With 13.0 47.x build, you can specify a range of RNAT client source ports in the `Retain Source Port range` (`retainsourceportrange`) parameter as part of an RNAT configuration. The Citrix ADC appliance reserves these RNAT client source ports on the RNAT IP address to be used only for RNAT connection to servers.

Removing RNAT Sessions

You can remove any unwanted or inefficient RNAT sessions from the Citrix ADC appliance. The appliance immediately releases resources (such as the port of the NAT IP address, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected RNAT sessions from the Citrix ADC appliance.

To clear all RNAT sessions by using the CLI:

At the command prompt, type:

- **flush rnatsession**

To clear selective RNAT sessions by using the CLI:

At the command prompt, type:

- **flush rnatsession** ((-**network** <ip_addr> -**netmask** <netmask>) | -**natIP** <ip_addr> | -**aclname** <string>)

To clear all or selective RNAT sessions by using the GUI:

1. Navigate to **System** > **Network** > **NATs**, and then click the **RNAT** tab.
2. In the **Actions** Menu, click **Flush RNAT Sessions** to remove all or selective RNAT sessions (for example, removing RNAT sessions with a specific RNAT IP, or belonging to a specific Network or ACL based RNAT rule).

Sample Configurations:

```
    Clear all RNAT sessions existing on a Citrix ADC appliance

    > flush rnatsession

    Done

    Clear all RNAT sessions belonging to network based RNAT rules that
       has 203.0.113.0/24 network as the matching condition.

    > flush rnatsession -network 203.0.113.0 -netmask 255.255.255.0

    Done

    Clear all RNAT sessions with RNAT IP 192.0.2.90.

    > flush rnatsession -natIP 192.0.2.90

    Done

    Clear all RNAT sessions belonging to ACL based RNAT rules that has
       ACL-RNAT-1 as the matching condition.

    > flush rnatsession -aclname ACL-RNAT-1

    Done
<!--NeedCopy-->
```


Configuring Prefix-Based IPv6-IPv4 Translation

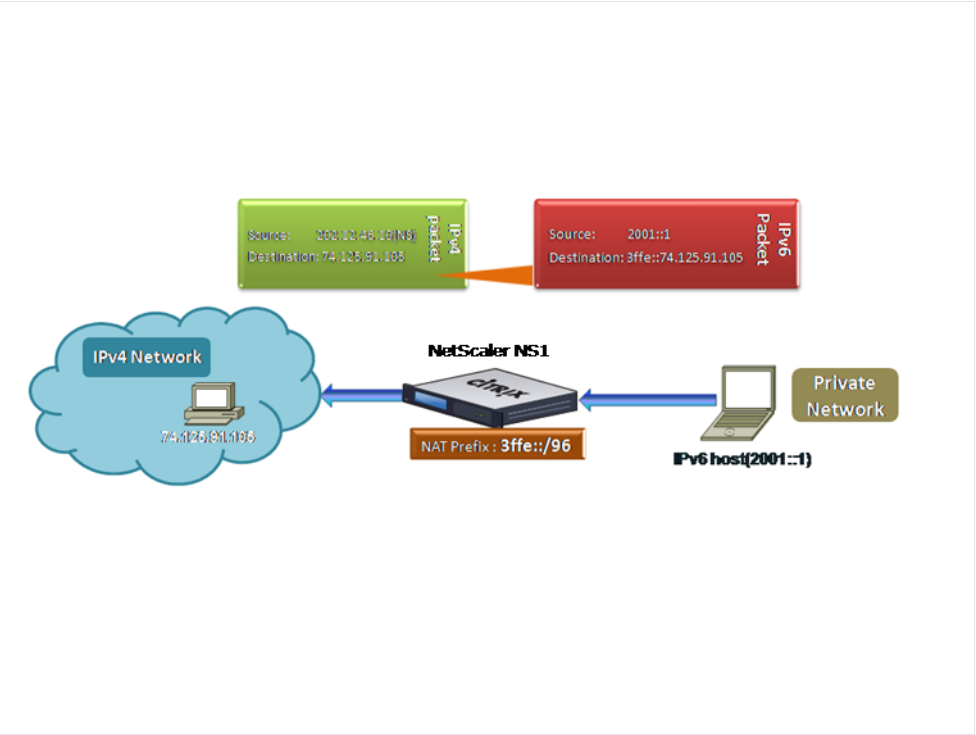
September 14, 2021

Prefix-based translation is a process of translating packets sent from private IPv6 servers into IPv4 packets, using an IPv6 prefix configured in the Citrix ADC appliance. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix.

The Citrix ADC appliance compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix. If there is a match, the Citrix ADC appliance generates an IPv4 packet and sets the destination IP address as the last 32 bits of the destination IP address of the matched IPv6 packet. IPv6 packets addressed to this prefix have to be routed to the Citrix ADC so that the IPv6-IPv4 translation is done by the Citrix ADC.

In the following diagram, 3ffe::/96 is configured as the IPv6 NAT prefix on Citrix ADC NS1. The IPv6 host sends an IPv6 packet with destination IP address 3ffe::74.125.91.105. NS1 compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix, and they match. NS1 then generates an IPv4 packet and sets the destination IP address as 74.125.91.105.

Figure 1. IPv6-IPv4 Prefix-Based Translation



To configure prefix-based IPv6-IPv4 translation by using the CLI:

At the command prompt, type:

```
set ipv6 [-natprefix <ipv6_addr                    *>]
```

-
- show ipv6

Example:

```
> set ipv6 -natprefix 3ffe::/96
 Done
<!--NeedCopy-->
```

To configure prefix-based IPv6-IPv4 translation by using the GUI:

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

IP Prefix NAT

September 14, 2021

The Citrix ADC appliance supports translating a part of the source IP address instead of the complete address of packets received on the appliance. IP prefix NAT includes changing one or more octets or bits of the source IP address.

The Citrix ADC appliance supports IP prefix NAT for load balancing configurations of the following types: ANY, UDP, DNS, TCP, and HTTP.

Use Case: Zonification of Clients for a Deployment of a Citrix ADC appliance and an Optimization Device

IP prefix NAT is very useful in a deployment that includes a Citrix ADC appliance and an optimization device (for example, Citrix ByteMobile). This type of deployment has different geographically located client networks, which share the same network address. The Citrix ADC appliance must send the traffic received from each of the client networks to the optimization device before forwarding to the destination.

The device sends the optimized traffic back to the Citrix ADC appliance. Because the optimization requirement is different for traffic from each client network, the optimization device must recognize the client network of each packet that it receives. The solution is to segregate traffic from each client

network into a different zone by using VLANs. IP prefix NAT with a different setting is configured for each zone. The Citrix ADC appliance translates the last octet of the source IP address of every packet, and the translated octet value is different for each zone.

Consider an example of two zones, Z1 and Z2, sharing network address 192.0.2.0/24. On the Citrix ADC appliance, IP prefix NAT entities named natrule-1 and natrule-2 are configured for these two zones. Before the appliance forwards a packet from Z1, natrule-1 translates the last octet of the packet's source IP address to 100. Similarly, for packets from Z2, natrule-2 translates the last octet of the source IP address to 200. For two clients, CL1-Z1 in zone Z1 and CL1-Z2 in zone Z2, each with IP address 192.0.2.30, the Citrix ADC appliance translates the source IP address of CL1-Z1's packets to 100.0.2.30 and of CL1-Z2's packets to 200.0.2.30. The optimization device to which the Citrix ADC appliance sends the translated packets is configured to use a packet's source IP address to recognize the zone, so it applies the appropriate optimization configured for the zone from which the packet originated.

Configuration Steps

Configuring IP prefix NAT consists of the following steps:

- **Create a net profile and set the NAT Rule parameter of a net profile**. A NAT rule specifies two IP addresses and a net mask. The first IP address (specified by IP Address parameter) is the source IP address that is to be translated with the second one (specified by IP Rewrite parameter). The net mask specifies the part of the source IP address that is to be translated with the same part of the second IP address.
- **Bind the net profile to load balancing virtual servers or services**. A net profile with NAT rule setting can be bound to a virtual server or service of type ANY, UDP, DNS, TCP, and HTTP. After binding a net profile to a virtual server or service, the Citrix ADC appliance matches the source IP address of the incoming packets related to the virtual server or service with the NAT rule setting. The Citrix ADC then performs IP prefix NAT for packets that match the NAT rule.

To configure IP prefix NAT translation by using the command line:

At the command prompt, type:

- **bind netProfile** <name> (-**natRule** <ip_addr> <netmask> <rewriteIp>)
- **show netprofile** <name>

To configure IP prefix NAT by using the GUI:

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the following parameters under NAT Rules while adding or modifying NetProfiles.
    - IP Address
    - Netmask
    - Rewrite IP

Sample configuration

In the following sample configuration, net profile PARTIAL-NAT-1 has IP prefix NAT settings and is bound to load balancing virtual server LBVS-1, which is of type ANY. For packets received on LBVS-1 from 192.0.0.0/8, the Citrix ADC appliance translates the last octet of the packet's source IP address to 100. For example, a packet with source IP address 192.0.2.30 received on LBVS-1, the Citrix ADC appliance translates the source IP address to 100.0.2.30 before sending it one of the bound servers.

```
> add netprofile PARTIAL-NAT-1
Done

> bind netprofile PARTIAL-NAT-1  - natrule 192.0.0.0 255.0.0.0 100.0.0.0
Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
Done
<!--NeedCopy-->
```

Static ARP

September 14, 2021

You can add static ARP entries to and remove static ARP entries from the ARP table. After adding an entry, you should verify the configuration. If the IP address, port, or MAC address changes after you create a static ARP entry, you must remove or manually adjust the static entry. Therefore, creating static ARP entries is not recommended unless necessary.

To add a static ARP entry by using the CLI:

At the command prompt, type:

- **add arp -IPAddress** <ip_addr> -**mac**<mac_addr> -**ifnum** <interface_name>
- **show arp** <IPAddress>

Example:

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
<!--NeedCopy-->
```

To remove a static ARP entry by using the CLI:

At the command prompt, type the **rm arp** command and the IP address.

To add a static ARP entry by using the GUI:

Navigate to **System** > **Network** > **ARP Table**, and add a static ARP entry.

Specify a VLAN in a Static ARP Entry

In a static ARP entry, you can specify the VLAN through which the destination device is accessible. This feature is useful when the interface specified in the static ARP entry is part of multiple tagged VLANs and the destination is accessible through one of the VLANs. The Citrix ADC appliance includes the specified VLAN ID in the outgoing packets matching the static ARP entry. If you don't specify a VLAN ID in an ARP entry, and the specified interface is part of multiple tagged VLANs, the appliance assigns the interface's native VLAN to the ARP entry.

For example, say Citrix ADC interface 1/2 is part of native VLAN 2 and of tagged VLANs 3 and 4, and you add a static ARP entry for network device A, which is part of VLAN 3 and is accessible through interface 1/2. You must specify VLAN 3 in the ARP entry for network device A. The Citrix ADC appliance then includes tagged VLAN 3 in all the packets destined to network device A, and sends them from interface 1/2.

If you don't specify a VLAN ID, the Citrix ADC appliance assigns native VLAN 2 for the ARP entry. Packets destined to device A are dropped in the network path, because they do not specify tagged VLAN 3, which is the VLAN for device A.

To specify a VLAN in a static ARP entry by using the CLI:

At the command prompt, type:

- **add arp -IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name> [-**vlan** <positive_integer>]
- **show arp** <IPAddress>

Example:

```
1 > add arp -ip 198.51.100.91 -mac 36:db:4b:f6:12:15 -ifnum 1/2 -vlan 3
2 Done
3 <!--NeedCopy-->
```

Set the Timeout for Dynamic ARP Entries

September 14, 2021

You can globally set an aging time (time-out value) for dynamically learned ARP entries. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing

ARP entries expire after the previously configured aging time. You can specify an ARP time-out value of from 1 through 1200 seconds.

To set the time-out for dynamic ARP entries by using the CLI:

At the command prompt, type:

- **set arpparam** -**timeout** <positive_integer>]
- **show arpparam**

Example:

```
> set arpparam -timeout 500
 Done
<!--NeedCopy-->
```

To set the time-out for dynamic ARP entries to its default value by using the CLI:

At the command prompt, type:

- **unset arpparam**
- **show arpparam**

Example:

```
> unset arpparam
 Done
<!--NeedCopy-->
```

To set the time-out for dynamic ARP entries by using the GUI:

Navigate to **System** > **Network**, in the **Settings** group, click **Configure ARP Global Parameters**, and set the **ARP Table Entry Timeout** parameter.

Neighbor Discovery

September 14, 2021

Neighbor discovery (ND) is one of the most important protocols of IPv6. It is a message-based protocol that combines the functionality of the Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and Router Discovery. ND allows nodes to advertise their link layer addresses and obtain the MAC addresses or link layer addresses of the neighboring nodes. This process is performed by the Neighbor Discovery protocol (ND6).

Neighbor discovery can perform the following functions:

- **Router Discovery**: Enables a host to discover the local routers on an attached link and automatically configure a default router.
- **Prefix Discovery**: Enables the host to discover the network prefixes for local destinations.

  Note: The Citrix ADC appliance does not support Prefix Discovery.
- **Parameter Discovery**: Enables a host to discover additional operating parameters, such as MTU and the default hop limit for outbound traffic.
- **Address Autoconfiguration**: Enables hosts to automatically configure IP addresses for interfaces both with and without stateful address configuration services such as DHCPv6. The Citrix ADC does not support Address Autoconfiguration for Global IPv6 addresses.
- **Address Resolution**: Equivalent to ARP in IPv4, enables a node to resolve a neighboring node's IPv6 address to its link-layer address.
- **Neighbor Unreachability Detection**: Enables a node to determine the reachability state of a neighbor.
- **Duplicate Address Detection**: Enables a node to determine whether an NSIP address is already in use by a neighboring node.
- **Redirect**: Equivalent to the IPv4 ICMP Redirect message, enables a router to redirect the host to a better first-hop IPv6 address to reach a destination.

Note: The Citrix ADC appliance does not support IPv6 Redirect.

Configuration Steps

Configuring neighbor discovery consists of the following tasks:

- Adding IPv6 neighbors
- (Optional) Removing IPv6 neighbors

CLI procedures

To add an IPv6 neighbor by using the CLI:

At the command prompt, type:

- **add nd6** <neighbor> <mac> <ifnum> [-**vlan** <integer>]
- **sh nd6**

Example:

```
> add nd6 2001::1 00:04:23:be:3c:06 1/1  - vlan 1
 Done

```



```
> show nd6
   Neighbor                 MAC-Address(Vlan, Interface)       State
             TIME
   --------                 ----------------------------       -----
            --------
1) ::1                      00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
        PERMANENT
2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
        PERMANENT
3) 2001::1                  00:04:23:be:3c:06(   1,   1/1)   REACHABLE
          STATIC
 Done
<!--NeedCopy-->
```

To remove a neighbor discovery entry by using the CLI:

At the command prompt, type:

- **rm nd6** <Neighbor> -**vlan** <VLANID>

Example:

```
 rm nd6 3ffe:100:100::1 -vlan 1
<!--NeedCopy-->
```

To remove all neighbor discovery entries by using the CLI:

At the command prompt, type:

- **clear nd6**

GUI procedures

To add an IPv6 neighbor by using the GUI:

Navigate to **System** > **Network** > **IPv6 Neighbors**, and add a new IPv6 neighbor.

To remove a neighbor discovery entry by using the GUI:

Navigate to **System** > **Network** > **IPv6 Neighbors**, delete the IPv6 neighbor.

To remove all neighbor discovery entries by using the GUI:

Navigate to **System** > **Network** > **IPv6 Neighbors**, and click **Clear**.

IP Tunnels

September 14, 2021

An IP Tunnel is a communication channel, that can be created by using encapsulation technologies, between two networks that do not have a routing path. Every IP packet that is shared between the two networks is encapsulated within another packet and then sent via the tunnel.

The Citrix ADC appliance implements IP Tunneling in the following ways:

- **Citrix ADC as an Encapsulator (Load Balancing with DSR Mode)**: Consider an organization that has multiple data centers across different countries, where the Citrix ADC maybe at one location and the back-end servers are located in a different country. In essence, the Citrix ADC and the back-end servers are on different networks and are connected via a router.

  When you configure Direct Server Return (DSR) on this Citrix ADC, the packet sent from the source subnet is encapsulated by the Citrix ADC and sent via a router and tunnel to the appropriate back-end server. The back-end server decapsulates the packet and responds directly to the client, without allowing the packet to pass via the Citrix ADC.

- **Citrix ADC as a Decapsulator:** Consider an organization having multiple data centers each having Citrix ADCs and back-end servers. When a packet is sent from data center A to data center B it is usually sent via an intermediary, say a router or another Citrix ADC. The Citrix ADC processes the packet and then forwards the packet to the back-end server. However, if an encapsulated packet is sent, the Citrix ADC must be able to decapsulate the packet before sending it to the back-end servers. To enable the Citrix ADC to function as a decapsulator, a tunnel is added between the router and the Citrix ADC. When the encapsulated packet, with additional header information, reaches the Citrix ADC, the data packet is decapsulated i.e. the additional header information is removed, and the packet is then forwarded to the appropriate back-end servers.

  The Citrix ADC can also be used as a decapsulator for the Load Balancing feature, specifically in scenarios when the number of connections on a vserver exceeds a threshold value and all the new connections are then diverted to a back-up vserver.

Configure IP Tunnels

Configuring IP tunnels on a Citrix ADC appliance consists of creating IP tunnel entities. An IP tunnel entity specifies the local and remote tunnel end-point IP addresses and the protocol to be used for the IP tunnel.

Note: While configuring an IP tunnel in a cluster setup, the local IP address must be a striped SNIP address.

CLI procedures

To create an IP tunnel by using the CLI:

At the command prompt type:

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**type** -**protocol** (**ipoverip** | **GRE**)
- **show iptunnel**

To remove an IP tunnel by using the CLI:

To remove an IP tunnel, type the **rm iptunnel** command and the name of the tunnel.

To create an IPv6 tunnel by using the CLI:

At the command prompt type:

- **add ip6tunnel** <name> <remoteIp> <local>
- **show ip6tunnel**

To remove an IPv6 tunnel by using the CLI:

To remove an IPv6 tunnel, type the **rm ip6tunnel** command and the name of the tunnel.

GUI procedures

To create an IP Tunnel by using the GUI:

Navigate to **System** > **Network** > **IP Tunnels**, add a new IP tunnel.

To create an IPv6 Tunnel by using the GUI:

Navigate to **System** > **Network** > **IP Tunnels** > **IPv6 Tunnels**, and add a new IPv6 tunnel.

Customizing IP Tunnels Globally

By globally specifying the source IP address, you can assign a common source IP address across all tunnels. Also, because fragmentation is CPU-intensive, you can globally specify that the Citrix ADC appliance drop any packet that requires fragmentation. Alternatively, if you would like to fragment all packets as long as a CPU threshold value is not reached, you can globally specify the CPU threshold value.

CLI procedures

To globally customize IP tunnels by using the CLI:

At the command prompt, type:

- **set ipTunnelParam** -**srcIP** <sourceIPAddress> -**srcIPRoundRobin** (**YES** | **NO**)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ipTunnelParam**

Example:


```
> set iptunnelparam  - srcIP 12.12.12.22 -dropFrag Yes  -
    dropFragCpuThreshold 50
 Done

> set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes  -
    dropFragCpuThreshold 50
 Done
<!--NeedCopy-->
```

To globally customize IPv6 tunnels by using the CLI:

At the command prompt, type:

- **set ip6tunnelparam** -**srcIP** <IPv6Address> -**srcIPRoundRobin** (**YES** | NO)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ip6tunnelparam**

GUI procedures

To globally customize IP tunnels by using the GUI:

Navigate to **System** > **Network**, in the Settings group, click **IPv4 Tunnel Global Settings**.

1. Navigate to **System** > **Network**, in the **Settings** group, click **IPv6 Tunnel Global Settings**.
2. In the **Configure IP Tunnel Global Parameters** dialog box, set the parameters.

To globally customize IPv6 tunnels by using the GUI:

1. Navigate to **System** > **Network**, in the **Settings** group, click **IPv6 Tunnel Global Settings**.
2. In the **Configure IP Tunnel Global Parameters** dialog box, set the parameters.

GRE Payload Options in a GRE IP Tunnel

For a configured GRE IP tunnel, the Citrix ADC appliance encapsulates the entire Layer 2 packet, including the Ethernet header and the VLAN header (dot1q VLAN tag). IP GRE tunnels between Citrix ADC appliances and some 3rd party devices might not be stable, because these 3rd party devices are not programmed to process some or the Layer 2 packet headers. To configure a stable IP GRE tunnel between a Citrix ADC appliance and a 3rd party device, you can use the GRE payload parameter of the GRE IP tunnel command set. GRE payload setting can also be applied to a GRE with IPSec tunnel.

You can set the GRE payload parameter to do one of the following before the packet is sent through the GRE tunnel:

- **Ethernet with DOT1Q**. Carry the Ethernet header as well the VLAN header. This is the default setting. For a tunnel bound to a netbridge, inner Ethernet header and VLAN header contains

information from the ARP and bridge table of the Citrix ADC appliance. For a tunnel set as a next hop to a PBR rule, Inner Ethernet destination MAC address is set to zero and the VLAN header specifies the default VLAN. The encapsulated (GRE) packet sent from the Citrix ADC tunnel end point has the following format:

Outer Ethernet Header	Outer IP Header	GRE Header	Inner Ethernet	Inner VLAN header	Inner IP/IPv6/ARP header	Inner TCP/UDP Header	Payload

- **Ethernet**. Carry the Ethernet header but drop the VLAN header. Because the packets do not carry any VLAN information in the tunnel, for a tunnel with this setting and bound to a netbridge, you must bind an appropriate VLAN to the netbridge so that on receiving any packets on the tunnel, the Citrix ADC can forward these packet to the specified VLAN. If the tunnel is set as a next hop in a PBR rule, the Citrix ADC routes the packets that are received on the tunnel. The encapsulated (GRE) packet sent from the Citrix ADC tunnel end point has the following format:

Outer Ethernet header	Outer IP header	GRE Header	Inner Ethernet header	Inner IP/IPv6/ARP header	Inner TCP/UDP header	Payload

- **IP**. Drop the Ethernet header as well as the VLAN header. Because tunnels with this setting do not carry Layer 2 headers, these tunnels cannot be bound to a netbridge but can be set as a next hop in a PBR rule. The peer tunnel endpoint device on receiving the packet either consumes or routes it. The encapsulated (GRE) packet sent from the Citrix ADC tunnel end point has the following format:

Outer Ethernet header	Outer IP header	GRE header	Inner IP/IPv6 header	Inner TCP/UDP header	Payload

To drop Layer 2 headers of packets in a GRE IP tunnel by using the CLI:

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> [-**protocol** <GRE> [-**vlan** <positive_integer>]] [-**grepayload** <grepayload>] [-**ipsecProfileName** <string>]
- **show iptunnel** <tunnelname>

Example:

```
> add iptunnel IPTUNNEL-1 203.0.113.133 255.255.255.0 198.51.100.15 -
    protocol GRE  - grepayload Ethernet -ipsecProfileName IPTUNNEL-IPSEC
    -1
Done
<!--NeedCopy-->
```


IPv6 Traffic through GRE IPV4 Tunnels

The Citrix ADC appliance supports transferring IPv6 traffic through an IPV4 GRE tunnel. This feature can be used for enabling communication between Isolated IPv6 networks without upgrading the IPv4 infrastructure between them.

For configuring this feature, you associate a PBR6 rule with the configured IPv4 GRE tunnel through which you want the Citrix ADC to send and receive IPv6 traffic. The source IPv6 address and destination IPv6 address parameters of the PBR6 rule specify the IPv6 networks whose traffic is to traverse the IPv4 GRE tunnel.

Note: IPSec protocol is not supported on GRE IPv4 tunnels that are configured to transfer IPv6 packets.

To create a GRE IPv4 tunnel by using the CLI:

At the command prompt, type:

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol GRE**
- **show ipTunnel** <name>

To associate a PBR6 rule with a GRE IPv4 tunnel by using the CLI:

- **add ns pbr6** <pbrName> **ALLOW** -**srcIPv6** <network-range> -**dstIPv6** <network-range> -**ipTunnel** <tunnelName>
- **show pbr**

Sample configuration

In the following sample configuration, GRE IP tunnel TUNNEL-V6onV4 is created with remote tunnel endpoint IP address 10.10.6.30 and local tunnel endpoint IP address 10.10.5.30. The tunnel is then bound to pbr6 PBR6-V6onV4. The srcIPv6 specifies the IPv6 network connected to the local endpoint and destIPv6 specifies the IPv6 network connected to the remote endpoint. The traffic from these Ipv6 networks are allowed to traverse through the GRE IPv4 tunnel.

```
1 > add ipTunnel TUNNEL-V6onV4 10.10.6.30 255.255.255.255 10.10.5.30 -
     protocol GRE
2 -ipsecProfileName None
3 Done
4 > add ns pbr6 PBR6-V6onV4 ALLOW -srcIPv6 = 2001:0db8:1::1-2001:0db8
     :1::255 -destIPv6 =
5 1-2001:0db8:4::255 -ipTunnel TUNNEL-V6onV4
6 <!--NeedCopy-->
```


Send Response Traffic Through an IP-IP Tunnel

You can configure a Citrix ADC appliance to send response traffic through an IP-IP tunnel instead of routing it back to the source. By default, when the appliance receives a request from another Citrix ADC or a third-party device through an IP-IP tunnel, it routes the response traffic instead of sending it through the tunnel. You can use policy based routes (PBRs) or enable MAC-Based Forwarding (MBF) to send the response through the tunnel.

In a PBR rule, specify the subnets at both end points whose traffic is to traverse the tunnel. Also set the next hop as the tunnel name. When response traffic matches the PBR rule, the Citrix ADC appliance sends the traffic through the tunnel.

Alternatively, you can enable MBF to meet this requirement, but the functionality is limited to traffic for which the Citrix ADC appliance stores session information (for example, traffic related to load balancing or RNAT configurations). The appliance uses the session information to send the response traffic through the tunnel.

CLI procedures

To create a PBR rule and associate the IP-IP tunnel to it by using the CLI:

At the command prompt, type:

- **add ns pbr** <pbr_name> **ALLOW** -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**
- **show ns pbr** <pbr_name>

To enable MAC-based forwarding by using the CLI:

At the command prompt, type:

- **enable ns mode MBF**
- **show ns mode**

GUI procedures

To create a PBR rule and associate the IP-IP tunnel to it by using the GUI:

1. Navigate to **System** > **Network** > **PBRs**. On the **PBRs** tab, create a **PBR** rule.
2. While creating the PBR, set the **Next Hop Type** to **IP tunnel** and **IP Tunnel Name** to the configured IP-IP tunnel name.

To enable MAC-based forwarding by using the GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features**, click **Configure Modes**.
2. On the **Configure Modes** page, select **MAC-based forwarding**.

Sample configuation

Consider an example of an IPIP tunnel, NS1-NS2-IPIP, which is set up between two Citrix ADC appliances NS1 and NS2.

By default, for any request that NS2 receives through the tunnel, its routes the response traffic to the source instead of sending it (to NS1) through the tunnel.

You can configure policy based routes (PBRs) or enable MAC-Based Forwarding (MBF) on NS2 for enabling it to send the response through the tunnel.

In the following sample configuration on NS2, NS1-NS2-IPIP is an IPIP tunnel and NS1-NS2-IPIP-PBR is a PBR rule. For requests (with inner source IP address in the range 10.102.147.0-10.102.147.255 and inner destination IP address in the range 10.102.147.0-10.102.147.255) received by NS2 through the tunnel, NS2 sends the corresponding response through the tunnel (to NS1) instead of routing it to the source. The functionality is limited to the traffic that matches the PBR rule.

```
1 > add iptunnel NS1-NS2-IPIP 192.0.2.99 255.255.255.255 203.0.113.99 -
     protocol IPIP
2
3 Done
4 > add pbr NS1-NS2-IPIP-PBR -srcIP 10.102.147.0-10.102.147.255  - destIP
     10.20.1.0-10.20.1.255  - ipTunnel NS1-NS2-IPIP
5
6 Done
7 > apply pbrs
8
9 Done
```

Alternatively, MBF can be enabled on NS2. The functionality is limited to traffic for which NS2 stores session information (for example, traffic related to load balancing or RNAT configurations).

```
1 > enable ns mode MBF
2
3 Done
```

Class E IPv4 packets

September 14, 2021

By default, the Citrix ADC appliance drops any packets if they contain any class E IPv4 address in the source IP or the destination IP fields. If your setup is using class E IPv4 addresses, you can configure the Citrix ADC appliance to process class E IPv4 packets.

Before you begin

Before you begin configuring a Citrix ADC appliance to process class E IPv4 packets, note the following points:

- Citrix ADC appliances do not support configuring any Citrix ADC owned IPv4 address (for example, SNIP and VIP) in the class E range. Citrix ADC appliances only support processing of class E IPv4 packets.
- A Citrix ADC appliance internally uses class E IPv4 addresses for the IPv6 feature. The Citrix ADC appliance does not support both features (processing class E IPv4 packets and IPv6 support) working at the same time. The Citrix ADC appliance imposes a restriction to not enable the IPv6 feature when processing of class E IPv4 packets is enabled, and vice versa.

Configuration steps

Configuring a Citrix ADC appliance to process class E IPv4 packets consist of the task of enabling the **IPv4 Class E address clients** (**allowClassEIPv4**) Layer 3 parameter.

CLI procedures

To configure the Citrix ADC appliance to process Class E IPv4 packets by using the CLI:

At the command prompt, type:

set l3param -allowClassEIPv4 (**ENABLED** **DISABLED**)

-
- **show l3param**

Sample Configuration:

```
> set l3param -allowClassEIPv4 ENABLED

Done

> sh l3param

    Network L3 related Configuration Parameters

    icmpgen_rate_threshold      : 100

    srcnat                      : ENABLED

```



```
13     override_rnat                 : DISABLED
14
15     drop_df_flag                  : DISABLED
16
17     .
18
19     .
20
21     .
22
23     IPv6DynamicRouting           : DISABLED
24
25     allowClassEIPv4              : ENABLED
26
27 Done
28 <!--NeedCopy-->
```

GUI procedures

To configure the Citrix ADC appliance to process Class E IPv4 packets by using the GUI:

1. Navigate to **System** > **Network**, and then in the **Settings** section, click **Configure Layer 3 Parameters**.
2. Select **IPv4 Class E address clients** and click **OK**.

Interfaces

September 14, 2021

Before you begin configuring interfaces, decide whether your configuration can use MAC-based forwarding mode, and either enable or disable this system setting accordingly. The number of interfaces in your configuration is different for the different models of the Citrix ADC appliance. In addition to configuring individual interfaces, you can logically group interfaces, using VLANs to restrict data flow within a set of interfaces, and you can aggregate links into channels. In a high availability setup, you can configure a virtual MAC address if necessary. If you use L2 mode, you might want to modify the aging of the bridge table.

When your configuration is complete, decide whether you should enable the system setting for path MTU discovery. Citrix ADC appliances can be deployed in active-active mode using VRRP. An active-active deployment, in addition to preventing downtime, makes efficient use of all the Citrix ADC appliances in the deployment. You can use the Network Visualizer tool to view the network configuration of a Citrix ADC deployment and configure interfaces, channels, VLANs, and bridge groups.

Configuring MAC-Based Forwarding

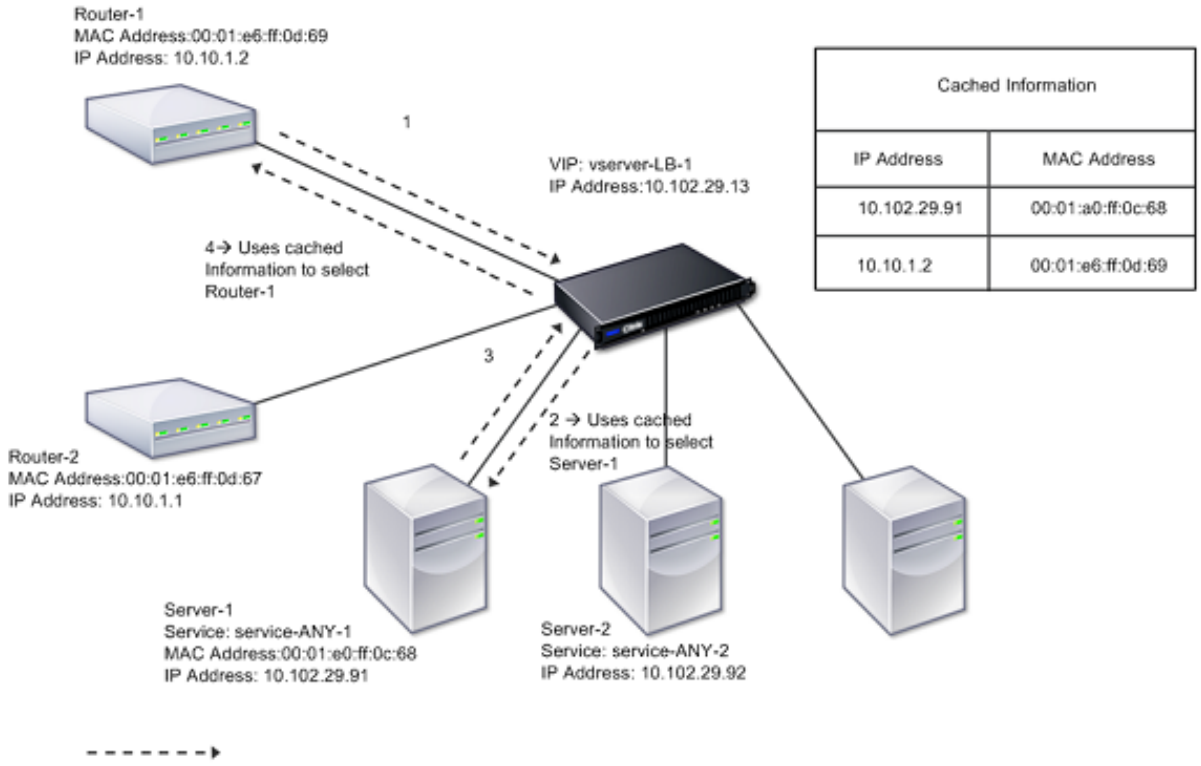
September 14, 2021

With MAC-based forwarding (MBF) enabled, when a request reaches the Citrix ADC appliance, the appliance remembers the source MAC address of the frame and uses it as the destination MAC address for the resulting replies. MAC-based forwarding can be used to avoid multiple-route/ARP lookups and to avoid asymmetrical packet flows. MAC-based forwarding may be required when the Citrix ADC is connected to multiple stateful devices, such as VPNs or firewalls, because it ensures that the return traffic is sent to the same device that the initial traffic came from.

MAC-based forwarding is useful when you use VPN devices, because it guarantees that all traffic flowing through a VPN passes back through the same VPN device.

The following topology diagram illustrates the process of MAC-based forwarding.

Figure 1. MAC-Based Forwarding Mode



When MAC-based forwarding (MBF) is enabled, the Citrix ADC caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server replies through the Citrix ADC appliance, the appliance sets the destination MAC ad-

dress of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when the Citrix ADC initiates a connection, it uses the route and ARP tables for the lookup function. In a direct server return configuration, you must enable MAC-based forwarding.

For more information about direct server return configurations, see Load Balancing.

Some deployment topologies may require the incoming and outgoing paths to flow through different routers. MAC-based forwarding would break this topology design.

MBF should be disabled in the following situations:

- **When a server uses network interface card (NIC) teaming without using LACP (802.1ad Link Aggregation).** To enable MAC-based forwarding in this situation, you must use a layer 3 device between the Citrix ADC and server.
  Note: MBF can be enabled when the server uses NIC teaming with LACP, because the virtual interface uses one MAC address.
- **When firewall clustering is used**. Firewall clustering assumes that ARP is used to resolve the MAC address for inbound traffic. Sometimes the inbound MAC address can be a non-clustered MAC address and should not be used for inbound packet processing.

When MBF is disabled, the appliance uses L2 or L3 connectivity to forward the responses from servers to the clients. Depending on the route table, the routers used for outgoing connection and incoming connection can be different. In the case of reverse traffic (response from the server):

- If the source and destination are on different IP subnets, the appliance uses the route lookup to locate the destination.
- If the source is on the same subnet as the destination, the Citrix ADC looks up the ARP table to locate the network interface and forwards the traffic to it. If the ARP table does not exist, the Citrix ADC requests the ARP entries.

To enable or disable MAC-based forwarding by using the CLI:

At the command prompt, type:

- **enable ns mode MBF**
- **disable ns mode MBF**

To enable or disable MAC-based forwarding by using the GUI:

1. Navigate to **System** > **Settings**, in the **Modes and Features** group, click **Configure modes**.
2. Select or clear the **MAC-based forwarding** option.

MAC based forwarding for a load balancing setup

Some load balancing setups require that the Citrix ADC appliance bypasses the global MBF (if enabled) for these setups and instead use the route/ARP lookups for sending packets to the destination.

The MBF parameter of a net profile is used to enable or disable MBF for a specific load balancing configuration. MBF can be set for the client side as well as the server side of a load balancing configuration by binding net profiles (MBF enabled or disabled) to the virtual server and the services.

For example, if a net profile with MBF disabled is bound to the virtual server of a load balancing configuration, the Citrix ADC appliance bypasses the global MBF (if enabled) and instead use the route/ARP lookups for sending response packets to clients.

Before you begin

Before you begin configuring MBF for a load balancing configuration, note the following points:

- In a load balancing configuration, the client side (virtual server) and the server side (service/service groups) can have different MBF settings.
- A load balancing configuration inherits global MBF setting if MBF is not set explicitly in the net profiles bound to the virtual server and the services.
- In a load balancing configuration, server side (service) inherits client side MBF setting of net profile bound to the virtual server if no net profile is bound to the service.
- In a load balancing configuration with direct server return mode, client side inherits the MBF setting in the net profile bound to the service.
- In a content switching configuration, client side takes the MBF setting in the net profile bound to the content switching virtual server instead of from the target load balancing virtual server.

Limitations

Before you begin configuring MBF for a load balancing configuration, note the following limitations:

- MBF setting for load balancing configurations is not supported in a cluster setup.
- For a load balancing virtual server with MAC mode or L2Conn settings, MBF is enabled irrespective of the MBF setting in the bound net profile to the virtual server.
- The Citrix ADC appliance does not support setting MBF for load balancing monitors using net profile. In other words, the MBF setting of a net profile is not applied to the monitors to which the net profile is bound. The global MBF setting is applied to monitors irrespective of the MBF setting of the bound net profile.

Configure MBF for load balancing configuration

Configuring MBF for a load balancing configuration consists of the following tasks:

- Enable MBF parameter in a net profile.
- Bind the net profile to a load balancing virtual server or services.

To enable MBF in a net profile by using the CLI:

- To enable MBF while adding a net profile, at the command prompt, type:
  - **add netProfile** <name> -**MBF** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>
- To enable MBF in an existing net profile, at the command prompt, type:
  - **set netProfile** <name> -**MBF** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>

To enable MBF in a net profile by using GUI**

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Enable the **MBF** parameter while adding or modifying a net profile.

In the following sample configuration, net profile NETPROFILE-MBF-LBVS has MBF enabled and is bound to load balancing virtual server LBVS-1. Also, net profile NETPROFILE-MBF-SVC has MBF enabled and is bound to a load balancing service SVC-1.

```
> add netprofile NETPROFILE-MBF-LBVS -MBF ENABLED

Done

> add netprofile NETPROFILE-MBF-SVC -MBF ENABLED

Done

> set lb vserver LBVS-1 -netprofile NETPROFILE-MBF-LBVS

Done

> set service SVC-1 -netprofile NETPROFILE-MBF-SVC

Done

<!--NeedCopy-->
```

Configure network interfaces

September 14, 2021

Network interfaces in the Citrix ADC appliance are numbered in <slot><port> notation. After configuring your interfaces, display the interfaces and their settings to verify the configuration. You can also display this information to troubleshoot a problem in the configuration.

To manage the network interfaces, you can do the following;

- Enable some interfaces and disable others.
- Reset an interface to renegotiate its settings.
- Clear the accumulated statistics for an interface.

To verify the configuration, you can display the interface settings. You can display the statistics for an interface to evaluate its health.

Set the network interface parameters

The network interface configuration is neither synchronized nor propagated. For an HA pair, you must perform the configuration on each unit independently.

To set the network interface parameters by using the CLI:

At the command prompt, type:

```
1 -  set interface <id> [-speed <speed>] [-duplex <duplex>] [-flowControl
      <flowControl>] [-autoneg ( DISABLED | ENABLED )] [-haMonitor ( ON |
      OFF )] [ ( ON | OFF )] [-tagall ( ON | OFF )] [-lacpMode <lacpMode
     >] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>]
     [-lacpTimeout (LONG | SHORT )] [-ifAlias <string>] [-throughput <
     positive_integer>][-bandwidthHigh <positive_integer> [-
     bandwidthNormal <positive_integer>]]
2 -  show interface [<id>]
3 <!--NeedCopy-->
```

Example:

```
1 > set interface 1/8 -duplex full
2  Done
3 <!--NeedCopy-->
```

To set the network interface parameters by using the GUI:

Navigate to **System > Network > Interfaces**, select the network interface that you want to modify (for example, 1/8), click **Edit**, and then set the parameters.

Setting the receive ring size and ring type for an interface

You can increase the receive ring size and ring type for IX, F1X, F2X, or F4X interfaces on Citrix ADC MPX and SDX platforms.

An increased ring size provides more cushion to handle burst traffic, but might impact the performance. A ring size of up to 8192 is supported for IX interfaces. A ring size of up to 4096 is supported for F1X, F2X, and F4X interfaces. The default ring size remains 2048.

Interface ring types are elastic by default. They increase or decrease in size based on packet arrival rate. You can configure the ring type as “fixed” in which case the ring size does not change based on traffic rate.

Note: This feature is supported from release 13.0 build 41.x, and supported on platforms that have IX, F1X, F2X, or F4X interfaces.

Use the `show hardware` command to identify whether your appliance has IX, F1X, F2X, or F4X interfaces.

Examples:

The following model has 16 F1X (10G) interfaces and 4 F4X (40G) interfaces.

```
> sh hardware
        Platform: NSMPX-25000-40G 20*CPU+16*F1X+4*F4X+2*E1K+2*CVM N3
            250040
        Manufactured on: 12/16/2016
        CPU: 2800MHZ
        Host Id: 234913926
        Serial no: N43RJCRV3X
        Encoded serial no: N43RJCRV3X
        Netscaler UUID: 336a32d6-2cfa-11e8-bf01-00e0ed5dd23c
        BMC Revision: 4.08
 Done
<!--NeedCopy-->
```

The following model has 2 1X (10G) interfaces.

```
> sh hardware
        Platform: NSMPX-10500 8*CPU+2*E1K+8*E1K+2*IX+8*CVM 1620 760100
        Manufactured on: 12/27/2010
        CPU: 2832MHZ
        Host Id: 1707114630
        Serial no: 7VZZV1ZXJ4
        Encoded serial no: 7VZZV1ZXJ4
        Netscaler UUID: eb1bfd72-5176-11e7-ba18-00e0ed1b0d12
 Done
<!--NeedCopy-->
```

To configure ring size and ring type by using the CLI
At the command line, type:


```
set interface <id> -ringsize <positive_integer> -ringtype ( Elastic |
    Fixed )
<!--NeedCopy-->
```

Parameters:

ringsize:

The receive ring size of the interface. A higher number provides more buffers to handle incoming traffic.

Default value: 2048
Minimum value: 512
Maximum value: 16384

ringtype:

The receive ring type of the interface. A fixed ring type preallocates the configured number of buffers irrespective of traffic rate. In contrast, an elastic ring, expands and shrinks based on incoming traffic rate.

Possible values: Elastic, Fixed

Default value: Elastic

Example:

```
> set interface 40/2 -ringsize 4096 -ringtype Fixed
 Done
> show interface 40/2

1)      Interface 40/2 (40G Ethernet, CR4, 40 Gbit) #21  flags=0xc020 <
    ENABLED, UP, UP, autoneg, HAMON, HEARTBEAT, 802.1q> MTU=1500, native
     vlan=10, MAC=00:e0:ed:75:14:2a, uptime 119h26m32s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
            throughput 0
        Actual: media UTP, speed 40000, duplex FULL, fctl OFF,
            throughput 40000
        LLDP Mode: NONE, LR Priority: 1024
        RX: Pkts(1443972660032) Bytes(1457207315336105) Errs(0) Drops
            (53319) Stalls(0)
        TX: Pkts(1452311431262) Bytes(1458534011197761) Errs(0) Drops
            (788) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
        Rx Ring: Configured size=4096, Actual size=4096, Type: Fixed
 Done
```



```
<!--NeedCopy-->
```

The last line shows the configured and actual ring size, and the ring type.

To configure ring size and ring type by using the GUI:

1. Navigate to **System > Network > Interfaces**.
2. Select your interface and click **Edit**.
3. In **Ring Size**, specify one of the following:
   - **IX interfaces:** 512, 1024, 2048, 4096, or 8192.
   - **F1X, F2X, or F4X interfaces:** 512, 1024, 2048, or 4096.
4. In **Ring Type**, select Elastic or Fixed.
5. Click **OK**.

Enable and disable network interfaces

By default, the network interfaces are enabled. Disable any network interface that is not connected to the network, so that it cannot send or receive packets. Disabling a network interface that is connected to the network in a high availability setup can cause a failover.

For more information about high availability, see High Availability.

To enable or disable a network interface by using the CLI:

At the command prompt, type:

```
-  enable interface <interface_num>
-  show interface <interface_num>
-  disable interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

Example:

```
> enable interface 1/8
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg,
           802.1q>
        MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
        Requested: media UTP, speed AUTO, duplex FULL, fctl OFF,
           throughput 0
```



```
        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
<!--NeedCopy-->
```

To enable or disable a network interface by using the GUI:

1. Navigate to **System > Network > Interfaces**.
2. Select the network interface and, in the **Action** list, select Enable or Disable.

Reset network interfaces

Network interface settings control properties such as duplex and speed. To renegotiate the settings of a network interface, you must reset it.

To reset a network interface by using the CLI:

At the command prompt, type:

```
-  reset interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

Example:

```
> reset interface 1/8
 Done
<!--NeedCopy-->
```

To reset a network interface by using the GUI:

1. Navigate to **System > Network > Interfaces**.
2. Select the network interface and, in the **Action** list, select **Reset Interface**.

Monitor a network interface

You can display network interface statistics to monitor parameters and use the information to check the health of the network interface. You can monitor parameters, such as packets sent and packets received, throughput, Link Aggregate Control Protocol (LACP) data units, and errors. You can clear the statistics of a network interface to monitor its statistics from the time the statistics are cleared.

To display the statistics of the network interfaces by using the CLI:

At the command prompt, type:


```
-  stat interface <interface_num>
<!--NeedCopy-->
```

Example:

```
> stat interface 1/8
 Done
<!--NeedCopy-->
```

To clear a network interface's statistics by using the CLI:

At the command prompt, type:

```
-  clear interface <interface_num>
<!--NeedCopy-->
```

Example:

```
> clear interface 1/8
 Done
<!--NeedCopy-->
```

To display the statistics of an Interface by using the GUI:

Navigate to **System > Network > Interfaces**, select the network interface, and click **Interface Statistics**.

To clear a network interface's statistics by using the GUI:

1. Navigate to **System > Network > Interfaces**.
2. Select the network interface and, in the **Action** list, select **Clear Statistics**.

Configuring Forwarding Session Rules

September 14, 2021

By default, the Citrix ADC appliance does not create session entries for traffic that it only forwards (L3 mode). For a case in which a client request that the appliance forwards to a server results in a response that has to return by the same path, you can create a forwarding-session rule. A forwarding-session rule creates forwarding-session entries for traffic that originates from or is destined for a particular network and is forwarded by the Citrix ADC. You can create forwarding session rules for IPv4 traffic as well as IPv6 traffic.

When configuring an IPv4 forwarding-session rule, you can specify either an IPv4 network address or an extended ACL as the condition for identifying IPv4 traffic for which to create a forwarding-session entry:

- **Network address**. When you specify an IPv4 network address, the appliance creates forwarding sessions for IPv4 traffic whose source or destination matches the network address.
- **Extended ACL rule**. When you specify an extended ACL rule, the appliance creates forwarding sessions for IPv4 traffic that matches the conditions specified in the extended ACL rule.

When configuring an IPv6 forwarding-session rule, you can specify either an IPv6 prefix or an ACL6 as the condition for identifying IPv6 traffic for which to create a forwarding-session entry:

- **IPv6 prefix**. When you specify an IPv6 prefix, the appliance creates forwarding sessions for IPv6 traffic whose source or destination matches the IPv6 prefix.
- **ACL6 rule**. When you specify an ACL6 rule, the appliance creates forwarding sessions for IPv6 traffic that matches the conditions specified in the ACL6 rule.

To create an IPv4 forwarding session rule by using the CLI:

At the command prompt, type the following commands to create a forwarding-session rule and verify the configuration:

- add forwardingSession <name> [<network> <netmask>] | [-aclname <string>] -connfailover (ENABLED | DISABLED)
- show forwardingSession

Example:

```
A network address as the condition:

> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done

An ACL as the condition:

> add forwardingSession fs-acl-1 acl1
 Done
<!--NeedCopy-->
```

To configure an IPv4 forwarding session rule by using the GUI:

Navigate to System > Network > Forwarding Sessions, add a new IPv4 forwarding session, or edit an existing forwarding session.

To create an IPv6 forwarding session rule by using the CLI:

- At the command prompt, type the following commands to create a forwarding-session rule and verify the configuration:

- add forwardingSession <name> [<IPv6 prefix>] | [-acl6name <string>]
- show forwardingSession

Example:

```
    An IPv6 prefix as the condition:

    > add forwardingSession fsv6-pfx-1 3ffe::/64
     Done

    An ACL6 rule as the condition:

    > add forwardingSession fsv6-acl6-1  -acl6name ACL6-FS
     Done
<!--NeedCopy-->
```

To configure an IPv6 forwarding session rule by using the GUI:

Navigate to System > Network > Forwarding Sessions, add a new IPv6 forwarding session, or edit an existing forwarding session.

Assigning an ACL rule to an Existing Forwarding Session Rule

You can assign an ACL rule to a Network-address/IPv6-prefix based forwarding session rule, in which case it becomes an ACL based forwarding session rule. You can also change an existing ACL rule to another ACL rule in an ACL based forwarding session rule. After the existing related forwarding session entries (if any) have timed out, the rules start using the newly assigned ACL to identify IPv4/IPv6 traffic for which to create a forwarding-session entry.

To assign an extended ACL rule to an existing IPv4 forwarding session rule by using the CLI:

At the command prompt, type

- `set forwardingSession <name> [-aclname <string>]`
- `show forwardingSession <name>`

To assign an ACL6 rule to an existing IPv6 forwarding session rule by using the CLI:

At the command prompt, type

- `set forwardingSession <name> [-acl6name <string>]`
- `show forwardingSession <name>`

Example:

```
> add forwardingSession FS-1 -aclname ACL-9
Done

```



```
> add forwardingSession FS6-1  -acl6name ACL6-9
Done
```

Disabling Steering for Forwarding Sessions on a Cluster Setup

The default behavior of a Citrix ADC cluster is for the node that receives traffic (flow receiver) to direct the traffic to another node (flow processor), which processes the traffic. Directing the traffic from flow receiver to flow processor occurs over the cluster backplane and is called steering.

Steering can be an overhead for real-time processing or when the setup includes high-latency links.

Steering for forwarding sessions can now be disabled so that the processing becomes local to the flow receiver. That is, the flow receiver becomes the flow processor.

Before you begin

Note the following points before configuring forwarding session rules in a cluster setup:

- You must configure linksets to be used for forwarding sessions.
- You must enable MAC Based Forwarding (MBF) on the cluster setup.

Configuring Forwarding Session Rules in a Cluster Setup

Disabling steering for forwarding session rules in a cluster setup can be done at the following two levels:

- **Specific forwarding session rule level**. Enable the Process Local parameter while adding a new forwarding session rule or editing an existing forwarding session rule.
- **Global level**. Enable the Process Local parameter while adding a new cluster instance or editing an existing cluster instance. The global setting takes precedence over the forwarding session rule setting.

CLI procedures

To disable steering for a forwarding session rule on a cluster setup by using the CLI:

At the command prompt, type one of the following sets of commands:

- If adding a new forwarding session rule:
  - **add forwardingSession** <name> ((<network> [<netmask>]) | -**acl6name** <string> | -**aclname** <string>) -**processLocal ENABLED**
  - **show forwardingSession** <name>
- If reconfiguring an existing forwarding session rule:

- **set forwardingSession** <name> -**processLocal ENABLED**
- **show forwardingSession** <name>

To disable steering for all (global level) forwarding session rules on a cluster setup by using the CLI:

At the command prompt, type one of the following sets of commands:

- If adding a new cluster instance:
  - **add cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>
- If reconfiguring an existing cluster instance:
  - **set cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>

Sample configuration:

Following are two examples of disabling steering at the forwarding session rule level, and an example of disabling steering at the global level.

```
An IPv4 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV4-1 10.102.105.51
    255.255.255.255 -processLocal Enabled
Done

An IPv6 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV6-1  -acl6name ACL6-
    FWD-SESSN-1 -processLocal Enabled
Done

A cluster setup, with an instance ID 10, has steering disabled at
    global level:

> set cluster instance 10 -processLocal Enabled
Done
<!--NeedCopy-->
```

GUI procedures

To disable steering for a forwarding session rule on a cluster setup by using the GUI:

Navigate to **System** > **Network** > **Forwarding Sessions**, select **Process Local** while adding a new forwarding session rule or editing an existing forwarding session rule.

To disable steering for all (global level) forwarding session rules on a cluster setup by using the GUI:

Navigate to **System** > **Cluster**, and select **Process Local** while adding a cluster configuration or modifying an existing cluster configuration.

Understanding VLANs

September 14, 2021

A Citrix ADC appliance supports Layer 2 port and IEEE 802.1q tagged VLANs. VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface as a part of multiple VLANs by using IEEE 802.1q tagging.

You can configure VLANs and bind them to IP subnets. The Citrix ADC then performs IP forwarding between these VLANs (if it is configured as the default router for the hosts on these subnets).

The Citrix ADC supports the following types of VLANs:

- **Port-Based VLANs.** The membership of a port-based VLAN is defined by a set of network interfaces that share a common, exclusive Layer 2 broadcast domain. You can configure multiple port-based VLANs. By default, all network interfaces on the Citrix ADC are members of VLAN 1.

  If you apply 802.1q tagging to the port, the network interface belongs to a port-based VLAN. Layer 2 traffic is bridged within a port-based VLAN, and Layer 2 broadcasts are sent to all members of the VLAN if Layer 2 mode is enabled. When you add an untagged network interface as a member of a new VLAN, it is removed from its current VLAN.

- **Default VLAN.** By default, the network interfaces on the Citrix ADC are included in a single, port-based VLAN as untagged network interfaces. This VLAN is the default VLAN. It has a VLAN ID (VID) of 1. This VLAN exists permanently. It cannot be deleted, and its VID cannot be changed.

  When you add a network interface to a to a different VLAN as an untagged member, the network interface is automatically removed from the default VLAN. If you unbind a network interface from its current port-based VLAN, it is added to the default VLAN again.

- **Tagged VLANs.** 802.1q tagging (defined in the IEEE 802.1q standard) allows a networking device (such as the Citrix ADC) to add information to a frame at Layer 2 to identify the VLAN membership of the frame. Tagging allows network environments to have VLANs that span multiple devices. A device that receives the packet reads the tag and recognizes the VLAN to which the frame belongs. Some network devices do not support receiving both tagged and untagged packets on the same network interface—in particular, Force10 switches. In such cases, you need to contact customer support for assistance.

  The network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of one VLAN only (its native VLAN). This network interface transmits the

frames for the native VLAN as untagged frames. A network interface can be a part of more than one VLAN if the other VLANs are tagged.

When you configure tagging, be sure to match the configuration of the VLAN on both ends of the link. The port to which the Citrix ADC connects must be on the same VLAN as the Citrix ADC network interface.

Note: This VLAN configuration is neither synchronized nor propagated, therefore you must perform the configuration on each unit in an HA pair independently.

Applying Rules to Classify Frames

VLANs have two types of rules for classifying frames:

- **Ingress rules.** Ingress rules classify each frame as belonging only to a single VLAN. When a frame is received on a network interface, the following rules are applied to classify the frame:
  - If the frame is untagged, or has a tag value equal to 0, the VID of the frame is set to the port VID (PVID) of the receiving interface, which is classified as belonging to the native VLAN. (PVIDs are defined in the IEEE 802.1q standard.)
  - If frame has a tag value equal to FFF, the frame is dropped.
  - If the VID of the frame specifies a VLAN of which the receiving network interface is not a member, the frame is dropped. For example, if a packet is sent from a subnet associated with VLAN ID 12 to a subnet associated with VLAN ID 10, the packet is dropped. If an untagged packet with VID 9 is sent from the subnet associated with VLAN ID 10 to a network interface PVID 9, the packet is dropped.
- **Egress Rules.** The following egress rules are applied:
  - If the VID of the frame specifies a VLAN of which the transmission network interface is not a member, the frame is discarded.
  - During the learning process (defined by the IEEE 802.1q standard), the Src MAC and VID are used to update the bridge lookup table of the Citrix ADC.
  - A frame is discarded if its VID specifies a VLAN that does not have any members. (You define members by binding network interfaces to a VLAN.)

VLANs and Packet Forwarding on the Citrix ADC

The forwarding process on the Citrix ADC appliance is similar to that on any standard switch. However, the Citrix ADC performs forwarding only when Layer 2 mode is on. The key features of the forwarding process are:

- Topology restrictions are enforced. Enforcement involves selecting each network interface in the VLAN as a transmission port (depending on the state of the network interface), bridging restrictions (do not forward on the receiving network interface), and MTU restrictions.

- Frames are filtered on the basis of information in the bridge table lookup in the forwarding database (FDB) table of the Citrix ADC. The bridge table lookup is based on the destination MAC and the VID. Packets addressed to the MAC address of the Citrix ADC are processed at the upper layers.
- All broadcast and multicast frames are forwarded to each network interface that is a member of the VLAN, but forwarding occurs only if L2 mode is enabled. If L2 mode is disabled, the broadcast and multicast packets are dropped. This is also true for MAC addresses that are not currently in the bridging table.
- A VLAN entry has a list of member network interfaces that are part of its untagged member set. When forwarding frames to these network interfaces, a tag is not inserted in the frame.
- If the network interface is a tagged member of this VLAN, the tag is inserted in the frame when the frame is forwarded.

When a user sends any broadcast or multicast packets without the VLAN being identified, that is, during duplicate address detection (DAD) for NSIP or ND6 for the next hop of the route, the packet is sent out on all the network interfaces, with appropriate tagging based on either the Ingress and Egress rules. ND6 usually identifies a VLAN, and a data packet is sent on this VLAN only. Port-based VLANs are common to IPv4 and IPv6. For IPv6, the Citrix ADC supports prefix-based VLANs.

Configuring a VLAN

September 14, 2021

You can implement VLANs in the following environments:

- Single subnet
- Multiple subnets
- Single LAN
- VLANs (no tagging)
- VLANs (802.1q tagging)

If you configure VLANs that have only untagged network interfaces as their members, the total number of possible VLANs is limited to the number of network interfaces available in the Citrix ADC. If more IP subnets are required with a VLAN configuration, 802.1q tagging must be used.

When you bind a network interface to a VLAN, the network interface is removed from the default VLAN. If the network interfaces need to be a part of more than one VLAN, you can bind the network interfaces to the VLANs as tagged members.

You can configure the Citrix ADC to forward traffic between VLANs at Layer 3. In this case, a VLAN is associated with a single IP subnet. The hosts in a VLAN that belong to a single subnet use the same subnet mask and one or more default gateways connected to that subnet. Configuring Layer 3 for a

VLAN is optional. Layer 3 is used for IP forwarding (inter-VLAN routing). Each VLAN has a unique IP address and subnet mask that define an IP subnet for the VLAN. In an HA configuration, this IP address is shared with the other Citrix ADC appliances. The Citrix ADC forwards packets between configured IP subnets (VLANs).

When you configure the Citrix ADC, you must not create overlapping IP subnets. Doing so impedes Layer 3 functionality.

Each VLAN is a unique Layer 2 broadcast domain. Two VLANs, each bound to separate IP subnets, cannot be combined into a single broadcast domain. Forwarding traffic between two VLANs requires a Layer 3 forwarding (routing) device, such as the Citrix ADC appliance.

Configuring VLANs in an HA Setup

VLAN configuration for a high-availability setup requires that the Citrix ADC appliances have the same hardware configuration, and the VLANs configured on them must be mirror images.

The correct VLAN configuration is implemented automatically when the configuration is synchronized between the Citrix ADC appliances. The result is identical actions on all the appliances. For example, adding network interface 0/1 to VLAN2 adds this network interface to VLAN 2 on all the appliances participating in the high-availability setup.

Note: If you use network-interface-specific commands in an HA setup, the configurations you create are not propagated to the other Citrix ADC appliance. You must perform these commands on each appliance in an HA pair to ensure that the configuration of the two appliances in the HA pair remains synchronized.

Creating or Modifying a VLAN

To configure a VLAN, you create a VLAN entity, and then bind network interfaces and IP addresses to the VLAN. If you remove a VLAN, its member interfaces are added to the default VLAN.

CLI procedures

To create a VLAN by using the CLI:

At the command prompt, type:

- `add vlan <id> [-aliasName <string>] [-ipv6DynamicRouting (ENABLED|DISABLED)]`
- `sh vlan <id>`

Example:


```
  > add vlan 2  -aliasName "Network A"   Done
<!--NeedCopy-->
```

To bind an interface to a VLAN by using the CLI:

At the command prompt, type:

- `bind vlan <id> -ifnum <slot/port>`
- `sh vlan <id>`

Example:

```
  > bind vlan 2 -ifnum 1/8   Done
<!--NeedCopy-->
```

To bind an IP address to a VLAN by using the CLI:

At the command prompt, type:

- `bind vlan <id> -IPAddress <IPAddress> <netMask>`
- `sh vlan <id>`

Example:

```
  > bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0   Done
<!--NeedCopy-->
```

To remove a VLAN by using the CLI:

At the command prompt, type:

- `rm vlan <id>`

GUI procedures

To configure a VLAN by using the GUI:

1. Navigate to System > Network > VLANs, add a new VLAN, or edit an existing VLAN.
2. To bind an IP address to a VLAN, under IP Bindings, select the Active option corresponding to the IP address that you want to bind to the VLAN (for example, 10.102.29.54). The Type column displays the IP address type (such as mapped IP, virtual IP, or subnet IP) for each IP address in the IP Address column.
3. To bind a network interface to a VLAN, under Interface Bindings, select the Active option corresponding to the interface that you want to bind to the VLAN.

Monitoring VLANs

You can display VLAN statistics such as packets received, bytes received, packets sent, and bytes sent, and use the information to identify anomalies and or debug a VLAN.

To view the statistics of a VLAN by using the CLI:

At the command prompt, type:

- `stat vlan <vlanID>`

Example:

```
1   stat vlan 2
2 <!--NeedCopy-->
```

To view the statistics of a VLAN by using the GUI:

1. Navigate to System > Network > VLANs.
2. Select the VLAN, and click Statistics.

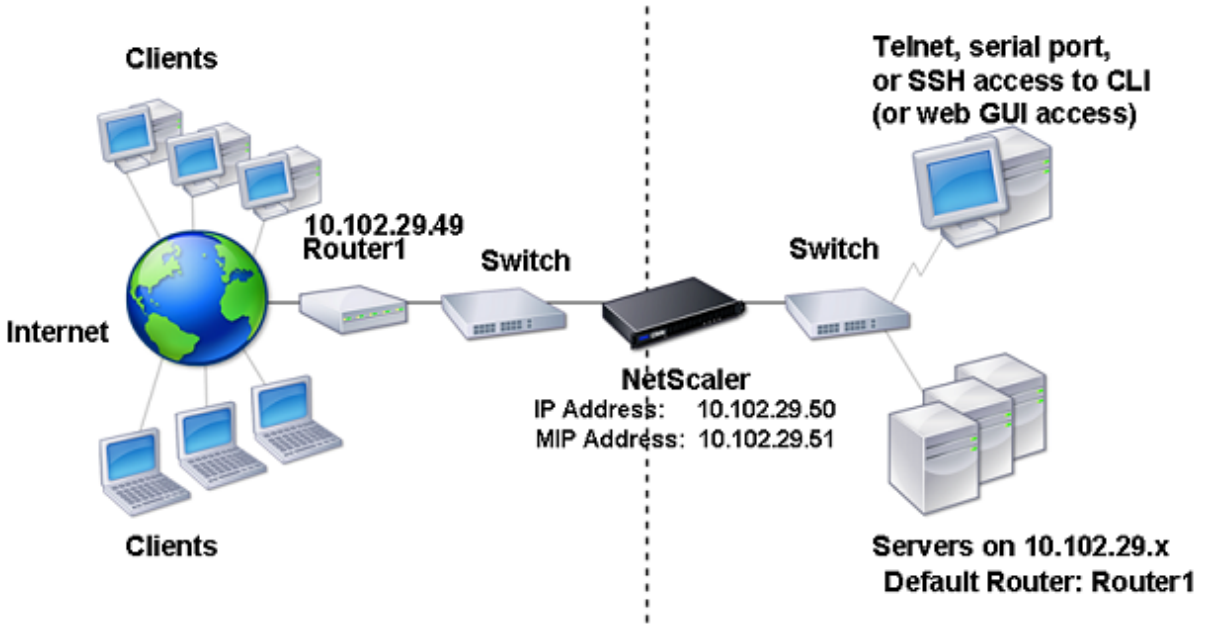
Configuring VLANs on a Single Subnet

September 14, 2021

Before configuring a VLAN on a single subnet, make sure that Layer 2 Mode is enabled.

The following figure shows a single subnet environment

Figure 1. VLAN on a Single Subnet



In the above figure:

1. The default router for the Citrix ADC and the servers is Router 1.
2. Layer 2 mode must be enabled on the Citrix ADC for the Citrix ADC to have direct access to the servers.
3. For this subnet, a virtual server can be configured for load balancing on the Citrix ADC appliance.

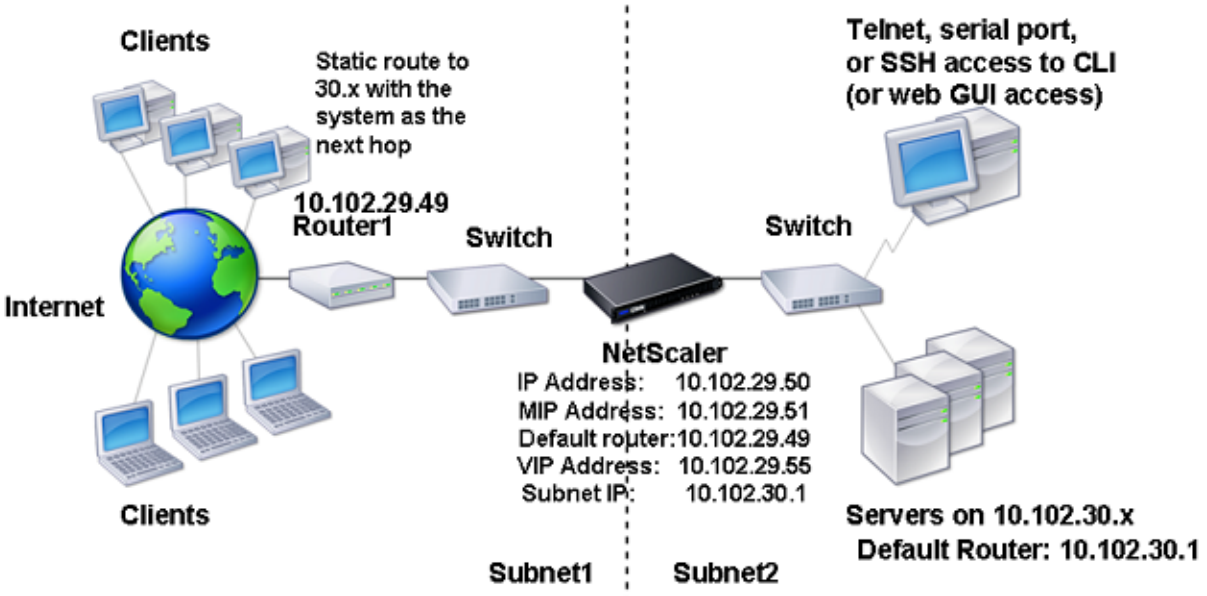
To configure a VLAN on a single subnet, follow the procedures described in Configuring a VLAN.

Configuring VLANs on Multiple Subnets

September 14, 2021

To configure a single VLAN across multiple subnets, you must add a VIP for the VLAN and configure the routing appropriately. The following figure shows a single VLAN configured across multiple subnets.

Figure 1. Multiple Subnets in a Single VLAN



To configure a single VLAN across multiple subnets, perform the following tasks:

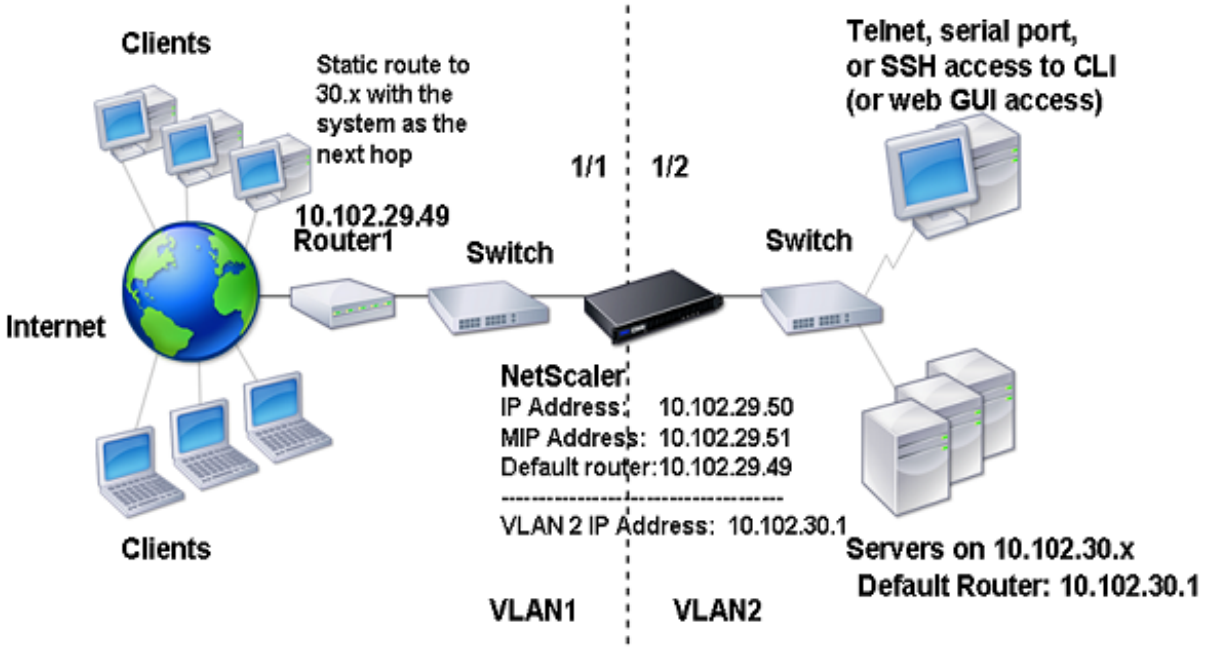
1. Disable Layer 2 mode. For the procedure to disable Layer 2 mode, see Packet forwarding modes.
2. Add a VIP address. For the procedure to add a VIP address, see Configuring and Managing Virtual IP Addresses (VIPs).
3. Configure RNAT rule. For the procedure to configure the RNAT ID, see Configuring RNAT.

Configuring Multiple Untagged VLANs across Multiple Subnets

September 14, 2021

In environments with multiple untagged VLANs across multiple subnets, a VLAN is configured for each IP subnet. A network interface is bound to one VLAN only. The following figure shows this configuration.

Figure 1. Multiple Subnets with VLANs - No Tagging



To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.
2. Bind the 1/2 network interface of the Citrix ADC to VLAN 2 as an untagged network interface.
3. Bind the IP address and subnet mask to VLAN 2.

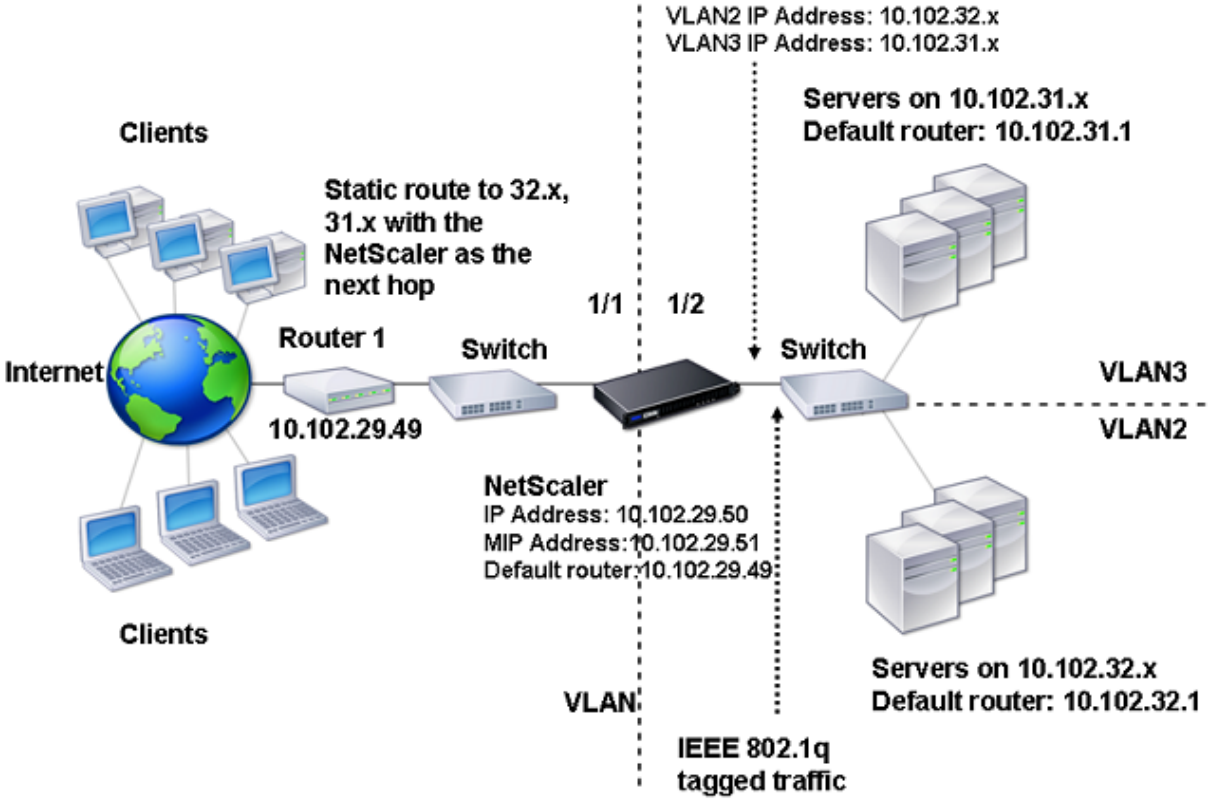
For procedures on these tasks, see Configuring a VLAN.

Configuring Multiple VLANs with 802.1q Tagging

September 14, 2021

For multiple VLANs with 802.1q tagging, each VLAN is configured with a different IP subnet. Each network interface is in one VLAN. One of the VLANs is set up as tagged. The following figure shows this configuration.

Figure 1. Multiple VLANs with IEEE 802.1q Tagging



To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.
2. Bind the 1/2 network interface of the Citrix ADC to VLAN 2 as an untagged network interface.
3. Bind the IP address and netmask to VLAN 2.
4. Add VLAN 3.
5. Bind the 1/2 network interface of the Citrix ADC to VLAN 3 as a tagged network interface.
6. Bind the IP address and netmask to VLAN 3.

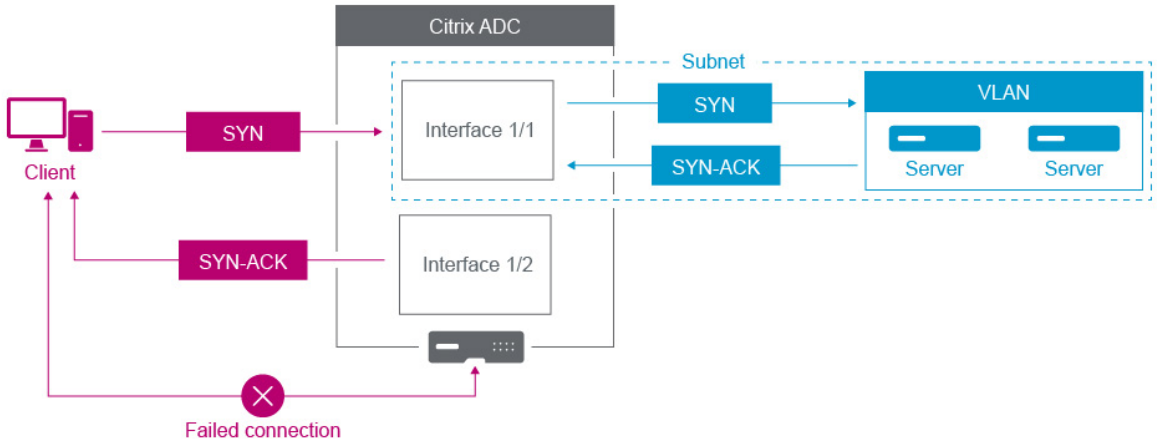
For procedures on these tasks, see Configuring a VLAN.

Associate an IP Subnet with a Citrix ADC Interface by Using VLANs

September 14, 2021

BY default, a Citrix ADC appliance does not provide any differentiation between network interfaces. The appliance functions more like a network hub than a switch. This can lead to Layer 3 network loops where duplicated traffic is transmitted on multiple interfaces.

In such scenarios, depending on network design, it is possible that a request could be transmitted on one interface and the corresponding response is received on a different interface.



For example, a SYN packet sent on one interface and the SYN-ACK response received on a different interface can result in a failed connection, as the appliance expects to receive the SYN-ACK on the same interface that sent the original SYN packet.

To resolve such issues, the appliance can use internal or external VLANs, to associate specific subnets with interfaces.

Before you begin

Before you begin associating an IP Subnet with a Citrix ADC interface by using VLANs, note the following points:

- The network connectivity might be accidentally lost when associating a VLAN to the subnet or interface that is currently being used to access the Citrix ADC GUI or command line interface. Therefore, in such scenarios, it is highly recommended that the change is made by accessing the command line interface through the serial console of a physical Citrix ADC appliance or through the virtual serial console of a Citrix ADC VPX.
- Citrix ADC management interfaces lack certain hardware optimization features which makes them less desirable for use with production data traffic. As such, it is recommended that the Citrix ADC be configured to only use the management interfaces for management (NSIP) traffic. In the default configuration, there is no logical differentiation between the management interfaces and data interfaces on a hardware NetScaler. In order to accomplish this goal, it is recommended that the NSIP be on a separate VLAN from data traffic, which allows management traffic to be on a separate interface.

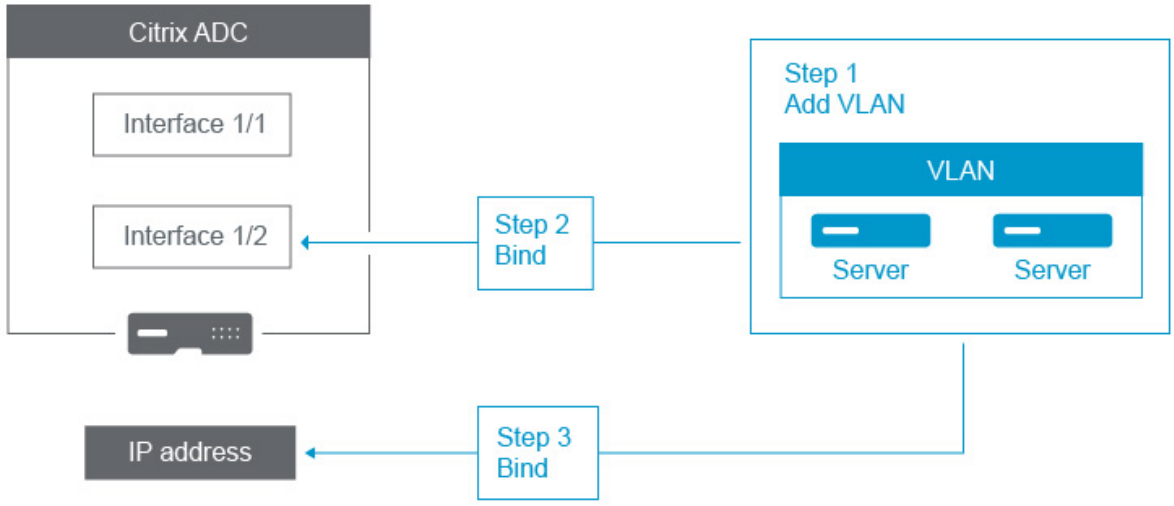
  Although the concept is the same, to change the VLAN associations of the subnet that contains the NSIP address, you must configure NSVLAN instead of the below instructions. Such changes

will also require a reboot of the Citrix ADC in order to take effect. For more information, see Configuring NSVLAN.

- On Citrix ADC SDX, it is highly recommended that the NSIP of each instance be on the same subnet and VLAN as the SDX's SVM (Management Service GUI) and XenServer. The SVM communicates with instances via the network. If the SVM, XenServer, and instances are not on the same VLAN and subnet, management traffic must flow outside of the SDX. In this situation, network issues can cause Instance state to show up as yellow or red and can prevent the management and configuration changes of the Citrix ADC instances.

Configuration Steps

Associate an IP subnet with a Citrix ADC Interface consists of the following tasks:



Add a VLAN. While adding a VLAN, if you are tagging the VLAN, then you must select a VLAN number that is defined in the network switch for the associated switch port. If the VLAN is untagged and is internal to the appliance, then it is recommended that you select the VLAN number that is available in the switch configuration for easy reference.

Bind an interface to the VLAN. While binding, if you are using Link Aggregation, associate the VLAN with the LA channel (for example, LA/1) instead of the physical interface. The VLAN must be associated with only one network interface.

If you want to tag the traffic on the interface, use the tagged (Tag) option. Else, the traffic leaves the appliance untagged and is associated with the native VLAN of the switch port.

Bind an IP address to the VLAN. While binding, if you bind more than one IP address from the same subnet, an error occurs. When an IP address is associated with a VLAN, all IP addresses in that subnet are automatically associated with the VLAN.

> **Note:**
>
> In a high availability (HA) setup, these VLAN configurations are added automatically from the primary node to the secondary node during HA synchronization. For more information about high availability setups, see High Availability.

CLI procedures

To add a VLAN by using the CLI:

At the command prompt, type:

- **add vlan** <id>
- **sh vlan** <id>

To bind an interface to a VLAN by using the CLI:

At the command prompt, type:

- **bind vlan** <id> -**ifnum** <slot/port>
- **sh vlan** <id>

To bind an IP address to a VLAN by using the CLI:

At the command prompt, type:

- **bind vlan** <id> -**IPAddress** <IPAddress> <netMask>
- **sh vlan** <id>

Example:

```
> add vlan 100

> bind vlan 100 -ifnum 1/1

> bind vlan 100 -ipAddress 10.0.1.0 255.255.255.0
<!--NeedCopy-->
```

GUI procedures

To configure a VLAN by using the GUI:

1. Navigate to **System** > **Network** > **VLANs**, add a new VLAN.
2. To bind a network interface to a VLAN, under **Interface Bindings**, select the **Active** option corresponding to the interface that you want to bind to the VLAN.

3. To bind an IP address to a VLAN, under **IP Bindings**, select the **Active** option corresponding to the IP address that you want to bind to the VLAN (for example, 10.102.29.54). The **Type** column displays the IP address type for each IP address in the **IP Address** column.

Citrix ADC appliance networking and VLAN best practices

September 14, 2021

A Citrix ADC appliance uses VLANs to determine which interface must be used for which traffic. In addition, Citrix ADC appliance does not participate in Spanning Tree. Without the proper VLAN configuration, the Citrix ADC appliance is unable to determine which interface to use, and it can function more like a HUB than a switch or a router. In other words, the Citrix ADC appliance can use all interfaces for each conversation.

Symptoms of VLAN misconfiguration

VLAN misconfiguration issue can manifest itself in many forms, including performance issues, inability to establish connections, randomly disconnected sessions, and in severe situations, network disruptions seemingly unrelated to the Citrix ADC appliance itself. The Citrix ADC appliance may also report MAC moves, muted interfaces, and/or management interface transmit or receive buffer overflows, depending on the exact nature of the interaction with your network.

MAC Moves (counter nic_tot_bdg_mac_moved): This issue indicates that the Citrix ADC appliance is using more than one interface to communicate with the same device (MAC address), because it could not properly determine which interface to use.

Muted interfaces (counter nic_err_bdg_muted): This issue indicates that the Citrix ADC appliance has detected that it is creating a routing loop due to VLAN configuration issues, and as such, it has shut down one or more of the offending interfaces in order to prevent a network outage.

Interface buffer overflows, typically referring to management interfaces (counter nic_err_tx_overflow):This issue can be caused if too much traffic is being transmitted over a management interface. Management interfaces on the Citrix ADC appliance is not designed to handle large volumes of traffic, which may result from network and VLAN misconfigurations triggering the Citrix ADC appliance to use a management interface for production data traffic. This often occurs because the Citrix ADC appliance has no way to differentiate traffic on the VLAN / subnet of the NSIP (NSVLAN) from regular production traffic. It is highly recommended that the NSIP be on a separate VLAN and subnet from any production devices such as workstations and servers.

Orphan ACKs (counter tcp_err_orphan_ack): This issue indicates that the Citrix ADC appliance received an ACK packet that it was not expecting, typically on a different interface than the ACK'd traffic

originated from. This situation can be caused by VLAN misconfigurations where the Citrix ADC appliance transmits on a different interface than the target device would typically use to communicate with the Citrix ADC appliance (often seen in conjunction with MAC moves)

High rates of retransmissions or retransmit give ups (counters: tcp_err_retransmit_giveups, tcp_err_7th_retransmit, various other retransmit counters): The Citrix ADC appliance attempts to retransmit a TCP packet a total of 7 times before it gives up and terminates the connection. While this situation can be caused by network conditions, it often occurs as a result of VLAN and interface misconfiguration.

High Availability Split Brain: Split Brain is a condition where both high availability nodes believe they are Primary, leading to duplicate IP addresses and loss of Citrix ADC appliance functionality. This is caused when the two high availability nodes cannot communicate with each-other using high availability Heartbeats on UDP Port 3003 using the NSIP, across any interface. This is typically caused by VLAN misconfigurations where the native VLAN on the Citrix ADC appliance interfaces does not have connectivity between Citrix ADC appliances.

Best practices for VLAN and network configurations

1. Each subnet must be associated with a VLAN.
2. More than one subnet can be associated with the same VLAN (depending on your network design).
3. Each VLAN should be associated to only one interface (for purposes of this discussion, a LA channel counts as a single interface).
4. If you require more than one subnet to be associated with an interface, the subnets must be tagged.
5. Contrary to popular belief, the Mac-Based-Forwarding (MBF) feature on the Citrix ADC appliance is not designed to mitigate this type of issue. MBF is designed primarily for the DSR (Direct Server Return) mode of the Citrix ADC appliance, which is rarely used in most environments (it is designed to allow traffic to purposely bypass the Citrix ADC appliance on the return path from the back-end servers). MBF may hide VLAN issues in some instances, but it should not be relied-upon to resolve this type of problem.
6. Every interface on Citrix ADC appliance requires a native VLAN (unlike Cisco, where native VLANs are optional), although the TagAll setting on an interface can be used so that no untagged traffic leaves the interface in question.
7. The native VLAN can be tagged if necessary for your network design (this is the TagAll option for the interface).
8. The VLAN for the subnet of your Citrix ADC appliance's NSIP is a special case. This is called the NSVLAN. The concepts are the same but the commands to configure it are different and changes

to the NSVLAN require a reboot of the Citrix ADC appliance to take effect. If you attempt to bind a VLAN to a SNIP that shares he same subnet as the NSIP, you get "Operation not permitted." This is because you have to use the NSVLAN commands instead. Also, on some firmware versions, you cannot set an NSVLAN if that VLAN number exists using `add VLAN` command. Simply remove the VLAN and then set the NSVLAN again.

9. High availability Heartbeats always use the Native VLAN of the respective interface (optionally tagged if the TagAll option is set on the interface).

10. There must be communication between at least one set of Native VLAN(s) on the two nodes of an high availability pair (this can be direct or via a router). The native VLANs are used for high availability heartbeats. If the Citrix ADC appliances cannot communicate between native VLANs on any interface, this will lead to high availability failovers and possibly a split-brain situation where both Citrix ADC appliances think they are primary (leading to duplicate IP addresses, amongst other things).

11. The Citrix ADC appliance does not participate in spanning tree. As such, it is not possible to use spanning tree to provide for interface redundancy when using a Citrix ADC appliance. Instead, use a form of Link Aggregation (LACP or manual LAG) for this purpose.

    Note: If you want to have link aggregation between multiple physical switches, you must have the switches configured as a virtual switch, using a feature such as Cisco's Switch Stack.

12. The high availability synchronization and command Propagation, by default, use the NSIP/NSVLAN. To separate these out to a different VLAN, you can use the SyncVLAN option of the `set HA node` command.

13. There is nothing built-in to the Citrix ADC appliance default configuration that denotes that a management interface (0/1 or 0/2) is restricted to management traffic only. This restriction must be enforced by the end user through VLAN configuration. The management interfaces are not designed to handle data traffic, so your network design must take this point into account. Management interfaces, contained on the Citrix ADC appliance motherboard, lack various offloading features such as CRC offload, larger packet buffers, and other optimizations, making them much less efficient in handling large amounts of traffic. To separate production data and management traffic, the NSIP must not be on the same subnet/VLAN as your data traffic.

14. If it is desired to use a management interface to carry management traffic, it is best practice that the Default Route be on a subnet other than the subnet of the NSIP (NSVLAN).

    In many configurations, the default route is relied-upon for workstation commmunication (in an internet scenario). If the default route is on the same subnet as the NSIP, then the ADC appliance can use the management interface to send and receive data traffic. This use of data traffic can overload the management interface.

15. Also, an SDX-the SVM, XenServer, and all Citrix ADC instance NSIPs must be on the same VLAN and subnet. There is no **backplane** in the SDX appliance that allows for communication be-

tween SVM/Xen/Instances. If they are not on the same VLAN/subnet/interface, traffic between them must leave the physical hardware, be routed on your network, and return.

This configuration can lead to obvious connectivity issues between the instances and SVM and as such, is not recommended. A common symptom of this is a Yellow Instance State indicator in the SVM for the VPX instance in question, and the inability to use the SVM to reconfigure a VPX instance.

16. If some VLANs are bound to subnets and some are not, during a high availability failover, GARP packets are not be sent for any IP addresses on any of the subnets that are not bound to a VLAN. This configuration can cause dropped connections and connectivity issues during high availability failovers. This issue is caused because the Citrix ADC appliance cannot notify the network MAC ownership IP addresses change on non-VMAC-configured Citrix ADC appliances.

    Symptoms of this are that during/after a high availability failover, the ip_tot_floating_ip_err counter increments on the former primary Citrix ADC appliance for more than a few seconds, indicating that the network did not receive or process GARP packets and the network is continuing to transmit data to the new secondary Citrix ADC appliance.

Configuring NSVLAN

September 14, 2021

NSVLAN is a VLAN to which the Citrix ADC management IP (NSIP) address's subnet is bound. The NSIP subnet is available only on interfaces that are associated with NSVLAN. By default, NSVLAN is VLAN 1, but you can designate a different VLAN as NSVLAN. If you do so, you must reboot the Citrix ADC appliance for the change to take effect. After the reboot, NSIP subnet traffic is restricted to the new NSVLAN.

The traffic from the Citrix ADC IP subnet can be tagged (802.1q) with the VLAN ID specified for NSVLAN. You must configure the attached switch interface to tag and allow this same VLAN ID on the connected interface. If you remove your NSVLAN configuration, the NSIP subnet is automatically bound to VLAN 1, restoring the default NSVLAN.

To configure NSVLAN by using the CLI:

At the command prompt, type:

- set ns config -nsvlan <positive_integer> -ifnum <interface_name> … [-tagged (YES|NO)]
- show ns config

  Note: The configuration takes effect after the Citrix ADC appliance is rebooted.

Example:


```
> set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged NO
 Done

> save config
 Done
<!--NeedCopy-->
```

To restore the default NSVLAN configuration by using the CLI:

At the command prompt, type:

- unset ns config -nsvlan
- show ns config

Example:

```
> unset ns config -nsvlan
 Done
<!--NeedCopy-->
```

To configure NSVLAN by using the GUI:

Navigate to System > Settings, in the Settings group, click Change NSVLAN Settings.

Setting the MTU on the NSVLAN

By default, the MTU of the NSVLAN is set to 1500 bytes. You can modify this setting to optimize throughput and network performance. For example, you can configure the NSVLAN to process jumbo frames.

To set the MTU of the NSVLAN by using the CLI:

At the command prompt, type:

- **set vlan** <id> **-mtu** <positive_integer>
- **show vlan** <id>

To set the MTU of the NSVLAN by using the GUI:

Navigate to **System** > **Network** > **VLANs**, open the NSVLAN, and set the **Maximum Transmission Unit** parameter.

Sample configuration:

In the following sample configuration, VLAN 100 is the NSVLAN.

```
> set ns config -nsvlan 100 -ifnum 1/1 -tagged no

  Warning: The configuration must be saved and the system rebooted for
     these settings to take effect
```



```
4
5  > set vlan 100 -mtu 1600
6
7    Done
8
9  > sh vlan
10
11   1)  VLAN ID: 1
12
13       Link-local IPv6 addr:
14     fe80::947b:52ff:fead:12d5/64
15
16       Interfaces : 1/2 LO/1
17
18   2)  VLAN ID: 100    VLAN Alias Name:
19
20       MTU: 1600
21
22       Interfaces : 1/1
23
24       IPs :
25
26           10.102.53.114     Mask: 255.255.255.0
27
28   Done
29
30 > save config
31
32   Done
33 <!--NeedCopy-->
```

Configuring Allowed VLAN List

September 14, 2021

Citrix ADC accepts and sends tagged packets of a VLAN on an interface if the VLAN is explicitly configured on the Citrix ADC appliance and the interface is bound to the VLAN. Some deployments (for example, Bump in the wire) require the Citrix ADC appliance to function as a transparent device to accept and forward tagged packets related to a large number of VLANs. For this requirement, configuring and managing a large number of VLANs is not a feasible solution.

Allowed VLAN list on an interface specifies a list of VLANs. The interface transparently accepts and

sends tagged packets related to the specified VLANs without the need for explicitly configuring these VLANs on the appliance.

Points to Consider before Configuring Allowed VLAN List

Consider the following points before configuring allowed VLAN list

- In a high availability setup, allowed VLAN list is not propagated or synchronized. Therefore, you have to configure allowed VLAN list on both the nodes.
- The traffic of a native VLAN might leak to the non-member interfaces that specifies the native VLAN in its allowed VLAN list.
- A Maximum of 60 VLAN ranges can be specified as part of allowed VLAN list for an interface.
- The Citrix ADC appliance does not support allowed VLAN list on interfaces that are part of link aggregation channels or redundant interface sets. For more information on redundant interface set, see Redundant Interface Set.
- Allowed VLAN list is not supported on a Citrix ADC cluster configuration.
- The Citrix ADC appliance does not support allowed VLAN list for Bridge groups.
- The Citrix ADC appliance does not support allowed VLAN list for VXLANs.

Configuring Allowed VLAN List

To configure allowed VLAN list by using the CLI:

At the command prompt, type:

- **set interface** <id> **-trunkmode** (ON|OFF) **-trunkAllowedVlan** <int[-int]> …
- **show interface** <id>

To configure allowed VLAN list by using the GUI:

Navigate to **System** > **Network** > **Interfaces**, select a network interface, click **Edit**, and then set the following parameters:

- Trunk Mode
- Trunk Allowed VLAN

Sample Configuration:

In the following sample configuration, VLANS in the ranges 100-120, 190-200, and 300-330 are specified as part of allowed VLAN list for interface 1/2.

```
1 > set int 1/2 -trunkmode on -trunkallowedVlan 100-120 190-200 300-330
2
3 Done
4
5 > sh int 1/2
```



```
6
7  1)      Interface 1/2 (Gig Ethernet 10/100/1000 MBits) #6
8          flags=0xc020
9
10         <ENABLED, UP, UP, AUTONEG OFF, HEARTBEAT, 802.1q, trunkmode>
11
12         Trunk Allowed Vlans:  100-120 190-200 300-330
13
14 Done
15
16 <!--NeedCopy-->
```

Configuring Bridge Groups

September 14, 2021

Typically, when you want to merge two or more VLANs into a single domain, you change the VLAN configuration on all the devices in the separate domains. This can be a tedious task. To more easily merge multiple VLANs into a single broadcast domain, you can use bridge groups.

The bridge groups feature works the same way as a VLAN. Multiple VLANS can be bound to a single bridge group, and all VLANs bound to same bridge group form a single broadcast domain. You can bind only Layer 2 VLANs to a bridge group. For Layer 3 functionality, you must assign an IP address to a bridge group.

In Layer 2 mode, a broadcast packet received on an interface belonging to a particular VLAN is bridged to other VLANs that belong to the same bridge group. In the case of a unicast packet, the Citrix ADC appliance searches its bridge table for the learned MAC addresses of all the VLANs belonging to same bridge group.

In Layer 3 forwarding mode, an IP subnet is bound to a bridge group. The Citrix ADC accepts incoming packets belonging to the bound subnet and forwards the packets only on VLANs that are bound to the bridge group.

IPv6 routing can be enabled on a configured bridge group.

> **Note**
>
> Bridge Group feature and Bridge BPDU mode cannot work together.

Configuration steps

Perform the following steps to configure a bridge group:

- Enable Layer 2 mode
- Add a bridgegroup and bind VLANs to the bridgegroup

CLI procedures

To enable Layer 2 mode by using the CLI:
At the command prompt, type:

- **enable ns mode l2**
- **show ns mode**

To add a bridge group and bind VLANs by using the CLI:

At the command prompt, type:

- **add bridgegroup** <id> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **bind bridgegroup** <id> -**vlan** <positive_integer>
- **show bridgegroup** <id>

Example:

```
1 > add bridgegroup 12
2  Done
3 <!--NeedCopy-->
```

To remove a bridge group by using the CLI:

At the command prompt, type:

- **rm bridgegroup** <id>

Example:

```
1  rm bridgegroup 12
2 <!--NeedCopy-->
```

GUI Procedures

To configure a bridge group by using the GUI:

Navigate to **System** > **Network** > **Bridge Groups**, add a new bridge group and bind VLANs to the bridgegroup, or edit an existing bridge group.

Configuring virtual MACs

September 14, 2021

The primary and secondary nodes in a high availability (HA) setup share the virtual MAC address floating entity. The primary node owns the floating IP addresses (such as MIP, SNIP, and VIP) and responds to ARP requests for these IP addresses with its own MAC address. Therefore, the ARP table of an external device, such as an upstream router, is updated with the floating IP address and the MAC address of the primary node.

When a failover occurs, the secondary node takes over as the new primary node. The former secondary node uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it had learned from the old primary node. The MAC address that the new primary node advertises is the MAC address of its own network interface. Some devices (a few routers) do not accept these GARP messages. Therefore, these external devices retain the IP address-to-MAC address mapping that the old primary node had advertised. This can result in a GSLB site going down.

Therefore, you must configure a virtual MAC on both nodes of an HA pair. This means that both nodes have identical MAC addresses. When a failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

For the procedures to configure a virtual MAC, see Configuring Virtual MAC Addresses.

Configuring Link Aggregation

September 14, 2021

Link aggregation combines data coming from multiple ports into a single high-speed link. Configuring link aggregation increases the capacity and availability of the communication channel between the Citrix ADC appliance and other connected devices. An aggregated link is also referred to as a "channel." You can configure the channels manually, or you can use Link Aggregation Control Protocol (LACP). You cannot apply LACP to a manually configured channel, nor can you manually configure a channel created by LACP.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. (That is, the network interface parameters are ignored.) A network interface can be bound only to one channel.

When a network interface is bound to a channel, it drops its VLAN configuration. When network interfaces are bound to a channel, either manually or by LACP, they are removed from the VLANs that they originally belonged to and added to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you bind the network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then bind them to a channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them back to VLAN 2.

Configuring Link Aggregation Manually

When you create a link aggregation channel, its state is DOWN until you bind an active interface to it. You can modify a channel at any time. You can remove channels, or you can enable/disable them.

CLI procedures

To create a link aggregation channel by using the CLI:

At the command prompt, type:

- add channel <id> [-ifnum <interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)][-tagall (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show channel

Example:

```
> add channel LA/1 -ifnum 1/8
Done
<!--NeedCopy-->
```

To bind an interface to or unbind an interface from an existing link aggregation channel by using the CLI:

At the command prompt, type one of the following commands:

- bind channel <id> <interfaceName>
- unbind channel <id> <interfaceName>

Example:

```
bind channel LA/1 1/8
<!--NeedCopy-->
```

To modify a link aggregation channel by using the CLI:

At the command prompt, type the
set channel command, the channel ID, and the parameters to be changed, with their new values.

To remove a link aggregation channel by using the CLI:

Important: When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.

At the command prompt, type:

- rm channel <id>

Example:

```
> rm channel LA/1
 Done
<!--NeedCopy-->
```

GUI procedures

To configure a link aggregation channel by using the GUI:

Navigate to System > Network > Channels, add a new channel, or edit an existing channel.

To remove a link aggregation channel by using the GUI:

> **Important:**
>
> When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.

Navigate to System > Network > Channels, select the channel that you want to remove and click Delete.

Configuring Link Aggregation by Using the Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) enables network devices to exchange link aggregation information by exchanging LACP Data Units (LACPDUs). Therefore, you cannot enable LACP on network interfaces that are members of a channel that you created manually.

When using LACP to configure link aggregation, you use different commands and parameters for modifying link aggregation channels than you do for creating link aggregation channels. To remove a channel, you must disable LACP on all interfaces that are part of the channel.

Note: In an High Availability configuration, LACP configurations are neither propagated nor synchronized.

Configuring the LACP System Priority

The LACP system priority determines which peer device of an LACP LA channel can have control over the LA channel. This number is globally applied to all LACP channels on the appliance. The lower the value, the higher the priority.

To configure the LACP system priority by using the CLI:

At the command prompt, type the following commands to set the priority for a standalone appliance and verify the configuration:

- set lacp -sysPriority <positive_integer>
- show lacp

Example:

```
set lacp -sysPriority 50
<!--NeedCopy-->
```

To set the priority for a specific cluster node, log on to the cluster IP address and at the command prompt, type the following commands:

- set lacp -sysPriority <positive_integer> -ownerNode <positive_integer>
- show lacp

Example:

```
set lacp -sysPriority 50 -ownerNode 2
<!--NeedCopy-->
```

To configure the LACP system priority by using the GUI:

1. Navigate to System > Network > Interfaces and, in the Action list, select Set LACP.
2. Specify the system priority and the owner node (applicable only for a cluster setup).

Creating Link Aggregation Channels

For creating a link aggregation channel by using LACP, you need to enable LACP and specify the same LACP key on each interface that you want to be the part of the channel. For example, if you enable LACP and set the LACP Key to 3 on interfaces 1/1 and 1/2, a link aggregation channel LA/3 is created and interfaces 1/1 and 1/2 are automatically bound to it.

Note:

- When enabling LACP on a network interface, you must specify the LACP Key.
- By default, LACP is disabled on all network interfaces.

To create an LACP channel by using the CLI:

At the command prompt, type:

- set interface <id> [-lacpMode <lacpMode>] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>] [-lacpTimeout (LONG | SHORT)]
- show interface [<id>]

To create an LACP channel by using the GUI:

Navigate to System > Network > Interfaces, open the network interface, and set the parameters.

Modifying Link aggregation Channels

After you have created an LACP channel by specifying interfaces, you can modify properties of the channel.

To modify an LACP channel by using the CLI:

At the command prompt, type:

- set channel <id> [-ifnum <interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-tagall (ON | OFF)] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show channel

Example:

```
> set channel LA/3 -state ENABLED -speed 10000
 Done
<!--NeedCopy-->
```

To modify an LACP channel by using the GUI:

Navigate to System > Network > Channels, and modify an existing LACP channel.

Removing a Link Aggregation Channel

To remove a link aggregation channel that was created by using LACP, you need to disable LACP on all the interfaces that are part of the channel.

To remove an LACP channel by using the CLI:

At the command prompt, type:

- set interface <id> -lacpMode Disable
- show interface [<id>]

To remove an LACP channel by using the GUI:

Navigate to System > Network > Interfaces, open the network interface, and clear the Enable LACP option.

Link Redundancy using LACP channels

Link Redundancy using LACP channels enables the Citrix ADC to divide an LACP channel into logical subchannels, with one subchannel active and the others in standby mode. If the active subchannel fails to meet a minimum threshold of throughput, one of the standby subchannels becomes active and takes over.

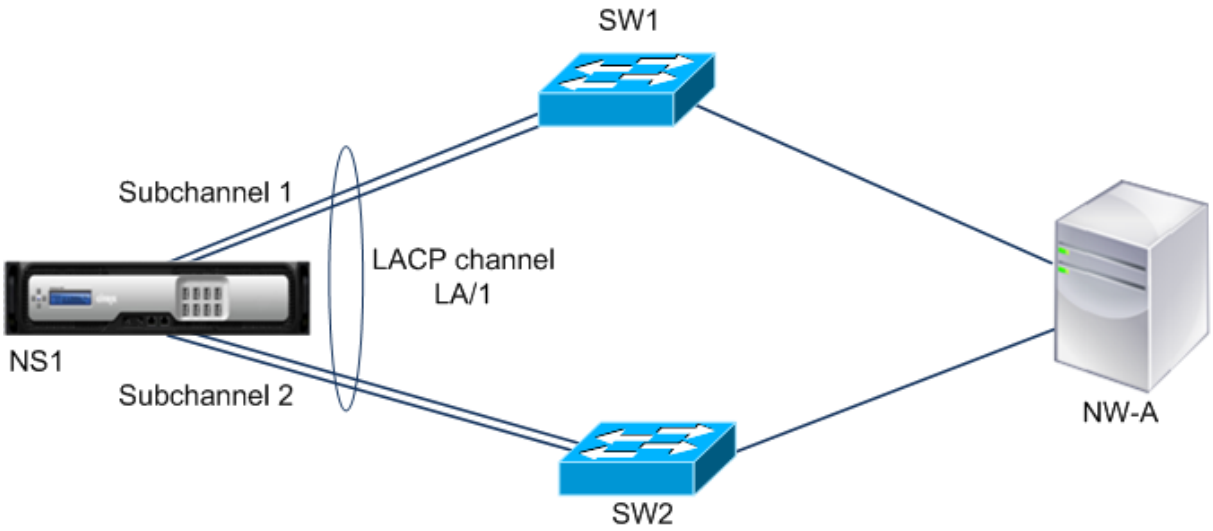
A subchannel is created from links that are part of the LACP channel and are connected to a particular device. For example, for an LACP channel with four interfaces on a Citrix ADC, with two of the interfaces connected to device A and the other two connected to device B, the ADC creates two logical subchannels, one subchannel with two links to device A, and another subchannel with two links to device B.

To configure link redundancy for an LACP channel, set the lrMinThroughput parameter, which specifies the minimum throughput threshold (in Mbps) to be met by the active subchannel. Setting this parameter automatically creates the subchannels. When the maximum supported throughput of the active channel falls below the lrMinThroughput value, link failover occurs and a standby subchannel becomes active.

If you unset the lrMinThroughput parameter of an LACP channel, or set the value to zero, link redundancy for that channel is disabled, which is the default setting.

Example

Consider an example of link redundancy configured between Citrix ADC NS1 and switches SW1 and SW2.



NS1 is connected to network device NW-A through SW1 and SW2.
On NS1, LACP channel LA/1 is created from interfaces 1/1, 1/2, 1/3, and 1/4. Interfaces 1/1 and 1/2 of NS1 are connected to SW1, and interfaces 1/3 and 1/4 are connected to SW2. Each of the four links supports a maximum throughput of 1000Mbps.
When the lrMinThroughput parameter is set to some value (say 2000), NS1 creates two logical subchannels from LA/1, one subchannel (say subchannel 1) using interfaces 1/1 and 1/2 (connected to SW1), and the other subchannel (subchannel 2) using interfaces 1/3 and 1/4 (connected to SW2).

NS1 applies an algorithm to make one subchannel (say subchannel 1) active and put the other on

standby. NS1 and network device NW-A are accessible to each other through only the active subchannel.

Say subchannel 1 is active, and its maximum supported throughput falls below the lrMinThroughput value (for example, one of its links fails, and the maximum supported throughput falls to 1000 Mbps). Subchannel 2 becomes active and takes over.

Link Redundancy using LACP channels in a High Availability setup

In a high availability (HA) configuration, if you want to configure throughput (throughput parameter) based HA failover and link redundancy (lrMinThroughput parameter) on an LACP channel, you must set the throughput parameter to a value less than or equal to that of the lrMinThroughput parameter.

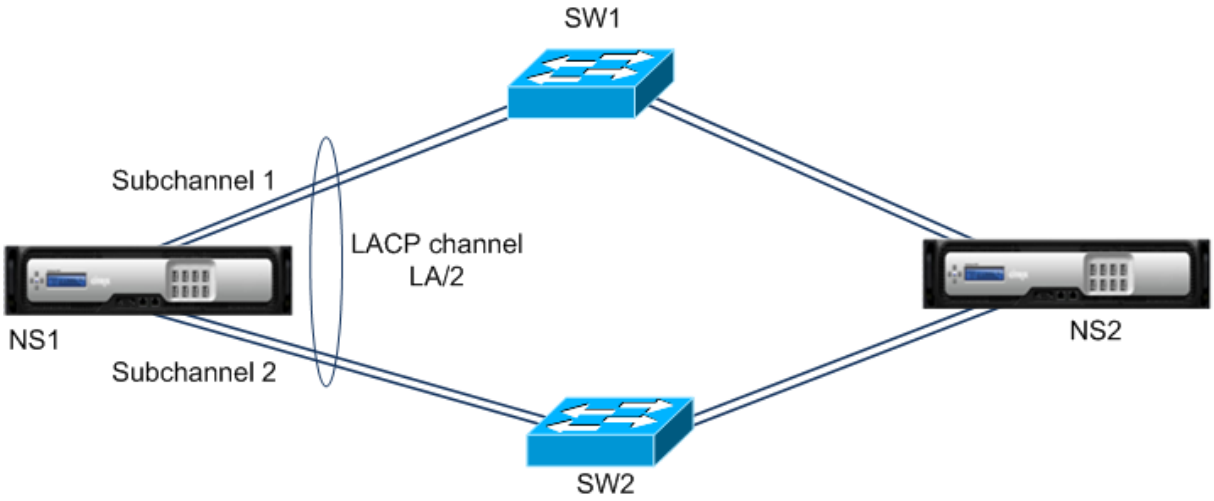
The maximum supported throughput of an LACP channel is calculated as the maximum supported throughput of the active subchannel.

If the throughput parameter value is equal to or less than the lrminthroughput parameter value, HA failover occurs when both of the following conditions exist at the same time:

- None of the subchannels' maximum supported throughput meet the lrMinThroughput parameter value.
- The maximum supported throughput of the LACP channel does not meet the throughput parameter value

Consider an example of an HA setup that has Citrix ADCs NS1 and NS2, with switches SW1 and SW2. NS1 is connected to NS2 through SW1 and SW2.

On NS1, LACP channel LA/1 is created from interfaces 1/1, 1/2, 1/3, and 1/4. Interfaces 1/1 and 1/2 of NS1 are connected to SW1, and interfaces 1/3 and 1/4 are connected to SW2. Each of the four links supports a maximum throughput of 1000 Mbps.



Following are the LACP-parameter settings in this example:

Parameter	Value
Throughput	2000
lrminthroughput	2000

NS1 forms two subchannels from LA/1, one subchannel (say subchannel 1) using interfaces 1/1 and 1/2 (connected to SW1), and the other subchannel (subchannel 2) using interfaces 1/3 and 1/4 (connected to SW2). Each of the two subchannels supports a maximum throughput of 2000 Mbps. Applying an algorithm, NS1 makes one subchannel (say subchannel 1) active and the other standby.

Say subchannel 1 is active, and its maximum supported throughput falls below the lrMinThroughput value (for example, one of its links fails, and maximum supported throughput falls to 1000 Mbps). Subchannel 2 becomes active and takes over. HA failover does not occur, because the maximum supported throughput of the LACP channel is not less than the throughput parameter value:

Maximum supported throughput of the LACP channel = Maximum supported throughput of the active channel = Maximum supported throughput of subchannel 2 = 2000 Mbps

If subchannel 2's maximum supported throughput also falls below the lrminthroughput value (for example, one of its links fails, and the maximum supported throughput falls to 1000 Mbps), HA failover occurs, because the maximum supported throughput of the LACP channel is then less than the throughput parameter value:

Configure Link Redundancy using LACP channels

To configure link redundancy for a LACP channel by using the CLI:

At the command prompt, type the following commands to configure the channel and verify the configuration:

- **set channel** <id> -lrMinThroughput <positive_integer>
- **show channel**

Example:

```
1 > set channel la/1 - lrMinThroughput 2000
2 Done
3 > set channel la/2 - throughput 2000 - lrMinThroughput 2000
4 Done
5 <!--NeedCopy-->
```

To configure link redundancy for a LACP channel by using GUI

1. Navigate to System > Network > Channels.

2. In the details pane, select an LACP channel for which you want to configure link redundancy, and then click Edit.
3. In the Configure LACP channel dialog box, set the lrMinThroughput parameter.
4. Click Close.

Redundant Interface Set

September 14, 2021

A redundant interface set is a set of interfaces where one of the interfaces is active and the remaining ones are standby. If the active interface fails, one of the standby interfaces takes over and becomes active.

The following are the main benefits of using redundant interface sets:

- A redundant interface set ensures connection reliability between the Citrix ADC appliance and a peer device by providing back up links between them.
- Unlike link redundancy using LACP, no configuration is required on the peer device for a redundant interface set. To the peer device, redundant interface set appear as individual interfaces and not as a set or collection.
- In an high availability configuration (HA), redundant interface sets can minimize the number the HA failovers.

> **Note**
>
> Redundant Interface Set was formerly known as 'NIC bundling' when first introduced in 10.5 release.

How Redundant Interface Set Works

For a redundant interface set, the Citrix ADC appliance derives a MAC address on the basis of an internal algorithm and assigns it to the redundant interface set. This MAC address is shared by all the member interfaces and is used only by the active interface at a time. The active interface broadcasts GARP messages, which contains the MAC address assigned to the redundant interface set and not the interface's own physical MAC address. When the current active interface fails and is taken over by another interface, the new active interface sends GARP messages. The peer device updates its forwarding table with the new active interface information. The standby interfaces do not send any GARP messages. The standby interfaces do not send any packets and they drop any packets they receive.

In a redundant interface set, selection of the member interface as active is based on either of the following factors:

- **Redundant interface priority**. This is a parameter of an interface and it defines the priority of the interface in a redundant interface set for the active member selection. This parameter specifies a positive integer. Lower the value higher the priority of active member selection. The member interface with the highest priority (lowest value) is selected as the active interface of the redundant interface set.
- **Binding order of the member interfaces**. If all the member interfaces have the same redundant interface priority, the member interface that was bound first to the redundant interface set is selected as the active interface of the redundant interface set.

In a redundant interface set, active interface selection is triggered in one of the following events:

- When the current active interface fails or you disable it.
- When you set the priority of a standby interface to a value lower than that of the current active interface. The standby interface takes over as the active interface.
- When you bind an interface whose priority is lower than that of the current active interface. The newly bound interface takes over as the active interface.

Points to Consider for Configuring Redundant Interface Sets

Consider the following points before you configure a redundant interface set:

- In a standalone appliance or an appliance in a high availability setup, a link redundant set is specified in LR/X notation, where X can range from 1 to 4. For example, LR/1.
- In a high availability configuration, redundant interface set configurations do not propagate or synchronize to the secondary node.
- You can configure a maximum of four redundant interface sets on a Citrix ADC appliance.
- You can bind a maximum of 16 interfaces to a redundant interface set.
- Member interfaces of a redundant interface set cannot be bound to another redundant interface set.
- Member interfaces of a redundant interface set cannot be bound to a link aggregate (LA) channel.
- LA channels cannot be bound to a redundant interface set.
- Redundant interface sets cannot be bound to an LA channel.
- In a cluster setup:
  - Redundant interface sets cannot be bound to a cluster link aggregation.
  - A link redundant set is specified in N/LR/X notation (for example, 1/LR/3). Where:
    N is the ID of the cluster node on which the redundant interface set is to be created.
    X is a link-redundant set identifier on a cluster node. X can range from 1-4.
  - A cluster link aggregation cannot be bound to a redundant interface set.
  - A redundant interface set can include only the interfaces of the node to which the redundant interface set belongs.

- An Existing elink redundancy set configuration on a standalone appliance automatically changes to cluster notation (N/LR/X) after the appliance is added to a cluster setup.

Configuration Steps

Configuring redundant interface set on a Citrix ADC appliance consists of the following tasks:

- **Create a redundant interface set**. Use the channel command operation for creating a redundant interface set.

  In a standalone appliance or an appliance in a high availability setup, a link redundant set is specified in LR/X notation, where X can range from 1 to 4. For example, LR/1.

  In a cluster setup, a link redundant set is specified in N/LR/X (for example, 1/LR/3), where: N is the ID of the cluster node on which the redundant interface set is to be created, X is link redundant set identifier on a cluster node. X can range from 1-4.

- **Bind interfaces to the redundant interface set**. Associate the desired interfaces with the redundant interface set. An interface cannot be a part of multiple redundant interface sets.

- **(Optional) Set a redundant interface priority on the member interface**. Use the interface command operation for setting the redundant interface priority on a desired member interface of a redundant interface set.

To create a redundant interface set by using the CLI:

At the command prompt:

- add channel <ID>
- show channel <ID>

To bind interfaces to a redundant interface set by using the CLI:

At the command prompt:

- bind channel <ID> <ifnum>
- show channel <ID>

To set a redundant interface priority of an interface by using the CLI:

At the command prompt:

- set interface <ID> -lrsetpriority <positive_integer>
- show interface <ID>

Sample configuration 1:

In the following example, redundant interface set LR/1 is created, and interfaces 1/1, 1/2, 1/3, and 1/4 are bound to LR/1. The redundant interface priority is set to a default value of 1024 for all these member interfaces. Output of the show channel command displays that the interface 1/1 is the current active interface for the redundant interface set lr/1.


```
> add channel lr/1
 Done
> bind channel lr/1 1/1 1/2 1/3 1/4
 Done
> show channel
1)    Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
Done
<!--NeedCopy-->
```

Sample configuration 2:

In the following example, redundant interface priority of the member interface 1/4 is set to 100, which is lower than the set redundant interface priority of all the other member interfaces of LR/1.

Output of the show channel command displays that the interface 1/4 is the current active interface for the redundant interface set LR/1.

```
> set interface 1/4 -lrsetPriority 100
Done
> show channel
1)      Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
```



```
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
Done
<!--NeedCopy-->
```

Sample configuration 3:

Consider a cluster setup of four nodes N1, N2, N3, and N4. In this example, redundant interface set 1/LR/3 is created on node N1, and interfaces 1/1/1, 1/1/2, and 1/1/3 are bound to it. The redundant interface priority is set to a default value of 1024 for all these member interfaces. Output of the show channel command indicates that interface 1/1/1 is the current active interface for redundant interface set 1/LR/3.

```
    > add channel 1/LR/3

    Done
    > bind channel 1/LR/3 1/1/1 1/1/2 1/1/3

    Done
    > show channel
    1)    Interface 1/LR/3 (Link Redundant) #14
            flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON,
               802.1q>
            MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0
               h00m00s
            Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
            throughput 0
            Actual: throughput 1000
            LLDP Mode: NONE,
            RX: Pkts(66) Bytes(4406) Errs(0) Drops(82) Stalls(0)
            TX: Pkts(55) Bytes(2626) Errs(0) Drops(145) Stalls(0)
            NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted
               (0)
            Bandwidth thresholds are not set.

```



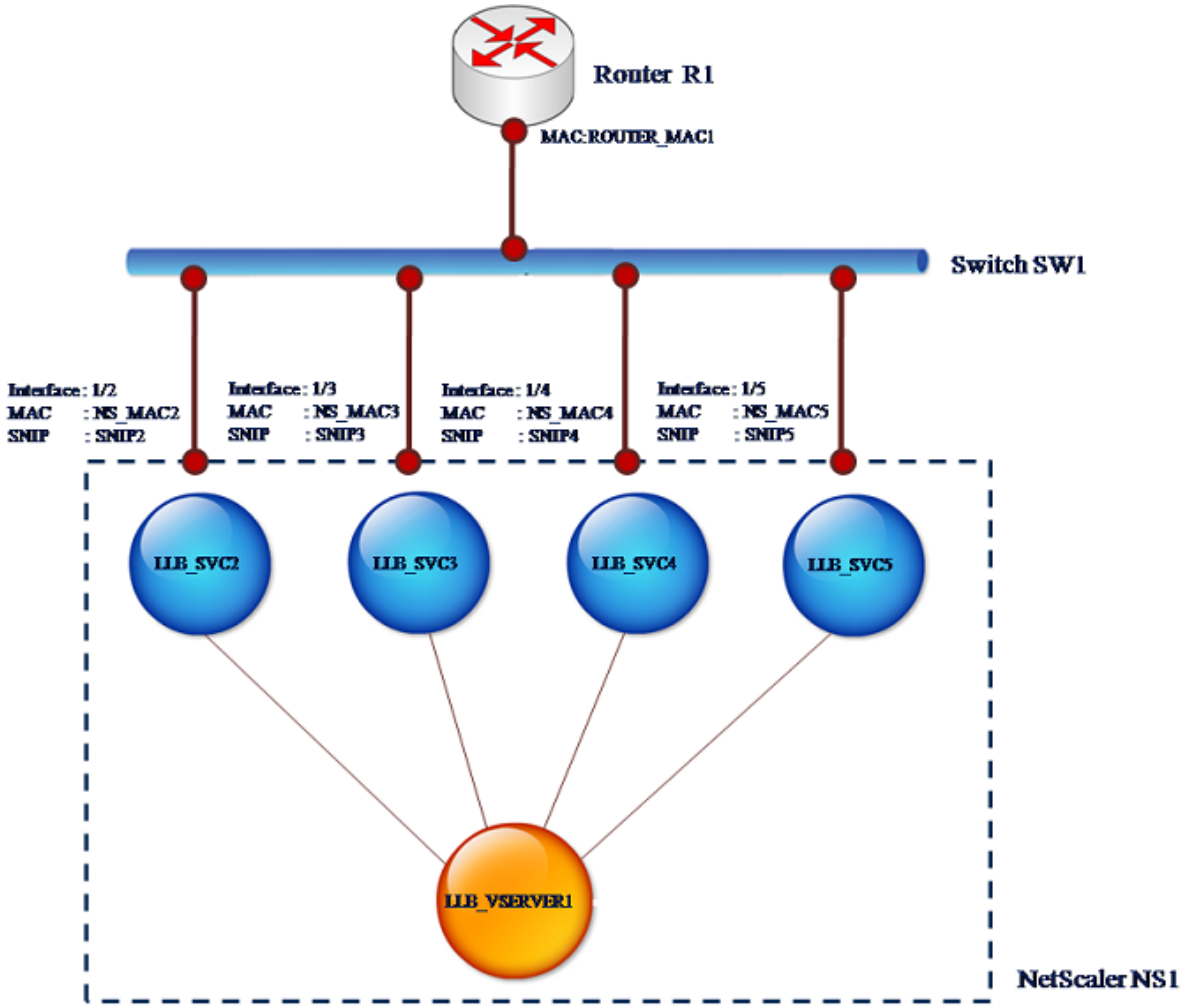
```
20              1/1/1: UTP-1000-FULL-OFF UP  0h14m06s LR Active Member
21              1/1/2: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
22              1/1/3: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
23
24     Done
25 <!--NeedCopy-->
```

Binding an SNIP address to an Interface

September 14, 2021

You can now bind a Citrix ADC owned SNIP address to an interface without using Layer 3 VLANs. Any packets related to the SNIP address will go only through the bound interface.

This feature can be useful in a scenario where the upstream switch does not support Link Aggregation channels and you want the Citrix ADC appliance to load balance traffic, originated from a server, across the four links to the upstream switch as shown in the following illustration.



The following tables describe the example settings for the scenario:

Entity	Name	Value
SNIP addresses on NS1	SNIP2 (for reference purpose only)	10.10.10.2
	SNIP3 (for reference purpose only)	10.10.10.3
	SNIP4 (for reference purpose only)	10.10.10.4
	SNIP5 (for reference purpose only)	10.10.10.5
LLB virtual server on NS1	LLB_VSERVER1	-
Transparent monitor on NS1	TRANS_MON	-

Entity	Name	Value
LLB services on NS1	LLB_SVC2	10.10.10.240
	LLB_SVC3	10.10.10.120
	LLB_SVC4	10.10.10.60
	LLB_SVC5	10.10.10.30
MAC address of interface 1/2 on NS1	NS_MAC_2 (for reference purpose only)	00:e0:ed:0f:bc:e0
MAC address of interface 1/3 on NS1	NS_MAC_3 (for reference purpose only)	00:e0:ed:0f:bc:df
MAC address of interface 1/4 on NS1	NS_MAC_4 (for reference purpose only)	00:e0:ed:0f:bc:de
MAC address of interface 1/5 on NS1	NS_MAC_5 (for reference purpose only)	00:e0:ed:1c:89:53
IP address of Router R1	Router_IP (for reference purpose only)	10.10.10.1
MAC address of interface of R1	ROUTER_MAC1 (for reference purpose only)	00:21:a1:2d:db:cc

To configure the example settings:

1. Add four different SNIPs in different subnet ranges. This is for ARP to be resolved on four different links. For more information on creating a SNIP address, see Configuring Subnet IP Addresses (SNIPs).

   CLI example:

```
> add ns ip 10.10.10.2 255.255.255.0 -type SNIP
 Done
> add ns ip 10.10.10.3 255.255.255.128 -type SNIP
Done
> add ns ip 10.10.10.4 255.255.255.192 -type SNIP
Done
> add ns ip 10.10.10.5 255.255.255.224 -type SNIP
Done
<!--NeedCopy-->
```

2. Add four different dummy services in the added SNIP subnets. This is to ensure that the traffic is sent out with source IP as one of the four configured SNIPs. For more information on creating a service, see Set up basic load balancing.

CLI example:

```
> add service LLB_SVC2 10.10.10.240 any *
 Done
> add service LLB_SVC3 10.10.10.120 any *
 Done
> add service LLB_SVC4 10.10.10.60 any *
 Done
> add service LLB_SVC5 10.10.10.30 any *
 Done
<!--NeedCopy-->
```

3. Add a transparent ping monitor for monitoring the gateway. Bind the monitor to each of the configured dummy services. This is to make the state of the services as UP. For more information on creating a transparent monitor, see Configure monitors in a load balancing setup.

   CLI example:

```
> add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
 Done
> bind monitor TRANS_MON LLB_SVC2
 Done
> bind monitor TRANS_MON LLB_SVC3
 Done
> bind monitor TRANS_MON LLB_SVC4
 Done
> bind monitor TRANS_MON LLB_SVC5
 Done
<!--NeedCopy-->
```

4. Add a link load balancing (LLB) virtual server and bind the dummy services to it. For more information on creating an LLB virtual server, see Configuring a Basic LLB Setup.

   CLI example:

```
> add lb vserver LLB_VSERVER1 any
 Done
> set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
```



```
 Done
<!--NeedCopy-->
```

5. Add the LLB virtual server as the default LLB route. For more information on creating an LLB route see Configuring a Basic LLB Setup.

   CLI example:

```
> add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
 Done
<!--NeedCopy-->
```

6. Add an ARP entry for each of the dummy services with the MAC address of the gateway. This way the gateway is reachable through these dummy services. For more information on adding an ARP entry, see Configuring Static ARP.

   CLI example:

```
> add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum
   1/2
 Done
> add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum
   1/3
 Done
> add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
 Done
> add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
 Done
<!--NeedCopy-->
```

7. Bind a specific interface to an SNIP by adding an ARP entry for each of these SNIPs. This is to ensure that the response traffic will reach the same interface through which the request went out. For more information on adding an ARP entry, see Configuring Static ARP.

   CLI example:

```
> add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
 Done
> add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
 Done
> add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
 Done
> add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
 Done
<!--NeedCopy-->
```


Monitor the Bridge Table and Changing the Aging time

September 14, 2021

Citrix ADC appliance bridges frames on the basis of bridge table lookup of the destination MAC address and the VLAN ID. However, the appliance performs forwarding only when Layer 2 mode is enabled.

The bridge table is dynamically generated, but you can display it, modify the aging time for the bridge table, and view bridging statistics. All the MAC entries in the bridge table are updated with the aging time.

To set the aging time of bridge table entries by using the CLI:

At the command prompt, type:

- **set l2param** -**bridgeagetimeout** <positive_integer>
- **show l2param**

Example:

```
> set l2param -bridgeagetimeout 90
 Done
<!--NeedCopy-->
```

To view the statistics of a bridge table by using the CLI:

At the command prompt, type:

- **stat bridge**

To set the aging time of bridge table entries by using the GUI:

Navigate to **System** > **Network**. On the **Network** page, in the **Settings** section, click **Configure Layer2 Parameters** and set the **Timeout Value For The Bridge Table Entries (seconds)** parameter.

To view the statistics of a bridge table by using the GUI:

Navigate to **System** > **Network** > **Bridge Table**, select the MAC address, and click **Statistics**.

Citrix ADC Appliances in Active-Active Mode Using VRRP

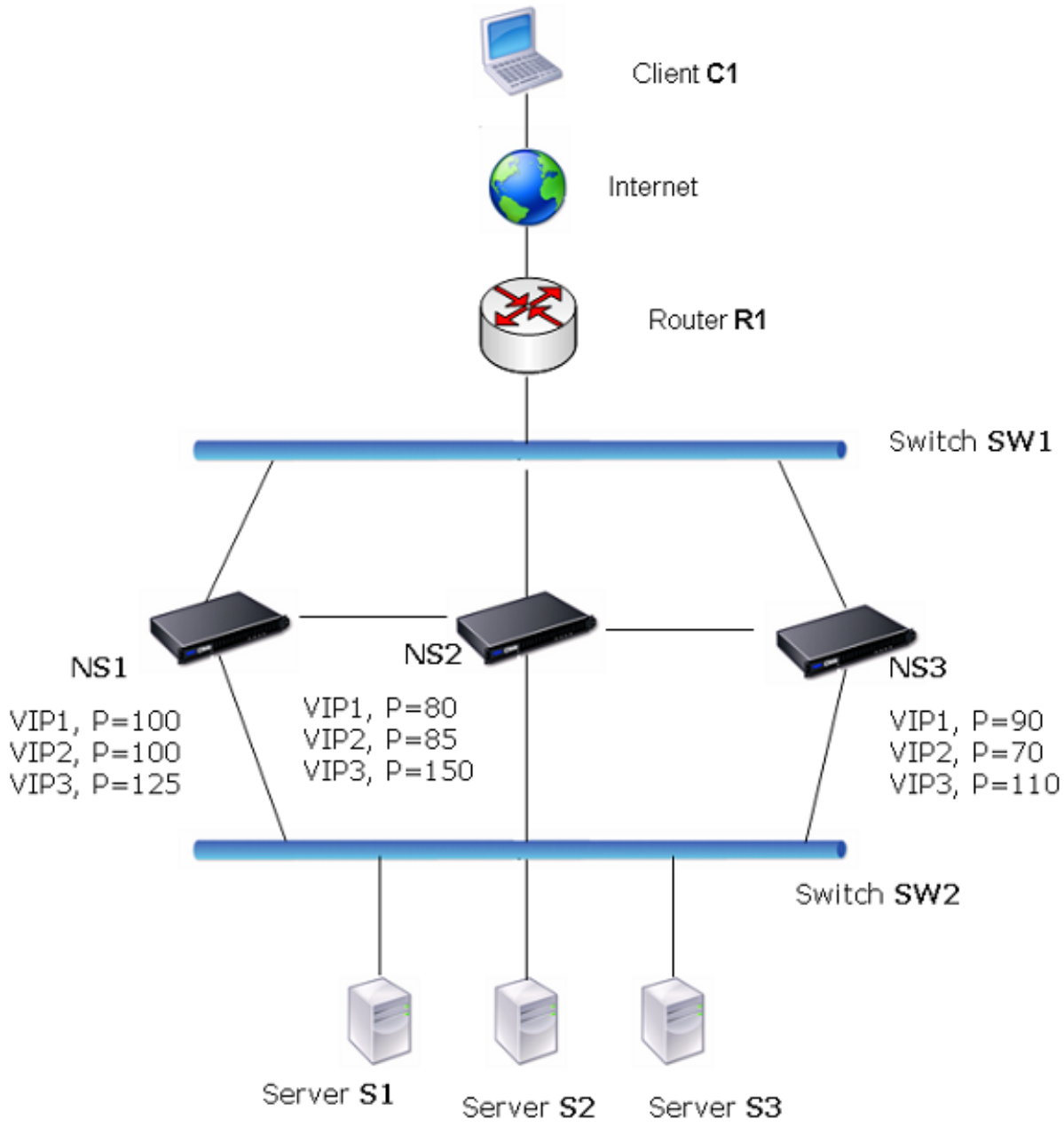
September 14, 2021

An active-active deployment, in addition to preventing downtime, makes efficient use of all the Citrix ADC appliances in the deployment. In active-active deployment mode, the same VIPs are configured on all Citrix ADC appliances in the configuration, but with different priorities, so that a given VIP can be active on only one appliance at a time.

The active VIP is called the master VIP, and the corresponding VIPs on the other Citrix ADC appliances are called the backup VIPs. If a master VIP fails, the backup VIP with the highest priority takes over and becomes the master VIP. All the Citrix ADC appliances in an active-active deployment use the Virtual Router Redundancy Protocol (VRRP) protocol to advertise their VIPs and the corresponding priorities at regular intervals.

Citrix ADC appliances in active-active mode can be configured so that no Citrix ADC is idle. In this configuration, different sets of VIPs are active on each Citrix ADC. For example, in the following diagram, VIP1, VIP2, VIP3, and VIP4 are configured on appliances NS1, NS2, and NS3. Because of their priorities, VIP1 and VIP 2 are active on NS1, VIP3 is active on NS2 and VIP 4 is active on NS3. If, for example, NS1 fails, VIP1 on NS3 and VIP2 on NS2 become active.

Figure 1. An Active-Active Configuration



The Citrix ADC appliances in the above diagram process traffic as follows:

1. Client C1 sends a request to VIP1. The request reaches R1.
2. R1 does not have an ARP entry for VIP1, so it broadcasts an ARP request for VIP1.
3. VIP1 is active in NS1, so NS1 replies with a source MAC address as the virtual MAC (for example virtual MAC1) associated with VIP1, and VIP1 as the source IP address.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table.
5. R1 updates the ARP entry with virtual MAC1 and VIP1.

6. R1 forwards the packet to the VIP1 on NS1.
7. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP addresses and S2.
8. S2 replies to the SNIP on the Citrix ADC.
9. NS1 sends S2's reply to the client. In the reply, NS1 inserts MAC address of the physical interface as the source MAC address and VIP1 as the source IP address.
10. Should NS1 fail, the Citrix ADC appliances use the VRRP protocol to select the VIP1 with the highest priority. In this case, VIP1 on NS3 becomes active, and the following two steps update the active-active configuration.
11. NS3 broadcasts a GARP message for VIP1. In the message, virtual MAC1 is the source MAC address and VIP1 is the source IP address.
12. SW1 learns the new port for virtual MAC1 from the GARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS3. R1 updates its ARP table.

The priority of a VIP can be modified by health tracking. If you enable health tracking, you should make sure that preemption is also enabled, so that a VIP whose priority is lowered can be preempted by another VIP.

In some situations, traffic might reach a backup VIP. To avoid dropping such traffic, you can enable sharing, on a per-node basis, as you create an active-active configuration. Or you can enable the global send to master option. On a node on which sharing is enabled, it takes precedence over send to master.

Health Tracking

Base priority (BP-range 1-255) ordinarily determines which VIP is the master VIP, but effective priority (EP) can also affect the determination.

For example, if a VIP on NS1 has a priority of 101 and same VIP on NS2 has a priority of 99, the VIP on NS1 is active. However, if two vservers are using the VIP on NS1 and one of them goes DOWN, health tracking can reduce the EP of VIP on NS1. VRRP then makes the VIP on NS2 the active VIP.

Following are the health tracking options for modifying EP:

- **NONE.** No tracking. EP = BP
- **ALL.** If all virtual servers are UP, then EP = BP. Otherwise, EP = 0.
- **ONE.** If at least one virtual server is UP, then EP = BP. Otherwise, EP = 0.
- **PROGRESSIVE.** If ALL virtual servers are UP, then EP = BP. If ALL virtual servers are DOWN then EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP and k is the number of virtual servers that are down.

Note: If you specify a value other than NONE, preemption should be enabled, so that the backup VIP with the highest priority becomes active if the priority of the master VIP is downgraded.

Preemption

Preemption of an active VIP by another VIP that attains a higher priority is enabled by default, and normally should be enabled. In some cases, however, you may want to disable it. Preemption is a per-node setting for each VIP.

Preemption can occur in the following situations:

- An active VIP goes down and a VIP with a lower priority takes its place. If the VIP with the higher priority comes back online, it preempts the currently active VIP.
- Health tracking causes the priority of a backup VIP to become higher than that of the active VIP. The backup VIP then preempts the active VIP.

Sharing

In the event that traffic reaches a backup VIP, the traffic is dropped unless the sharing option is enabled on the backup VIP. This behavior is a per node setting for each VIP and is disabled by default.

In the figure **An Active-Active Configuration** VIP1 on NS1 is active and VIP1 VIPs on NS2 and NS3 are backups. Under certain circumstances, traffic may reach VIP1 on NS2. If Sharing is enabled on NS2, this traffic is processed instead of dropped.

Configuring Active-Active Mode

September 14, 2021

On each Citrix ADC appliance that you want to deploy in active-active mode, you must add a virtual MAC and bind the virtual MAC to a VIP. The virtual MAC for a given VIP must be same on each appliance. For example, if VIP 10.102.29.5, is created on the appliances, a virtual router ID (VRID) must be created on each Citrix ADC and bound to VIP 10.102.29.5 on each Citrix ADC. When you bind a virtual MAC to a VIP, the appliance sends VRRP advertisements to each VLAN that is bound to that VIP. The virtual MAC can be shared by different VIPs configured on the same Citrix ADC.

Configuring IPv4 Active-Active Mode

Perform the following tasks on each of the Citrix ADC appliances to be included in the active-active configuration:

- **Add a virtual MAC address**. Add a virtual MAC address by adding a VRID. You can also specify a priority and enable or disable preemption and sharing on this VRID address.

- **Add a VIP address and associate the virtual MAC's VRID**. Add a VIP address and set the VRID parameter to the newly created VRID. The attributes of the VRID (for example, priority and preemption) are bound to this VIP address.
  Note: The same VIP address must be added to all the other Citrix ADC appliances.

To add a virtual MAC address by using the CLI

At the command prompt, type:

- **add vrID** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED**|**DISABLED**)][-**sharing** (**ENABLED**|**DISABLED**)] [-**tracking** <tracking>]
- **show vrid**

To add a VIP address by using the CLI:

At the command prompt, type:

- **add ns ip** <IPv4Address> -type VIP -vrid <value>
- **show ns ip**

To configure a virtual MAC by using the GUI:

1. Navigate to **System** > **Network** > **VMAC**, on the **VMAC** tab, add a new virtual MAC, or edit an existing virtual MAC.
2. Set the following parameters:
   - Virtual Router ID
   - Priority
   - Tracking
   - Preemption
   - Sharing

To configure a VIP address and associate the VRID to it by using the GUI:

1. Navigate to **System** > **Network** > **IPs**, on the **IPV4s** tab, add an IP address of type VIP.
2. While adding the IP address, select the virtual router ID from the **Virtual Router Id** drop down box.

Sample Configuration:

The following sample configuration is for deploying Citrix ADC appliances NS1 and NS2 in IPv4 active-active mode. VIP address 203.0.113.10 is configured on both NS1 and NS2, with a different priority value on each appliance. On each appliance, this VIP address is bound to a virtual MAC address. 203.0.113.10 is master on NS2, because its priority (200) on NS2 is higher than on NS1 (100).

```
1     Settings on NS1
2
3     > add vrid 10 -Priority 100 -Preemption Enabled -sharing Enabled
4
```



```
5      Done
6
7      > add ns ip 203.0.113.10  -type VIP  -vrid 10
8
9      Done
10
11     Settings on NS2
12
13     > add vrid 10  -Priority 200  -Preemption Enabled  -sharing Enabled
14
15     Done
16
17     > add ns ip 203.0.113.10  -type VIP  -vrid 10
18
19     Done
20 <!--NeedCopy-->
```

Configuring IPv6 Active-Active Mode

Perform the following tasks on each of the Citrix ADC appliances to be included in the active-active configuration:

- **Add a virtual MAC6 address**. Add a virtual MAC6 address by adding a VRID6. You can also specify a priority and enable or disable preemption and sharing on this VRID6 address.
- **Add a VIP6 address**. Add a VIP6 address. Set the VRID6 parameter to the VRID6 of the newly created virtual MAC6.The attributes of the virtual MAC6 (for example, priority and preemption) are bound to this VIP6 address.
  Note: The same VIP6 address must be added to all the other Citrix ADC appliances.

To add a virtual MAC6 address by using the CLI:

At the command prompt, type:

- **add vrID6** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED** | **DISABLED**)] [-**sharing** (**ENABLED** | **DISABLED**)]
- **show vrID6**

To add a VIP6 address by using the CLI:

At the command prompt, type:

- **add ns ip6** <IPv6Address> -**type** VIP -**vrid** <value>
- **show ns ip6**

To configure a virtual MAC6 by using the GUI:

1. Navigate to **System** > **Network** > **VMAC**, on the **VMAC6** tab, add a new virtual MAC6, or edit an existing **VMAC6**.
2. Set the following parameters:
   - Virtual Router ID
   - Priority
   - Preemption
   - Sharing

To configure a VIP6 address and associate the VRID to it by using the GUI:

1. Navigate to **System** > **Network** > **IPs**, on the **IPV6s** tab, add an IPv6 address of type VIP.
2. While adding the VIP6 address, select the VRID6 from the **Virtual Router Id** drop down box.

Sample Configuration:

The following sample configuration is for deploying Citrix ADC appliances NS1 and NS2 in IPv6 active-active mode. VIP6 address 2001:db8::5001 is configured on both NS1 and NS2, with a different priority value on each appliance. On each appliance, this VIP6 address is bound to a virtual MAC6 address. 2001:db8::5001 is master on NS2, because it's priority (200) on NS2 is higher than on NS1 (100).

```
    Settings on NS1
    > add vrid6 10 -Priority 100 -Preemption Enable -sharing Enable

    Done
    > add ns ip6 2001:db8::5001 -type VIP -vrid6 10

    Done
    Settings on NS2
    > add vrid6 10 -Priority 200 -Preemption Enable -sharing Enable

    Done
    > add ns ip6 2001:db8::5001 -type VIP -vrid6 10

    Done
<!--NeedCopy-->
```

Configuring Send to Master

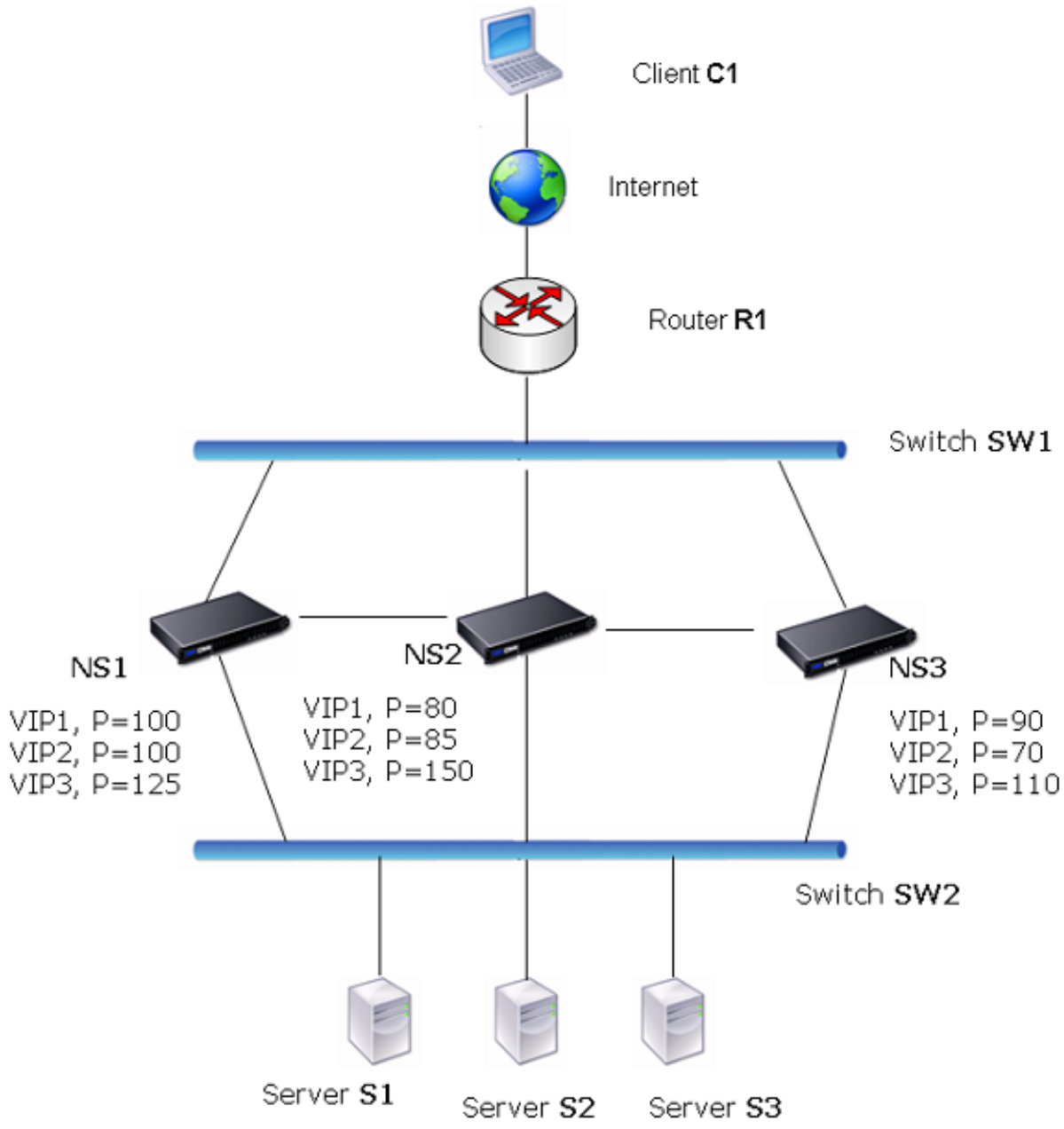
September 14, 2021

Usually, the traffic destined to a VIP reaches the Citrix ADC appliance on which the VIP is active, because an ARP request with the VIP and a virtual MAC on that appliance has reached the upstream router. But in some cases, such as static routes configured on the upstream router for the VIP subnet,

or a topology that blocks this route, the traffic can reach a Citrix ADC appliance on which the VIP is in backup state. If you want this appliance to forward the data packets to the appliance on which the VIP is active, you need to enable the send to master option. This behavior is a per node setting and is disabled by default.

For example, in the following diagram, VIP1 is configured on NS1, NS2, and NS3 and is active on NS1. Under certain circumstances, traffic for VIP1 (active on NS1) may reach VIP1 on NS3. When the send to master option is enabled on NS3, NS3 forwards the traffic to NS1 through NS2 by using route entries for NS1.

Figure 1. An Active-Active Configuration with Send to Master Option Enabled



To enable send to master by using the CLI:

At the command prompt, type:

```
set vrIDParam -sendToMaster (ENABLED          DISABLED)
```

Example:


```
> set vrIDParam -sendToMaster ENABLED
 Done
<!--NeedCopy-->
```

To enable send to master by using the GUI:

1. Navigate to **System** > **Network**, in the **Settings** group, click **Virtual Router Parameters**.
2. Select the **Send to Master** option.

Configuring VRRP Communication Intervals

September 14, 2021

In an active-active deployment, all Citrix ADC nodes use the Virtual Router Redundancy Protocol (VRRP) to advertise their master VIP addresses and the corresponding priorities in VRRP advertisement packets (hello messages) at regular intervals.

VRRP uses the following communication intervals:

- **Hello Interval**. Interval between the VRRP hello messages that a node of a master VIP address sends to its peer nodes.
- **Dead Interval**. Time after which a node of a backup VIP address considers the state of the master VIP address as DOWN if VRRP hello messages are not received from the node of the master VIP address. After the dead interval, the backup VIP address takes over and becomes the master VIP address.

You can change these intervals to a desired value. Both of these communication intervals are per node setting for all VIP addresses in that node.

To configure VRRP communication intervals by using the CLI:

At the command prompt, type:

- **set vrIDParam** [-**helloInterval** <msecs>] [-**deadInterval** <secs>]
- **sh vrIDParam**

Example:

```
> set vrIDParam -helloInterval 500 -deadInterval 2
 Done
<!--NeedCopy-->
```

To configure VRRP communication intervals by using the GUI:

1. Navigate to **System** > **Network**, in the **Settings** group, click **Virtual Router Parameters**.
2. In **Configure Virtual Router Parameter**, set the **Hello Interval** and **Dead Interval** parameters.

3. Click **OK**.

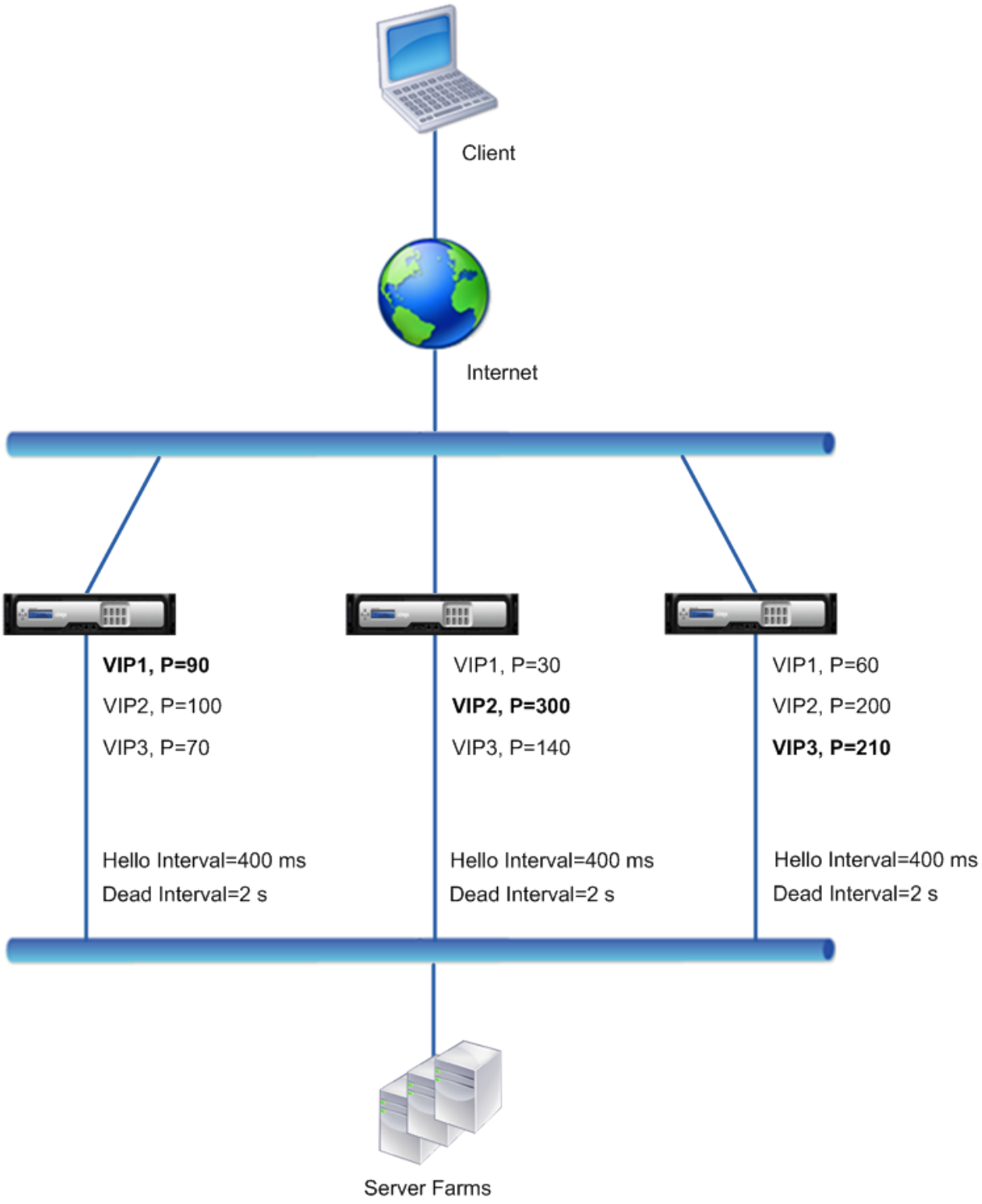
Example 1: Nodes with the Same VRRP Dead Intervals

Consider an active-active deployment consisting of Citrix ADCs NS1, NS2, and NS3. Virtual IP addresses VIP1, VIP2, VIP3 are configured on each of these ADCs. Because of their priorities, VIP1 is active on NS1, VIP2 is active on NS2, and VIP3 is active on NS3.

As shown in the table below, the dead interval is set to the same value (2 seconds) on all the three nodes. The VRRP communication intervals (hello interval and dead interval) of a node apply to all the VRIDs configured on the node, and in turn, apply to all VIP addresses associated with the VRIDs on the node.

On each node, the VIP addresses that are active (master) on that node use the hello interval, and the dead interval is used by the VIP addresses that are inactive (backup) on that node. Preemption is disabled for the VIP addresses in all the three nodes.

The following table lists the settings used in this example: VRRP interval example 1 settings.



The execution flow is as follows:

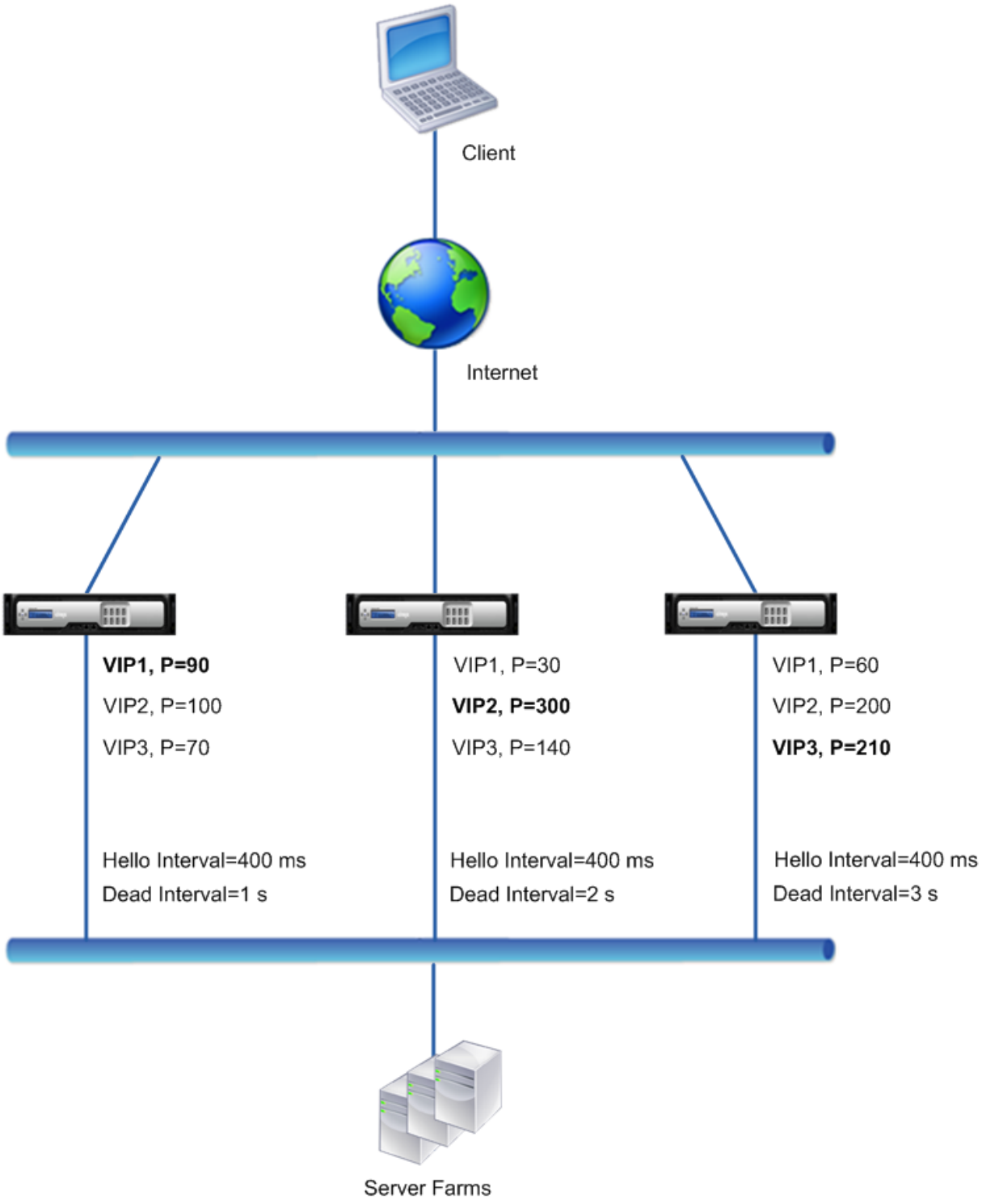
1. NS1 sends hello messages at a set hello interval of 400 ms to NS2 and NS3 for the VIP1 address, because VIP1 is active (the master) on NS1. Similarly, NS2 sends hello messages for VIP2, and NS3 sends hello messages for VIP3.

2. On NS1, the set dead interval applies to VIP2 and VIP3, because they are inactive (backups) on NS1. Similarly, on NS2, the set dead interval applies to VIP1 and VIP3, and on NS3, the set dead interval applies to VIP1 and VIP2.
3. If NS1 goes down, NS2 and NS3 consider NS1 to be down if they receive no hello messages from NS1 for 2 seconds (the dead interval). VIP1 on NS3 takes over and becomes active (master) because its VRID priority (60) is higher than that of VIP1 of NS2 (30).

Example 2: Nodes with Different VRRP Dead Intervals

Consider a VRRP deployment similar to the deployment described in Example1 but with a different dead interval on each node (NS1, NS2, and NS3). Preemption is disabled for the VIP addresses in all the three nodes.

The following table lists the settings used in this example: VRRP interval example 2 settings.



The execution flow is as follows when NS1 goes down:

1. NS2 considers NS1 to be down after not receiving any hello messages from NS1 for 2 seconds (NS2's dead interval).
2. VIP1 on NS2 takes over and becomes active (master). NS2 now starts sending hello messages

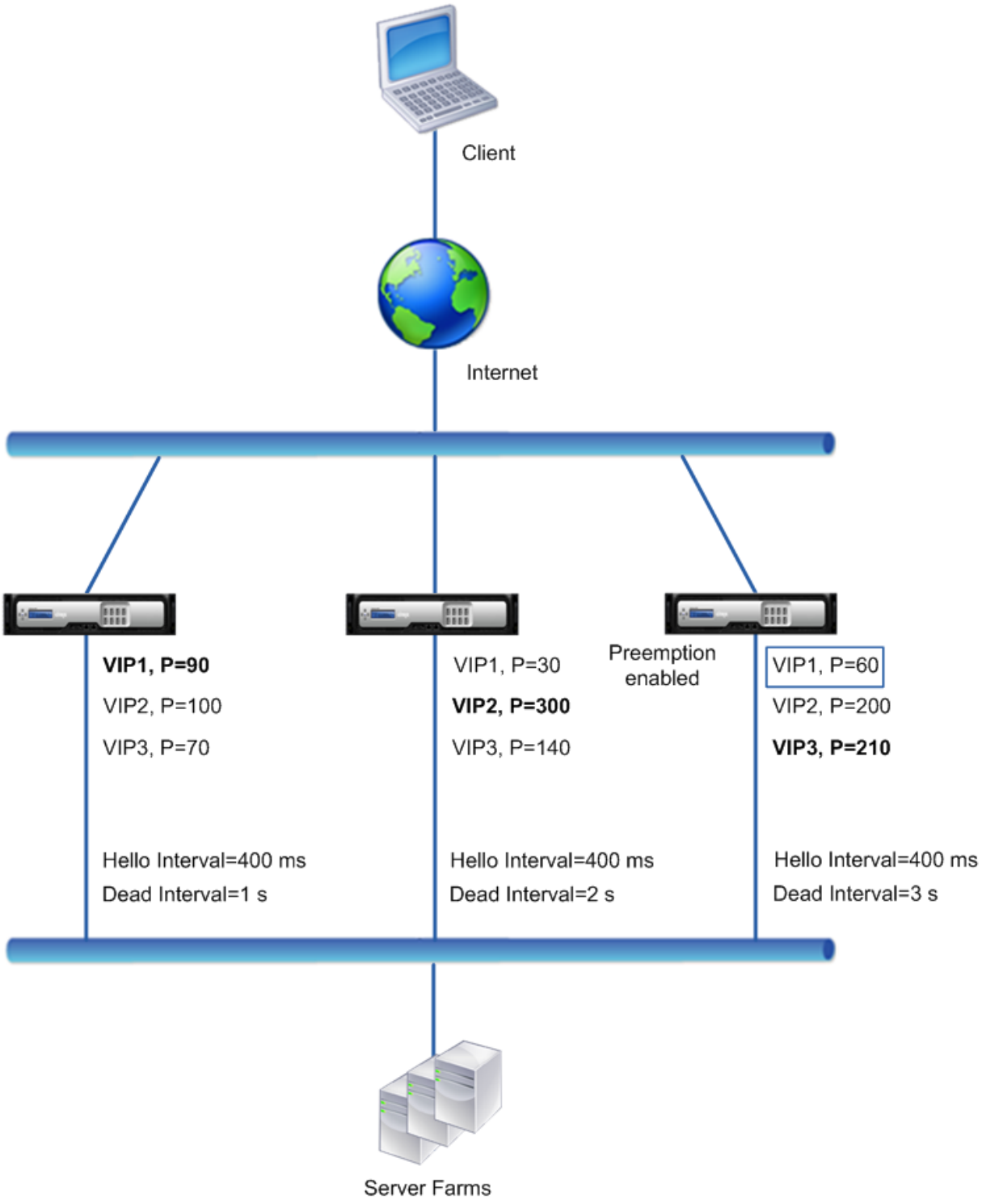
for VIP1.

Even though VIP1 on NS3 has a higher VRIP priority (60) than does VIP1 on NS2 (30), NS3's larger dead interval (3 seconds, vs. 2 seconds for NS2), prevents VIP1 on NS3 from taking over before VIP 1 on NS2 has already done so.

Example 3: Nodes with Different Dead Intervals and Preemption Enabled

Consider a VRRP deployment similar to the deployment described in Example1 but with different dead intervals on the three nodes, NS1, NS2, and NS3, and with preemption enabled for the VIP1 address on NS3.

The following table lists the settings used in this example: VRRP interval example 3 settings.



The execution flow is as follows when NS1 goes down:

1. NS2 considers NS1 to down after not receiving any hello messages from NS1 for 2 seconds (NS2's set dead interval). At this time, NS3, with a dead interval of 3 seconds, does not consider NS1 to be down.

2. VIP1 on NS2 takes over and becomes active (master). NS2 now starts sending hello messages for VIP1.
3. Upon receiving hello messages from NS2 for VIP1, NS3 preempts NS2 for VIP1 because preemption is enabled for VIP1 of NS3 and the VRID priority (60) of VIP1 of NS3 is higher than that (30) of VIP1 of NS2.
4. VIP1 on NS3 takes over and becomes active (master). NS3 now starts sending hello messages for VIP1.

Configuring Health Tracking based on Interface State

September 14, 2021

To ensure that a backup VIP address takes over as the master VIP before the node of the current master VIP address goes down completely, you can configure a node to change the priority of a VIP address when the state of an interface on the node changes. For example, the node reduces the priority of a VIP address when the state of an interface changes to DOWN, and increases the priority when the state of the interface changes to UP. This feature is a per node configuration for each VIP address.

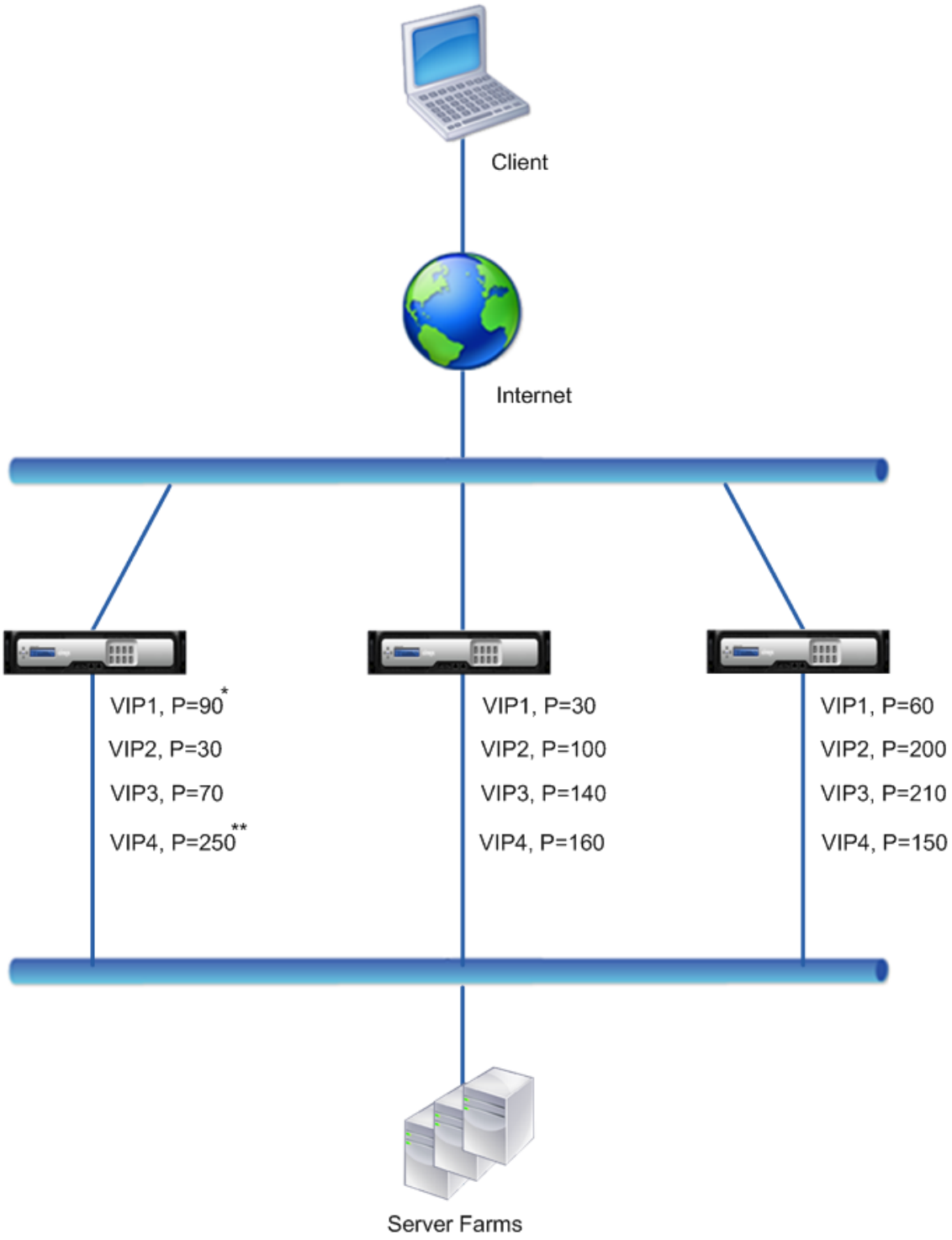
Example

Consider an active-active deployment consisting of Citrix ADCs NS1, NS2, and NS3. Virtual IP addresses VIP1, VIP2, VIP3, and VIP4 are configured on each of these ADCs. Because of their priorities, VIP1 and VIP4 are active on NS1, VIP2 is active on NS2, and VIP3 is active on NS3.

To ensure that the active VIP addresses on NS1 are taken over by either NS2 or NS3 before NS1 goes down completely, interface-based health tracking is configured for the VIP1 and VIP4 addresses on NS1. Configuring interface-based health tracking for a VIP address includes associating the desired interfaces and setting the reduced priority (trackifNumPriority) parameter for the associated VRID of the VIP address. For example, on NS1, interfaces 1/2, 1/3, and 1/5 are associated to the VRID of VIP1, and reduced priority is set to 20.

Preemption is enabled for these VIP addresses in all three nodes.

The following table lists the settings used in this example: Health tracking example settings.



* Packet Interfaces = 1/2, 1/3, 1/5
Reduced Priority = 20

** Packet Interfaces = 1/5, 1/7
Reduced Priority = 55

The execution flow is as follows on NS1 when multiple interface on NS1 goes down:

1. If interface 1/3 goes down, the priority of address VIP1 is reduced by 20 (VIP1's reduced priority value), because interface 1/3 is associated with VIP1:
   - Effective priority of VIP1 = (Current priority - reduced priority) = (90-20) = 70
2. Similarly, if interface 1/5 goes down, the priority of address VIP1 is further reduced:
   - Effective priority of VIP1 = (Current priority - reduced priority) = (70-20) = 50
3. At this point, the effective priority of VIP1 on NS1 is less than the priority of VIP1 on NS3. NS3 preempts NS1 for VIP1. VIP1 on NS3 takes over and becomes active (master).
4. Also, because interface 1/5 is also associated with VIP4, the priority of VIP4 is reduced by the VIP4's reduced priority value (55).
   - Effective priority of VIP4 = (250 - 55) = 195
5. If interface 1/7 goes down, the priority of VIP4 is further reduced:
   - Effective priority of VIP4 = (Current priority - reduced priority) = (195-55) = 145
6. At this point, the effective priority of VIP4 on NS1 is less than the priority of VIP4 on NS2. NS2 preempts NS1 for VIP4. VIP4 on NS3 takes over and becomes active (master). This configuration ensures that none of the four VIP addresses are active on NS1 before it completely goes down.

Configuration Steps for IPv4 Active-Active Mode

To configure this feature on a node for a VIP address, you set the Reduced Priority (trackifNumPriority) parameter, and then associate the interfaces whose state is to be tracked for changing the priority of the VIP address. When any of the associated interface's state changes to DOWN or UP, the node reduces or increases the priority of the VIP address by the configured Reduced Priority (trackifNumPriority) value.

To set reduced priority and bind interfaces to the virtual router ID by using the CLI:

At the command prompt, type:

- **set vrID** <id> [-**trackifNumPriority** <positive_integer>]
- **bind vrID** <id> -**trackifNum** <interface_name>
- **show vrID** <id>

Example:

```
    > set vrID 125 -trackifNumPriority 10
    Done

    > bind vrID 125 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```

To set reduced priority and bind interfaces to the virtual router ID by using the GUI:

1. Navigate to **System** > **Network** > **VMAC.**
2. On the **VMACs** tab, select a virtual router ID, and click **Edit**.
3. Under **Configure virtual MAC**, set the **Reduced Priority** parameter.
4. Select the **Interfaces tracked for the VRID** option and, under **Associate Interfaces**, add interfaces to the virtual router ID.

Configuration Steps for IPv6 Active-Active Mode

To configure this feature on a node for a VIP6 address, you set the Reduced Priority (trackifNumPriority) parameter, and then associate the interfaces whose state is to be tracked for changing the priority of the VIP6 address. When any of the associated interface's state changes to DOWN or UP, the node reduces or increases the priority of the VIP6 address by the configured Reduced Priority (trackifNumPriority) value.

To change the priority of a VIP address automatically by using the CLI:

At the command prompt, type one of the following sets of commands.

- If adding a new virtual MAC6:
  - **add vrID6** <id> [-**trackifNumPriority** <positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>
- If reconfiguring an existing virtual MAC6:
  - **set vrID6** <id> [-**trackifNumPriority** <positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>

Example:

```
    > set vrID6 130 -trackifNumPriority 10
    Done

    > bind vrID6 130 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```

Delaying Preemption

September 14, 2021

By default, a backup VIP address preempts the master VIP address immediately after its priority becomes higher than that of the master VIP. When configuring a backup VIP address, you can specify an amount of time by which to delay the preemption. Preemption delay time is a per-node setting for each backup VIP address.

The preemption delay setting for a backup VIP does not apply in the following conditions:

- The node of the master VIP goes down. In this case, the backup VIP takes over as the master VIP after the dead interval set on the backup VIP's node.
- The priority of the master VIP is set to zero. The backup VIP takes over as the master VIP after the dead interval set on the backup VIP's node.

Example: Delaying Preemption

Consider an active-active deployment consisting of Citrix ADC appliances NS1 and NS2. Virtual IP address VIP1 is configured on each of these appliances. Because of their priorities, VIP1 is master on NS2. Preemption is enabled and preemption delay time is set for VIP1 on these two nodes.

The following table lists the settings used in this example.

Entity and Parameters	Settings on NS1	Settings on NS2
VIP1 (for reference purposes only)	**IP address:** 192.0.1.10, **VRID:** 10, **Priority:** 100, **Preemption:** Enabled, **Preemption delay time:** 1000 seconds	**IP address:** 192.0.1.10, **VRID: 10**, **Priority:** 200, **Preemption:** Enabled, **Preemption delay time:** 2000 seconds
Dead Interval	1 Seconds	2 Seconds

Following are some examples of possible preemption behavior in this setup:

- If the priority of VIP1 on NS1 is set to a value (for example, 210) higher than that of VIP1 on NS2, VIP1 on NS1 takes over as master after its set preemption delay time (1000 secs).
- If a third node NS3 with the following VRRP settings is added to this deployment, VIP1 on NS3 becomes master after its set preemption delay time (3000 secs).
  - VIP1
    - VRID: 30
    - IP address:
    - Priority = 300
    - Preemption delay time = 3000 seconds
- If NS2 goes down, VIP1 on NS1 takes over as master after 1 second (set dead interval on NS1). Preemption delay time for VIP1 on NS1 does not apply in this case.

- If NS2 goes down and NS1 restarts, VIP1 on NS1 becomes master 1 second (set dead interval on NS1) after NS1 comes up. Preemption delay time for VIP1 on NS1 does not apply in this case.
- If the priority of VIP1 on NS2 is set to zero, VIP1 goes to standby mode. VIP1 on NS1 takes over as master after 1 second (set dead interval on NS1). Preemption delay time for VIP1 on NS1 does not apply in this case.

Configuring Delay Preemption for IPv4 Active-Active Mode

To configure preemption delay time for a VIP address, you set the preemption delay timer parameter of the associated virtual MAC address. You can the set this parameter when you add the address, or you can modify an existing virtual MAC address.

To configure preemption delay time by using the CLI:

- To set the preemption delay time while adding a virtual MAC, at the command prompt, type:
  - **add vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**
- To set the preemption delay time while modifying a virtual MAC, at the command prompt, type:
  - **set vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

To configure preemption delay time by using the GUI:

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab. While adding a new virtual MAC, or editing an existing virtual MAC, set the **Preemption Delay Timer** parameter.

Sample configuration:

The following configuration uses the settings listed in table in section Example: Delaying Preemption.

```
1     Settings on NS1
2
3     > set vrid param  -deadInterval 1
4
5     Done
6
7     > add ns ip 192.0.1.10 255.255.255.255  -type VIP
8
9     Done
10
11    > add vrid 10  -Priority 100  -Preemption Enable  -
         preemptiondelaytimer 1000
12
13    Done
14
```



```
15      > bind ns ip 192.0.1.10 255.255.255.255 -vrid 10
16
17      Done
18
19      Settings on NS2
20
21      > set vrid param -deadInterval 2
22
23      Done
24
25      > add ns ip 192.0.1.10 255.255.255.255 -type VIP
26
27      Done
28
29      > add vrid 20 -Priority 200 -Preemption Enable -
            preemptiondelaytimer 2000
30
31      Done
32
33      > set ns ip 192.0.1.10 255.255.255.255 -vrid 10
34
35      Done
36  <!--NeedCopy-->
```

Configuring Delay Preemption for IPv6 Active-Active Mode

To configure preemption delay time for a VIP6 address, you set the preemption delay timer parameter of the associated virtual MAC6 address. You can the set this parameter when you add the virtual MAC6 address, or you can modify an existing virtual MAC6 address.

To configure preemption delay time by using the CLI:

- To set the preemption delay time while adding a virtual MAC6, at the command prompt, type:
  - **add vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**
- To set the preemption delay time while modifying a virtual MAC6, at the command prompt, type:
  - **set vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**

To configure preemption delay time by using the GUI:

1. Navigate to **System** > **Network** > **VMAC**.

2. On the **VMAC6** tab. While adding a virtual MAC6 address, or editing an existing virtual MAC6 address, set the **Preemption Delay Timer** parameter.

Keeping a VIP Address in the Backup State

September 14, 2021

You can force a VIP address to always stay in backup state. This operation is helpful in maintenance or testing of a VRRP deployment.

When a VIP address is forced to stay in backup state, it does not participate in VRRP state transitions. Also, it cannot become master even if all other nodes go down.

To force a VIP address to stay in backup state, you set the priority of the associated virtual MAC address to zero. To ensure that none of the VIP addresses of a node handle traffic during a maintenance process on the node, set all the priorities to zero.

You can the set the priority of a virtual MAC address while adding or modifying the address.

To force a VIP address to stay in the backup state by using the CLI:

- To set the priority while adding a virtual MAC, at the command prompt, type:
  - **add vrID** <id> -**priority** 0
  - **show vrID**
- To set the priority while modifying a virtual MAC, at the command prompt, type:
  - **set vrID** <id> -**priority** 0
  - **show vrID**

To force a VIP address to stay in backup state by using the GUI:

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab, while adding a new virtual MAC or editing an existing virtual MAC, set the **Priority** parameter to zero.

Network Visualizer

September 14, 2021

The network visualizer shows a graphical view of all the interfaces, channels, VLANs, IP addresses, and bindings to VLANs on a Citrix ADC appliance. An enabled interface or channel has a black label. A disabled interface or channel has a red label.

This complete picture of the appliance's network connections can be useful for detecting flaws in the network design and for optimizing the network. It can also help a new administrator easily understand the appliance's network configuration.

To open the Network Visualizer:

Navigate to **System** > **Network**. In **Monitor Connections**, click **Network Visualizer**.

Configuring Link Layer Discovery Protocol

September 14, 2021

The Citrix ADC supports the industry standard (IEEE 802.1AB) Link Layer Discovery Protocol (LLDP). LLDP is a layer 2 protocol that enables the Citrix ADC to advertise its identity and capabilities to the directly connected devices, and also learn the identity and capabilities of these neighbour devices.

> **Note:**
> Link Layer Discovery Protocol (LLDP) is supported only in Citrix ADC MPX platforms.

Using LLDP, the Citrix ADC transmits and receives information in the form of LLDP messages known as LLDP packet data units (LLDPUs). An LLDPU is a sequence of type, length, value (TLV) information elements. Each TLV holds a specific type of information about the device that transmits the LLDPDU. The Citrix ADC sends the following TLVs in each LLDPU:

- Chassis ID
- Port ID
- Time-to-live value
- System name
- System description
- Port description
- System capabilities
- Management address
- Port VLAN ID
- Link aggregation

Note: You cannot specify the TLVs to be sent in LLDP messages.

Citrix ADC interfaces support the following LLDP modes:

- **NONE**. The interface neither receives from nor transmits LLDP messages to the directly connected device.
- **TRANSMITTER**. The interface transmits LLDP messages to the directly connected device but does not receive LLDP messages from the directly connected device.

- **RECEIVER**. The interface receives LLDP messages from the directly connected device but does not transmit LLDP messages to the directly connected device.
- **TRANSCEIVER**. The interface transmits LLDP messages to and receives LLDP messages from the directly connected device.

The LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. The following table shows the modes resulting from the available combinations of global- and interface-level settings: Interface and global level LLDP modes.

Note the following points related to LLDP messages transmitted or received by the Citrix ADC:

- **Transmitting LLDP messages.** The Citrix ADC transmits LLDPUs from interfaces that are operating in either TRANSMITTER or TRANSCEIVER LLDP mode.

  Following are the global LLDP transmitting parameters on the Citrix ADC:

  - **Timer**. Interval, in seconds, between LLDPUs that the Citrix ADC sends to a directly connected device.
  - **Holdtime Multiplier**. A multiplier for calculating the duration for which the receiving device stores the LLDP information in its database before discarding or removing it. The duration is calculated as the **Holdtime Multiplier** parameter value multiplied by the Timer parameter value.

- **Receiving LLDP Messages.** The Citrix ADC stores the LLDPDU information in its Management Information base (MIB). The stored LLDP information is classified or grouped under the ID of the interface that received the LLDPDU. The Citrix ADC retains this LLDP information for the duration specified in the received LLDPU.

  If the ADC receives another LLDPDU on an interface before the stored LLDP information for that interface is discarded, the ADC replaces the stored LLDP information for that interface with information in the new LLDPDU.

Configuration Steps

Configuring LLDP on a Citrix ADC appliance consists of the following tasks:

1. **Set global level LLDP parameters.** In this task, you set the global LLDP parameters such as LLDP Timer, Hold Time Multiplier, and LLDP mode.
2. **Set the interface level LLDP parameters.** In this task, you set the LLDP mode for an interface.
3. **(Optional) Display neighbor-device information.** You can display the neighbor-device LLDP information collected on all of the Citrix ADC's interfaces, or just the LLDP information collected on specified interfaces. If you do not specify an interface, the information is shown for all interfaces.

Following are the prerequisites for configuring LLDP on a Citrix ADC:

1. Make sure that you understand the standard LLDP protocol (IEEE 802.1AB).
2. Verify that you have configured LLDP on the desired directly connected devices.

CLI procedures

To set global level LLDP parameters by using the CLI:

At the command prompt, type:

- set lldp param [-[-holdtimeTxMult <positive_integer>][-timer <positive_integer>] [-Mode <Mode>]
- show lldp param

To configure an interface for LLDP by using the CLI:

At the command prompt, type:

- set interface <id> -lldpmode <lldpmode>
- show interface <id>

To display neighbor device information by using the CLI:

At the command prompt, type one of the following commands:

- show lldp neighbors
- show lldp neighbors <ifnum>

GUI procedures

To set the global level LLDP parameters by using the GUI:

1. Navigate to System > Network, and click Configure LLDP Parameters.
2. Set the following parameters:
    - Hold Timer Multiplier
    - Timer
    - Mode

To configure an interface for LLDP by using the GUI:

Navigate to System > Network > Interfaces, open the interface, and set the LLDP mode parameter.

To display neighbor device information by using the GUI:

Navigate to System > Network > Interfaces and, in the Action list, select View LLDP Neighbors.

LLDP Support in a Cluster Setup

In a cluster setup, the GUI and CLI display the LLDP neighbour configuration of all or specific cluster nodes when the GUI or CLI is accessed through the Cluster IP address (CLIP). Any change made to the global level LLDP mode is applied to the global level LLDP mode on each of the cluster nodes.

Consider an example of a cluster setup of three nodes, NS1, NS2, and NS3. Each of these nodes are connected to both routers Router-1 and Router-2. The following output is displayed when the **show lldp neighbor -summary** operation is performed on the Cluster CLI that is accessed through the Cluster IP address (CLIP) of the cluster setup. The output shows the LLDP neighbour information of all these nodes.

```
> show lldp neighbor -summary

Node Id: 1
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     1/1/1      fe:c7:3b:13:bd:11    1/1         Router-1

2     1/1/2      12:68:7b:9e:4c:11    1/1         Router-2

Node Id: 2
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     2/1/1      fe:c7:3b:13:bd:12    1/2        Router-1

2     2/1/2      12:68:7b:9e:4c:12    1/2        Router-2

Node Id: 3
------------------------------------------------------------
    Interface    ChassisId            PortId     System name
------------------------------------------------------------

1     3/1/1       fe:c7:3b:13:bd:13    1/3        Router-1

2     3/1/2       12:68:7b:9e:4c:13    1/3        Router-2

Done
<!--NeedCopy-->
```


Jumbo Frames

September 14, 2021

Citrix ADC appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A Citrix ADC appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

The Citrix ADC appliance supports jumbo frames in a load balancing configuration for the following protocols:

- TCP
- Any protocol over TCP (for example, HTTP)
- SIP
- RADIUS

Configuring Jumbo Frames Support on a Citrix ADC Appliance

September 14, 2021

To enable the Citrix ADC appliance to support jumbo frames, you set the MTU to more than 1500 on interfaces or LA channels, and on VLANs on which you want the Citrix ADC appliance to support jumbo frames.

Points to consider before setting the MTU of interfaces, LA channels, or VLANs on a Citrix ADC appliance

1. When you create an LA channel, the channel takes the MTU of the first bound interface if no MTU is specified for the channel.
2. The MTU for a channel is propagated to all the bound interfaces.
3. When an interface is bound to the channel whose MTU is different from the interface's MTU, the interface goes onto the inactive list.
4. When you change the MTU of a member interface, the interface goes onto the inactive list.

5. When an interface is unbound from the channel, the interface retains the MTU value of the channel.
6. You can set the MTU for an interface, channel, or VLAN to a value in the range of 1500-9216.
7. You cannot set the MTU on the default VLAN. The Citrix ADC appliance uses the MTU of the interface through which it receives or sends data from or to the default VLAN.
8. For TCP based traffic on a load balancing configuration on a Citrix ADC appliance, MSSs are set accordingly at each end point for supporting jumbo frames:
    - For a connection between a client and a load balancing virtual server on the Citrix ADC appliance, the MSS on the Citrix ADC appliance is set in a TCP profile, which is then bound to the load balancing virtual server.
    - For a connection between the Citrix ADC appliance and a server, the MSS on NS1 is set in a TCP profile, which is then bound to the service representing the server on the Citrix ADC appliance.
    - By default, a TCP profile nstcp_default_profile is bound to all TCP based load balancing servers and services on the Citrix ADC appliance.
    - For supporting jumbo frames, you can either change the MSS value of the TCP profile nstcp_default_profile, or create a custom TCP profile and set its MSS accordingly, and then bind the custom TCP profile to the desired load balancing virtual servers and services.
    - The default MSS value of any TCP profile is 1460.

CLI procedures

To set the MTU of an interface by using the CLI:

At the command prompt, type:

- set interface <id> -mtu <positive_integer>
- show interface <id>

Example:

```
> set interface 10/1  - mtu 9000
Done
<!--NeedCopy-->
```

To set the MTU of a channel by using the CLI:

At the command prompt, type:

- set channel <id> -mtu <positive_integer>
- show channel <id>

Example:


```
> set channel LA/1  - mtu 9000
Done
<!--NeedCopy-->
```

To set the MTU of a VLAN by using the CLI:

At the command prompt, type:

- add vlan <id> -mtu <positive_integer>
- show vlan <id>

Example:

```
> set vlan 20  - mtu 9000
Done
<!--NeedCopy-->
```

GUI procedures

To set the MTU of an interface by using the GUI:

Navigate to System > Network > Interfaces, open the interface, and set the Maximum Transmission Unit parameter.

To set the MTU of a channel by using the GUI:

Navigate to System > Network > Channels, open the channel, and set the Maximum Transmission Unit parameter.

To set the MTU of a VLAN by using the GUI:

Navigate to System > Network > VLANs, open the VLAN, and set the Maximum Transmission Unit parameter.

Use Case 1 – Jumbo to Jumbo Setup

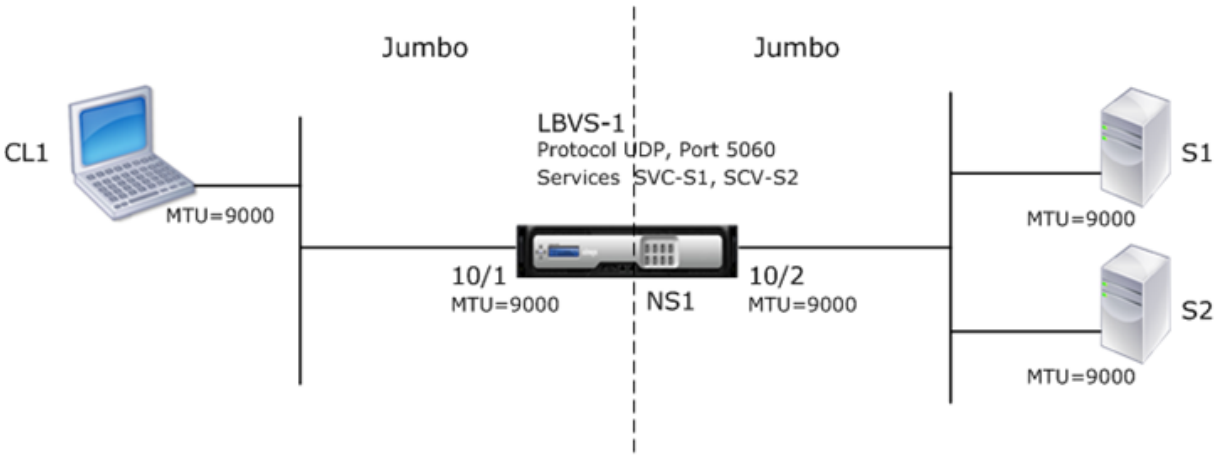
September 14, 2021

Consider an example of a jumbo to jumbo setup in which SIP load balancing virtual server LBVS-1, configured on Citrix ADC appliance NS1, is used to load balance SIP traffic across servers S1 and S2. The connection between client CL1 and NS1, and the connection between NS1 and the servers support jumbo frames.

Interface 10/1 of NS1 receives or sends traffic from or to client CL1. Interface 10/2 of NS1 receives or sends traffic from or to server S1 or S2. Interfaces 10/1 and 10/2 of NS1 are part of VLAN 10 and VLAN 20, respectively.

For supporting jumbo frames, the MTU is set to 9216, on NS1, for interfaces 10/1, 10/2, and VLANs VLAN 10, VLAN 20.

All other network devices, including CL1, S1, S2, in this setup example are also configured for supporting jumbo frames.



The following table lists the settings used in the example.

Entity	Name	Details
IP address of client CL1	-	192.0.2.10
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20
SNIP address on NS1		198.51.100.18
MTU specified for interfaces and VLANs on NS1	10/1	9000
	10/2	9000
	VLAN 10	9000
	VLAN 20	9000
Services on NS1 representing servers	SVC-S1	**IP address:** 198.51.100.19, **Protocol:** SIP, **Port:** 5060
	SVC-S2	**IP address:** 198.51.100.20, **Protocol:** SIP, **Port:** 5060

Entity	Name	Details
Load balancing virtual server on VLAN 10	LBVS-1	**IP address:** 203.0.113.15, **Protocol:** SIP, **Port:** 5060, **Bound services:** SVC-S1, SVC-S2

Following is the traffic flow of CL1's request to NS1:

1. CL1 creates a 20000-byte SIP request to send to LBVS-1 of NS1.
2. CL1 sends the request data in IP fragments to LBVS-1. The size of each IP fragment is either equal to or less than the MTU (9000) set on the interface from which CL1 sends these fragments to NS1.
    - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
    - Size of the second IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
    - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 2048] = 2068
3. NS1 receives the request IP fragments at interface 10/1. NS1 accepts these fragments, because the size of each of these fragments is equal to or less than the MTU (9000) of interface 10/1.
4. NS1 reassembles these IP fragments to form the 20000-byte SIP request. NS1 processes this request.
5. LBVS-1's load balancing algorithm selects server S1.
6. NS1 sends the request data in IP fragments to S1. The size of each IP fragment is either equal or less than the MTU (9000) of the interface 10/2, from which NS1 sends these fragments to S1. The IP packets are sourced with a SNIP address of NS1.
    - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
    - Size of the second IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
    - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 2048] = 2068

Following is the traffic flow of S1's response to CL1 in this example:

1. Server S1 creates a 30000-byte SIP response to send to the SNIP address of NS1.
2. S1 sends the response data in IP fragments to the SNIP address of NS1. The size of each IP fragment is either equal to or less than the MTU (9000) set on the interface from which S1 sends these fragments to NS1.
    - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000

- Size of the second and third IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
- Size of the last IP fragment=[IP header + SIP data segment] = [20 + 3068] = 3088

3. NS1 receives the response IP fragments at interface 10/2. NS1 accepts these fragments, because the size of each fragment is equal to or less than the MTU (9000) of interface 10/2.
4. NS1 reassembles these IP fragments to form the 30000-byte SIP response. NS1 processes this response.
5. NS1 sends the response data in IP fragments to CL1. The size of each IP fragment is either equal or less than the MTU (9000) of the interface 10/1, from which NS1 sends these fragments to CL1. The IP fragments are sourced with LBVS-1's IP address.
   - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
   - Size of the second and third IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
   - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 3068] = 3088

Configuration Tasks

The following table list the tasks, Citrix ADC commands, and examples for creating the required configuration on the Citrix ADC appliance.

Task	Citrix ADC Command Syntax	Example
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1 -mtu 9000 set int 10/2 -mtu 9000
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 9000 add vlan 20 -mtu 9000
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name>, show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP

Task	Citrix ADC Command Syntax	Example
Create services representing SIP servers	add service <serviceName> <ip> SIP_UDP <port>, show service <name>	add service SVC-S1 198.51.100.19 SIP_UDP 5060 add service SVC-S2 198.51.100.20 SIP_UDP 5060
Create SIP load balancing virtual servers and bind the services to it	add lb vserver <name> SIP_UDP <ip> <port> bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 SIP_UDP 203.0.113.15 5060 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
Save the configuration	save ns config, show ns config	

Use Case 2 – Non-Jumbo to Jumbo Setup

September 14, 2021

Consider an example of a regular to jumbo setup in which load balancing virtual server LBVS-1, configured on a Citrix ADC appliance NS1, is used to load balance traffic across servers S1 and S2. The connection between client CL1 and NS1 supports regular frames, and the connection between NS1 and the servers supports jumbo frames.

Interface 10/1 of NS1 receives or sends traffic from or to client CL1. Interface 10/2 of NS1 receives or sends traffic from or to server S1 or S2.

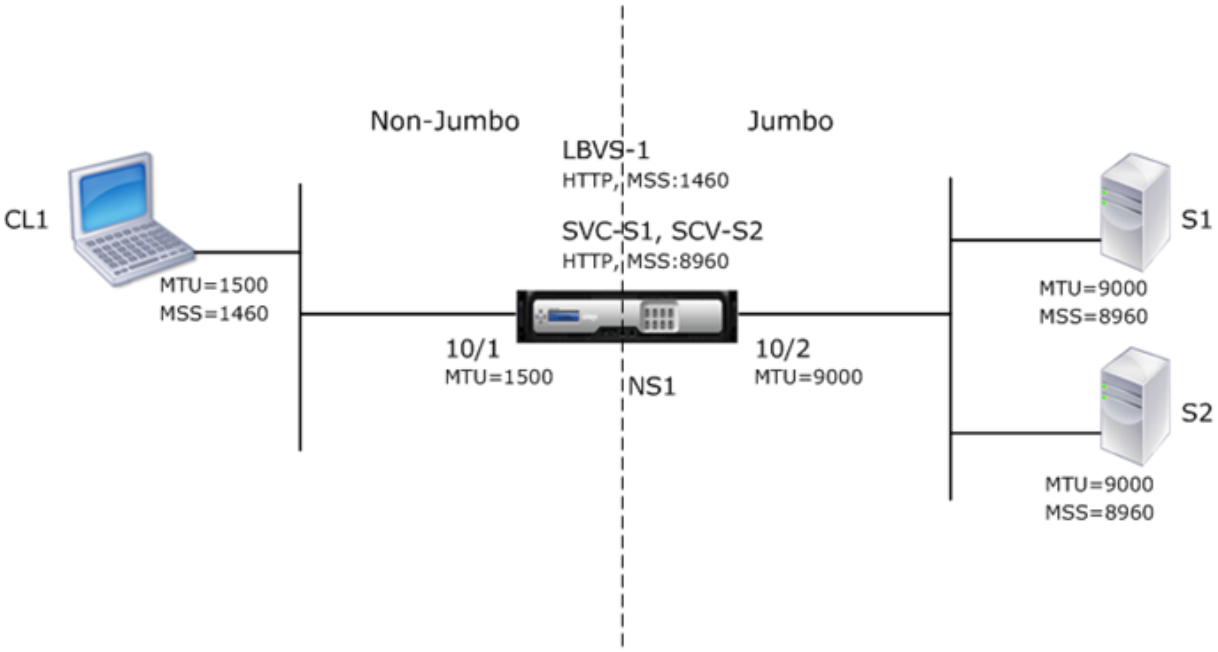
Interfaces 10/1 and 10/2 of NS1 are part of VLAN 10 and VLAN 20, respectively. For supporting only regular frames between CL1 and NS1, the MTU is set to the default value of 1500 for both interface 10/1 and VLAN 10

For supporting jumbo frames between NS1 and the servers, the MTU is set to 9000 for interface 10/2 and VLAN 20. Servers and all other network devices between NS1 and the servers are also configured for supporting jumbo frames.

Since HTTP traffic is based on TCP, MSSs are set accordingly at each end point for supporting jumbo frames.

- For supporting jumbo frames for the connection between a SNIP address of NS1 and S1 or S2, the MSS on NS1 is set accordingly in a custom TCP profile, which is bound to the services (SVC-S1 and SVC-S1) representing S1 and S2 on NS1.
- For supporting only regular frames for the connection between CL1 and virtual server LBVS-1 of NS1, default TCP profile nstcp_default_profile is used that is by default bound to LBVS-1 and

has the MSS set to the default value of 1460.



The following table lists the settings used in this example.

Entity	Name	Details
IP address of client CL1		192.0.2.10
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20
SNIP address on NS1		198.51.100.18
MTU specified for interfaces and VLANs on NS1	10/1	1500
	10/2	9000
	VLAN 10	1500
	VLAN 20	9000
Default TCP profile	nstcp_default_profile	MSS:1460
Custom TCP profile	NS1-SERVERS-JUMBO	MSS: 8960
Services on NS1 representing servers	SVC-S1	IP address: 198.51.100.19, Protocol: HTTP, Port: 80, TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)

Entity	Name	Details
	SVC-S2	IP address: 198.51.100.20, Protocol: HTTP, Port: 80, TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)
Load balancing virtual server on VLAN 10	LBVS-1	IP address = 203.0.113.15, Protocol: HTTP, Port:80, Bound services: SVC-S1, SVC-S2, TCP Profile: nstcp_default_profile (MSS:1460)

Following is the traffic flow of CL1's request to S1 in this example:

1. Client CL1 creates a 200-byte HTTP request to send to virtual server LBVS-1 of NS1.
2. CL1 opens a connection to LBVS-1 of NS1. CL1 and NS1 exchange their respective TCP MSS values while establishing the connection.
3. Because NS1's MSS is larger than the HTTP request, CL1 sends the request data in a single IP packet to NS1.

   Size of the request packet = [IP Header + TCP Header + TCP Request] = [20 + 20 + 200] = 240
4. NS1 receives the request packet at interface 10/1 and then processes the HTTP request data in the packet.
5. LBVS-1's load balancing algorithm selects server S1, and NS1 opens a connection between one of its SNIP addresses and S1. NS1 and CL1 exchange their respective TCP MSS values while establishing the connection.
6. Because S1's MSS is larger than the HTTP request, NS1 sends the request data in a single IP packet to S1.

   Size of the request packet = [IP Header + TCP Header + [TCP Request] = [20 + 20 + 200] = 240

Following is the traffic flow of S1's response to CL1 in this example:

1. Server S1 creates an 18000-byte HTTP response to send to the SNIP address of NS1.
2. S1 segments the response data into multiples of NS1's MSS and sends these segments in IP packets to NS1. These IP packets are sourced from S1's IP address and destined to the SNIP address of NS1.
   - Size of the first two packet = [IP Header + TCP Header + (TCP segment=NS1's MSS size)] = [20 + 20 + 8960] = 9000

- Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120

3. NS1 receives the response packets at interface 10/2.
4. From these IP packets, NS1 assembles all the TCP segments to form the HTTP response data of 18000 bytes. NS1 processes this response.
5. NS1 segments the response data into multiples of CL1's MSS and sends these segments in IP packets, from interface 10/1, to CL1. These IP packets are sourced from LBVS-1's IP address and destined to CL1's IP address.
   - Size of all packets except the last one = [IP Header + TCP Header + (TCP payload=CL1's MSS size)] = [20 + 20 + 1460] = 1500
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 480] = 520

Configuration Tasks

The following table list the tasks, Citrix ADC commands, and examples for creating the required configuration on the Citrix ADC appliance.

Tasks	CLI Syntax	Examples
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1 -mtu 1500 set int 10/2 -mtu 9000
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 1500 add vlan 20 -mtu 9000
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name>, show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
Create services representing HTTP servers	add service <serviceName> <ip> HTTP <port>, show service <name>	add service SVC-S1 198.51.100.19 http 80, add service SVC-S2 198.51.100.20 http 80

Tasks	CLI Syntax	Examples
Create HTTP load balancing virtual servers and bind the services to it	add lb vserver <name> HTTP <ip> <port>, bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 http 203.0.113.15 80, bind lb vserver LBVS-1 SVC-S1, bind lb vserver LBVS-1 SVC-S2
Create a custom TCP profile and set its MSS for supporting jumbo frames	add tcpProfile <name> -mss <positive_integer>, show tcpProfile <name>	add tcpprofile NS1-SERVERS-JUMBO -mss 8960
Bind the custom TCP profile to the desired services	set service <Name> -tcpProfileName <string>, show service <name>	set service SVC-S1 -tcpProfileName NS1-SERVERS-JUMBO, set service SVC-S2 -tcpProfileName NS1-SERVERS-JUMBO
Save the configuration	save ns config, show ns config	

Use Case 3 – Coexistence of Jumbo and Non-Jumbo flows on the Same Set of Interfaces

September 14, 2021

Consider an example in which load balancing virtual servers LBVS-1 and LBVS-2 are configured on Citrix ADC appliance NS1. LBVS-1 is used to load balance HTTP traffic across servers S1 and S2, and LBVS-2 is used to load balance traffic across servers S3 and S4.

CL1 is on VLAN 10, S1 and S2 are on VLAN20, CL2 is on VLAN 30, and S3 and S4 are on VLAN 40. VLAN 10 and VLAN 20 support jumbo frames, and VLAN 30 and VLAN 40 support only regular frames.

In other words, the connection between CL1 and NS1, and the connection between NS1 and server S1 or S2 support jumbo frames. The connection between CL2 and NS1, and the connection between NS1 and server S3 or S4 support only regular frames.

Interface 10/1 of NS1 receives or sends traffic from or to clients. Interface 10/2 of NS1 receives or sends traffic from or to the servers.

Interface 10/1 is bound to both VLAN 10 and VLAN 30 as a tagged interface, and interface 10/2 is bound to both VLAN 20 and VLAN 40 as a tagged interface.

For supporting jumbo frames, the MTU is set to 9216 for interfaces 10/1 and 10/2.

On NS1, the MTU is set to 9000 for VLAN 10 and VLAN 20 for supporting jumbo frames, and the MTU is set to the default value of 1500 for VLAN 30 and VLAN 40 for supporting only regular frames.

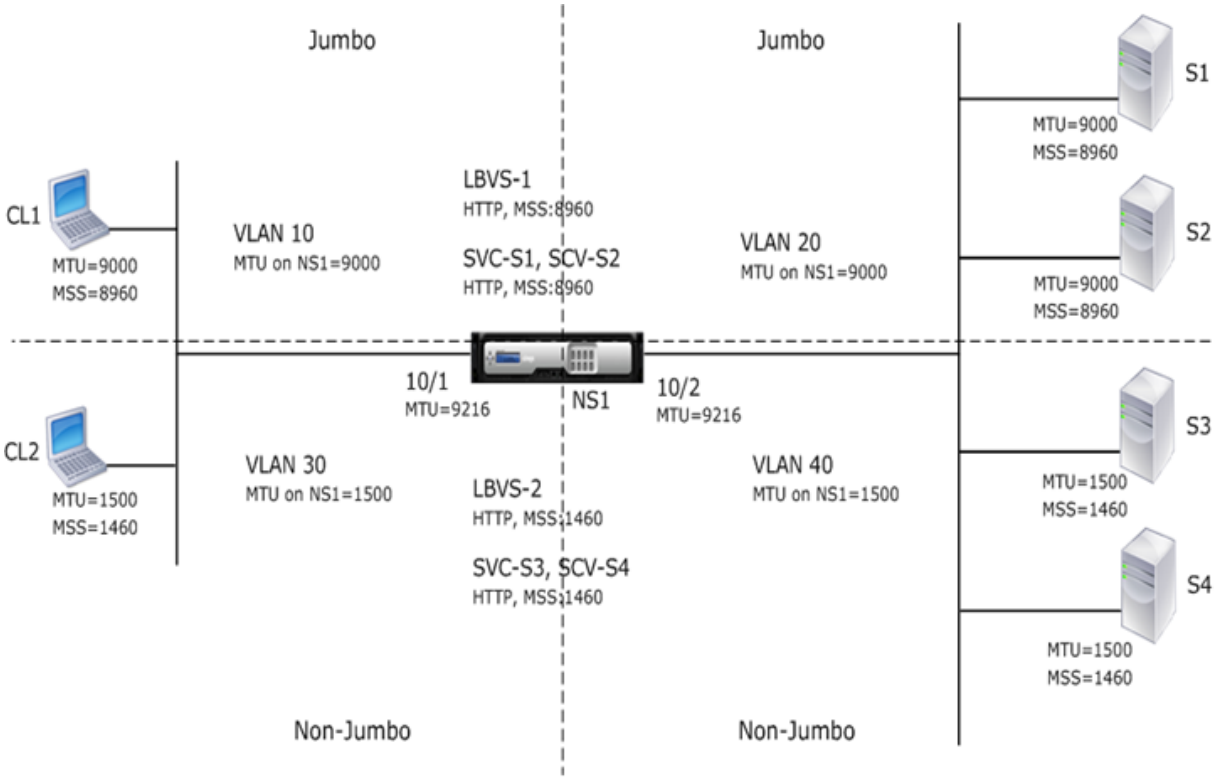
The effective MTU on a Citrix ADC interface for VLAN tagged packets is of the MTU of the interface or the MTU of the VLAN, whichever is lower. For example:

- The MTU of interface 10/1 is 9216. The MTU of VLAN 10 is 9000. On interface 10/1, the MTU of VLAN 10 tagged packets is 9000.
- The MTU of interface 10/2 is 9216. The MTU of VLAN 20 is 9000. On interface 10/2, the MTU of VLAN 20 tagged packets is 9000.
- The MTU of interface 10/1 is 9216. The MTU of VLAN 30 is 1500. On interface 10/1, the MTU of VLAN 30 tagged packets is 1500.
- The MTU of interface 10/2 is 9216. The MTU of VLAN 40 is 1500. On interface 10/2, the MTU of VLAN 40 tagged packets is 9000.

CL1, S1, S2, and all network devices between CL1 and S1 or S2 are configured for jumbo frames.

Since HTTP traffic is based on TCP, MSSs are set accordingly at each end point for supporting jumbo frames.

- For the connection between CL1 and virtual server LBVS-1 of NS1, the MSS on NS1 is set in a TCP profile, which is then bound to LBVS-1.
- For the connection between a SNIP address of NS1 and S1, the MSS on NS1 is set in a TCP profile, which is then bound to the service (SVC-S1) representing S1 on NS1.



The following table lists the settings used in this example: Jumbo frames use case 3 example settings.

Following is the traffic flow of CL1's request to S1:

1. Client CL1 creates a 20000-byte HTTP request to send to virtual server LBVS-1 of NS1.
2. CL1 opens a connection to LBVS-1 of NS1. CL1 and NS1 exchange their TCP MSS values while establishing the connection.
3. Because NS1's MSS value is smaller than the HTTP request, CL1 segments the request data into multiples of NS1's MSS and sends these segments in IP packets tagged as VLAN 10 to NS1.
   - Size of the first two packets = [IP Header + TCP Header + (TCP segment=NS1 MSS)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120
4. NS1 receives these packets at interface 10/1. NS1 accepts these packets because the size of these packets is equal to or less than the effective MTU (9000) of interface 10/1 for VLAN 10 tagged packets.
5. From the IP packets, NS1 assembles all the TCP segments to form the 20000-byte HTTP request. NS1 processes this request.
6. LBVS-1's load balancing algorithm selects server S1, and NS1 opens a connection between one of its SNIP addresses and S1. NS1 and CL1 exchange their respective TCP MSS values while establishing the connection.
7. NS1 segments the request data into multiples of S1's MSS and sends these segments in IP pack-

ets tagged as VLAN 20 to S1.

- Size of the first two packets = [IP Header + TCP Header + (TCP payload=S1 MSS)] = [20 + 20 + 8960] = 9000
- Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120

Following is the traffic flow of S1's response to CL1:

1. Server S1 creates a 30000-byte HTTP response to send to the SNIP address of NS1.
2. S1 segments the response data into multiples of NS1's MSS and sends these segments in IP packets tagged as VLAN 20 to NS1. These IP packets are sourced from S1's IP address and destined to the SNIP address of NS1.
   - Size of first three packet = [IP Header + TCP Header + (TCP segment=NS1's MSS size)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 3120] = 3160
3. NS1 receives the response packets at interface 10/2. NS1 accepts these packets, because their size is equal to or less than the effective MTU value (9000) of interface 10/2 for VLAN 20 tagged packets.
4. From these IP packets, NS1 assembles all the TCP segments to form the 30000-byte HTTP response. NS1 processes this response.
5. NS1 segments the response data into multiples of CL1's MSS and sends these segments in IP packets tagged as VLAN 10, from interface 10/1, to CL1. These IP packets are sourced from LBVS's IP address and destined to CL1's IP address.
   - Size of first three packet = [IP Header + TCP Header + [(TCP payload=CL1's MSS size)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 3120] = 3160

Configuration Tasks

Following table lists tasks, commands, and examples for creating the required configuration on the Citrix ADC appliance: Jumbo frames use case 3 configuration tasks.

Citrix ADC Support for Microsoft Direct Access Deployment

September 14, 2021

Microsoft Direct Access is a technology that enables remote users to seamlessly and securely connect to the enterprise's internal networks, without the need to establish a separate VPN connection. Un-

like VPN connections, which require user intervention to open and close connections, a Direct Access-enabled client connects automatically to the enterprise's internal networks whenever the client connects to the Internet.

Manage-Out is a Microsoft Direct Access feature that allows administrators inside the enterprise network to connect to Direct Access clients outside the network and manage them (for example, performing administration tasks, such as scheduling service updates, and providing remote support.

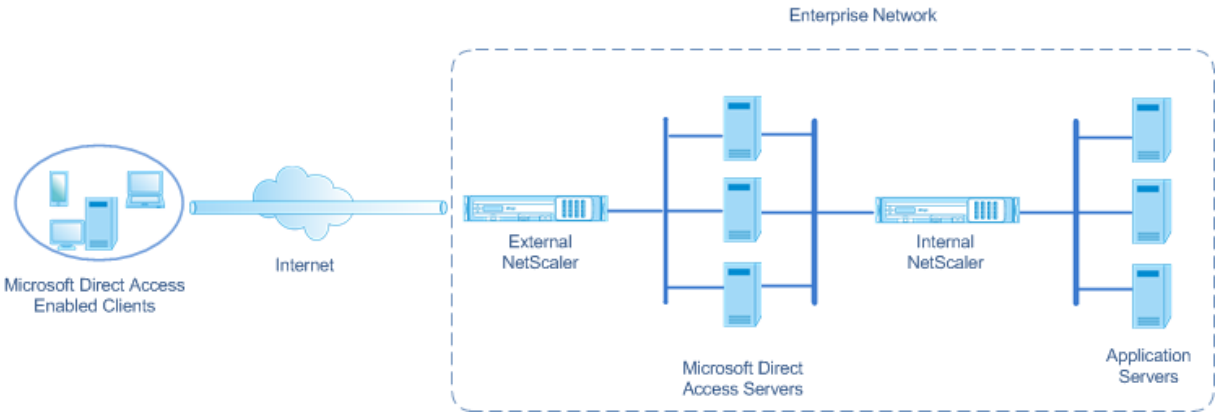
In a Direct Access deployment, Citrix ADC appliances provide high availability, scalability, high performance, and security. Citrix ADC load balancing functionality sends client traffic through the most appropriate server. The appliances can also forward Manage-Out traffic through the right path to reach the client.

Architecture

The architecture of a Microsoft Direct Access deployment consists of Direct Access enabled clients, Direct Access servers, application servers, and internal and external Citrix ADC appliances. Clients connect to an application server through a Direct Access server. An external Citrix ADC appliance load balance the client traffic to a Direct Access server, and an Internal Citrix ADC appliance forwards the client traffic from the Direct Access server to the destination application server. Direct Access is used for tunneling the client's IPv6 traffic over the IPv4 network. An IPv4 load balancing virtual server on the external Citrix ADC appliance load balances the client's tunneled traffic to one of the Direct Access servers. The Direct Access server extracts the IPv6 packets from the received client's IPv4 packets and sends them to the destination application server through the internal Citrix ADC appliance. The Internal Citrix ADC appliance has forwarding session rules with the source route cache option enabled for storing Layer 2 and Layer 3 connection information about the client's traffic from the Direct Access Server. The Citrix ADC appliance stores the following Layer 2 and Layer 3 information in a table called the source route cache table:

- Source IP address of the received packet
- MAC address of the Direct Access server that sent the packet
- VLAN ID of the Citrix ADC appliance that received the packet
- Interface ID of the Citrix ADC appliance that received the packet

The Citrix ADC appliance uses the information in the source route cache table for forwarding a response to the same Direct Access server because it has the tunneling information to reach the client. Also, the Internal appliance uses the source route cache table to forward application server's Manage-out traffic to the appropriate Direct Access server to reach a particular client.



Configuring the Internal Citrix ADC Appliance in a Microsoft Direct Access Deployment

To configure the Internal Citrix ADC appliance for forwarding an application server's response and manage-out traffic to the appropriate Direct Access Gateway, configure forwarding session rules. In each rule, set the sourceroutecache parameter to ENABLED.

To create a forwarding session rule by using the CLI:

At the command prompt, type:

- **add forwardingSession** <name> ((<network> [<netmask>]) | **-acl6name** <string> | **-aclname** <string>) **-sourceroutecache** (**ENABLED** | **DISABLED**]
- **show forwardingsession** <name>

Sample configuration:

In the following example, forwarding-session rule MS-DA-FW-1 is created on the internal Citrix ADC appliance. The forwarding session stores Layer 2 and Layer 3 information for any incoming IPv6 packets from a Direct Access server that matches source IPv6 prefix 2001:DB8::/96.

```
1 > add forwardingSession MS-DA-FW-1 2001:DB8::/96 -sourceroutecache -
    ENABLED
2  Done
```

Displaying the Source Route Cache Table

You can display the source route cache table for monitoring or detecting any unwanted connections between direct access servers and application servers.

To display the source route cache table by using the CLI:

At the command prompt, type:

- **show sourceroutecachetable**

Example:

```
> show sourceroutecachetable
SOURCEIP            MAC                 VLAN    INTERFACE
2001:DB8:5001:10    56:53:24:3d:02:eb    30        1/2
2001:DB8:5003:30    60:54:35:3e:04:bd    60        1/3
Done
```

Clearing the Source Route Cache Table

You can clear all the entries from the source route cache table on a Citrix ADC appliance.

To clear the source route cache table by using the CLI:

At the command prompt, type:

- **flush ns sourceroutecachetable**

Access Control Lists

September 14, 2021

Access Control Lists (ACLs) filter IP traffic and secure your network from unauthorized access. An ACL is a set of conditions that the Citrix ADC evaluates to determine whether to allow access. For example, the Finance department probably does not want to allow its resources to be accessed by other departments, such as HR and Documentation, and those departments want to restrict access to their data.

When the Citrix ADC receives a data packet, it compares the information in the data packet with the conditions specified in the ACL and allows or denies access. The administrator of the organization can configure ACLs to function in the following processing modes:

- ALLOW—Process the packet.
- BRIDGE—Bridge the packet to the destination without processing it. The packet is directly sent by Layer 2 and Layer 3 forwarding.
- DENY—Drop the packet.

ACL rules are the first level of defense on the Citrix ADC.

Citrix ADC supports the following types of ACLs:

- **Simple ACLs** filter packets based on their source IP address and, optionally, their protocol, destination port, or traffic domain. Any packet that has the characteristics specified in the ACL is dropped.

- **Extended ACLs** filter data packets based on various parameters, such as source IP address, source port, action, and protocol. An extended ACL defines the conditions that a packet must satisfy for the Citrix ADC to process the packet, bridge the packet, or drop the packet.

Nomenclature

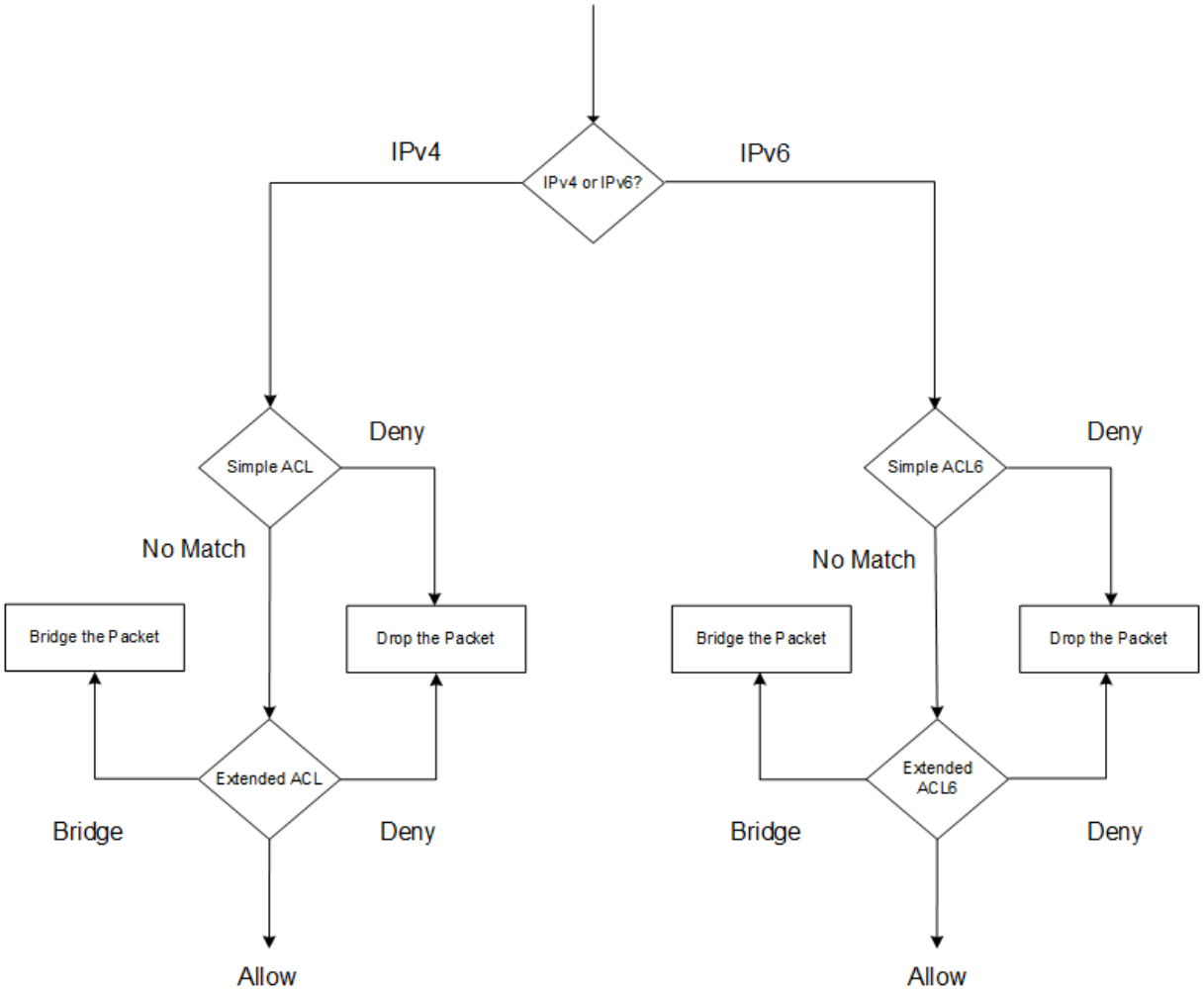
In the Citrix ADC user interfaces, the terms simple ACL and extended ACL refer to ACLs that process IPv4 packets. An ACL that processes IPv6 packets is called a simple ACL6 and or extended ACL6. When discussing both types, this documentation sometimes refers to both of them as simple ACLs or extended ACLs.

ACL Precedence

If both simple and extended ACLs are configured, incoming packets are compared to the simple ACLs first.

The Citrix ADC first determines whether the incoming packet is an IPv4 or an IPv6 packet, and then compares the packet's characteristics to either simple ACLs or simple ACL6s. If a match is found, the packet is dropped. If no match is found, the packet is compared to extended ACLs or extended ACL6s. If that comparison results in a match, the packet is handled as specified in the ACL. The packet can be bridged, dropped, or allowed. If no match is found, the packet is allowed.

Figure 1. Simple and Extended ACLs Flow Sequence



Simple ACLs and Simple ACL6s

September 14, 2021

A simple ACL or simple ACL6 uses few parameters and can be configured only to drop IP packets. Packets can be dropped based on their source IP address and, optionally, their protocol, destination port, or traffic domain.

When creating a simple ACL or simple ACL6, you can specify a time to live (TTL), in seconds, after which the ACL expires. ACLs with TTLs are not saved when you save the configuration. You can display simple ACLs and simple ACL6s to verify their configuration, and you can display their statistics.

Configuring Simple ACLs and Simple ACL6s

Configuring a simple ACL or simple ACL6 on a Citrix ADC can include the following tasks.

- **Create simple ACLs or simple ACL6s**. Creating simple ACLs or simple ACL6s to drop (deny) packets based on their source IP address and, optionally, their protocol, destination port, or traffic domain.
- **Remove simple ACLs or simple ACL6s**. These ACLs cannot be modified once created. If you must modify a simple ACL or simple ACL6, you must remove it and create a one.

CLI procedures

To create a simple ACL by using the CLI:

At the command prompt, type:

```
1 - ns simpleacl <aclname> DENY -srcIP <ip_addr> [-destPort <port> -
    protocol ( TCP | UDP )] [-TTL \<positive_integer>]
2 - show ns simpleacl [\<aclname>]
3 <!--NeedCopy-->
```

Example:

```
1 > add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
2 Done
3 <!--NeedCopy-->
```

To create a simple ACL6 by using the CLI:

At the command prompt, type:

```
1 - add ns simpleacl6 <aclname> DENY - srcIPv6 <ipv6_addr|null> [-
    destPort <port> -protocol ( TCP | UDP )] [-TTL <positive_integer>]
2 - show ns simpleacl6 [<aclname>]
3 <!--NeedCopy-->
```

Example:

```
1 > add ns simpleacl6 rule1 DENY - srcIPv6 3ffe:192:168:215::82 -
    destPort 80 -Protocol TCP -TTL 9000
2  Done
3 <!--NeedCopy-->
```

To remove a single simple ACL by using the CLI:

At the command prompt, type:

- **rm ns simpleacl** <aclname>
- **show ns simpleacl**

To remove a single simple ACL6 by using the CLI:

At the command prompt, type:

- **rm ns simpleacl6**<aclname>
- **show ns simpleacl6**

To remove all simple ACLs by using the CLI:

At the command prompt, type:

- **clear ns simpleacl**
- **show ns simpleacl**

To remove all simple ACL6s by using the CLI:

At the command prompt, type:

- **clear ns simpleacl6**
- **show ns simpleacl6**

GUI procedures

To create a simple ACL by using the GUI:

Navigate to **System > Network > ACLs** and, on the **Simple ACLs** tab, add a new simple ACL.

To create a simple ACL6 by using the GUI:

Navigate to **System > Network > ACLs** and, on the **Simple ACL6s** tab, add a new simple ACL6.

To remove a single simple ACL by using the GUI:

Navigate to **System > Network > ACLs** and, on the **Simple ACLs** tab, delete the simple ACL.

To remove a single simple ACL6 by using the GUI:

Navigate to **System > Network > ACLs** and, on the **Simple ACL6s** tab, delete the simple ACL6.

To remove all simple ACLs by using the GUI:

1. Navigate to **System > Network > ACLs**.
2. On the **Simple ACLs** tab, in the **Action** list, click **Clear**.

To remove all simple ACL6s by using the GUI:

1. Navigate to **System > Network > ACLs**.
2. On the **Simple ACL6s** tab, in the **Action** list, click **Clear**.

Displaying Simple ACL and Simple ACL6 Statistics

You can display the simple ACL (or simple ACL6) statistics, which include the number of matches, the number of misses, and the number of simple ACLs configured.

The following table describes the statistics you can display for simple ACLs and simple ACL6s.

Statistics	Indicates
ACL match	Packets matching an ACL
ACL misses	Packets not matching any ACL
ACL count	Number of ACLs configured

CLI procedures

To display simple ACL statistics by using the CLI:

At the command prompt, type:

- **stat ns simpleacl**

Example:

```
> stat ns simpleacl

SimpleACL Statistics

                                          Rate (/s)
                                             Total
SimpleACL hits                                  0
                        0
SimpleACL misses                                0
    51872
SimpleACLs count                               --
                        2
Done
<!--NeedCopy-->
```

To display simple ACL6 statistics by using the CLI:

At the command prompt, type:

- **stat ns simpleacl6**

GUI procedures

To display simple ACL statistics by using the GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Simple ACLs** tab, select the ACL and click **Statistics**.

To display simple ACL6 statistics by using the GUI:

Navigate to **System > Network > ACLs** and, on the **Simple ACL6s** tab, select the simple ACL6 and click **Statistics**.

Terminating Established Connections

For a simple ACL or simple ACL6, the Citrix ADC blocks any new connections that match the conditions specified in the ACL. Packets related to existing connections that were established before the ACL was created are not blocked. To terminate previously established connections that match an existing ACL, you can run a flush operation from the CLI or the GUI.

Flush can be useful in the following cases:

- You receive a list of blacklisted IP addresses and want to completely block those IP addresses from accessing the Citrix ADC. In this case, you create simple ACLs or simple ACL6s to block any new connections from these IP addresses, and then flush any existing connections associated with those addresses.
- You want to terminate many connections from a particular network without taking the time to terminate them one by one.

Before you begin

- When you run flush, the Citrix ADC searches through all of its established connections and terminates the connections that match the conditions specified in any of the simple ACLs configured on the ADC.
- If you plan to create more than one simple ACL and flush existing connections that match any of them, you can minimize the effect on performance by first creating all simple ACLs and then running flush only once.

CLI procedures

To terminate all established IPv4 connections that match any of your configured simple ACLs by using the CLI:

At the command prompt, type:

- **flush simpleacl -estSessions**

To terminate all established IPv6 connections that match any of your configured simple ACL6s by using the CLI:

At the command prompt, type:

- **flush simpleacl6 -estSessions**

GUI procedures

To terminate all established IPv4 connections that match any of your configured simple ACLs by using the GUI:

1. Navigate to **System > Network > ACLs**.
2. On the **Simple ACLs** tab, in the **Action** list, click **Flush**.

To terminate all established IPv6 connections that match any of your configured simple ACL6s by using the GUI:

1. Navigate to **System > Network > ACLs**.
2. On the **Simple ACL6s** tab, in the **Action** list, click **Flush**.

Extended ACLs and Extended ACL6s

September 14, 2021

Extended ACLs and extended ACL6s provide parameters and actions not available with simple ACLs. You can filter data based on parameters such as source IP address, source port, action, and protocol. You can specify tasks to allow a packet, deny a packet, or bridge a packet.

Extended ACLs and ACL6s can be modified after they are created, and you can renumber their priorities to specify the order in which they are evaluated.

Note: If you configure both simple and extended ACLs, simple ACLs take precedence over extended ACLs.

The following actions can be performed on extended ACLs and ACL6s: Modify, Apply, Disable, Enable, Remove, and Renumber (the priority). You can display extended ACLs and ACL6s to verify their configuration, and you can display their statistics.

You can configure the Citrix ADC to log details for packets that match an extended ACL.

Applying Extended ACLs and Extended ACL6s: Unlike simple ACLs and ACL6s, extended ACLs and ACL6s created on the Citrix ADC do not work until they are applied. Also, if you make any changes to an extended ACL or ACL6, such as disabling the ACLs, changing a priority, or deleting the ACLs, you must reapply the extended ACLs or ACL6s. You must reapply them after enabling logging. The procedure to

apply extended ACLs or ACL6s reapplies all of them. For example, if you have applied extended ACL rules 1 through 10, and you then create and apply rule 11, the first 10 rules are applied afresh.

If a session has a DENY ACL related to it, that session is terminated when you apply the ACLs.

Extended ACLs and ACL6s are enabled by default. When they are applied, the Citrix ADC starts comparing incoming packets against them. However, if you disable them, they are not used until you reenable them, even if they are reapplied.

Renumbering the priorities of Extended ACLs and Extended ACL6s: Priority numbers determine the order in which extended ACLs or ACL6s are matched against a packet. An ACL with a lower priority number has a higher priority. It is evaluated before ACLs with higher priority numbers (lower priorities), and the first ACL to match the packet determines the action applied to the packet.

When you create an extended ACL or ACL6, the Citrix ADC automatically assigns it a priority number that is a multiple of 10, unless you specify otherwise. For example, if two extended ACLs have priorities of 20 and 30, respectively, and you want a third ACL to have a value between those numbers, you might assign it a value of 25. If you later want to retain the order in which the ACLs are evaluated but restore their numbering to multiples of 10, you can use the renumber procedure.

Configuring Extended ACLs and Extended ACL6s

Configuring an extended ACL or ACL6 on a Citrix ADC consists of the following tasks.

- **Create an extended ACL or ACL6.** Create an extended ACL or ACL6 to either allow, deny, or bridge a packet. You can specify an IP address or range of IP addresses to match against the source or destination IP addresses of the packets. You can specify a protocol to match against the protocol of incoming packets.
- (Optional) **Modify an extended ACL or ACL6**. You can modify extended ACLs or ACL6s that you previously created. Or, if you want to temporarily take one out of use you can disable it, and later reenable it.
- **Apply extended ACLs or ACL6s**. After you create, modify, disable or reenable, or delete an extended ACL or ACL6, you must apply the extended ACLs or ACL6s to activate them.
- (Optional) **Renumber the priorities of extended ACLs or ACL6s**. If you have configured ACLs with priorities that are not multiples of 10 and want to restore the numbering to multiples of 10, use the renumber procedure.

CLI procedures

To create an extended ACL by using the CLI:

At the command prompt, type:

- **add ns acl** <aclname> <aclaction> [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>]

[-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-**icmpCode** <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)] [-**logstate** (ENABLED | DISABLED) [-**ratelimit** <positive_integer>]]

- **show ns acl** [<aclName>]

To create an extended ACL6 by using the CLI:

At the command prompt, type:

- **add ns acl6** <acl6name> <acl6action> [-**srcIPv6** [<operator>] <srcIPv6Val>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIPv6** [<operator>] <destIPv6Val>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-**icmpCode** <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)]
- **show ns acl6** [<aclName>]

To modify an extended ACL by using the CLI:

To modify an extended ACL, type the **set ns acl** command, the name of the extended ACL, and the parameters to be changed, with their new values.

To modify an extended ACL6 by using the CLI:

To modify an extended ACL6, type the **set ns acl6** command, the name of the extended ACL6, and the parameters to be changed, with their new values.

To disable or enable an extended ACL by using the CLI:

At the command prompt, type one of the following commands:

- **disable ns acl** <aclname>
- **enable ns acl** <aclname>

To disable or enable an extended ACL6 by using the CLI:

At the command prompt, type one of the following commands:

- **disable ns acl6** <aclname>
- **enable ns acl6** <aclname>

To apply extended ACLs by using the CLI:

At the command prompt, type:

- **apply ns acls**

To apply extended ACL6s by using the CLI:

At the command prompt, type:

- **apply ns acls6**

To renumber the priorities of extended ACLs by using the CLI:

At the command prompt, type:

- **renumber ns acls**

To renumber the priorities of extended ACL6s by using the CLI:

At the command prompt, type:

- **renumber ns acls6**

GUI procedures

To configure an extended ACL by using the GUI:

- Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, add a new extended ACL or edit an existing extended ACL. To enable or disable an existing extended ACL, select it, and then select **Enable** or **Disable** from the **Action** list.

To configure an extended ACL6s by using the GUI:

- Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, add a new extended ACL6 or edit an existing extended ACL6. To enable or disable an existing extended ACL6, select it, and then select **Enable** or **Disable** from the **Action** list.

To apply extended ACLs by using the GUI:

- Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, in the **Action** list, click **Apply**.

To apply extended ACL6s by using the GUI:

- Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, in the **Action** list, click **Apply**.

To renumber the priorities of extended ACLs by using the GUI:

- Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, in the **Action** list, click **Renumber Priority (s)**.

To renumber the priorities of extended ACL6s by using the GUI:

- Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, in the **Action** list, click **Renumber Priority (s)**.

Sample Configurations

The following table shows examples of configuring extended ACL rules through the command line interface: ACLs sample configurations.

Logging extended ACLs

You can configure the Citrix ADC to log details for packets that match extended ACLs.

In addition to the ACL name, the logged details include packet-specific information such as the source and destination IP addresses. The information is stored either in the syslog file or in the `nslog` file, depending on the type of global logging (`syslog or nslog`) enabled.

Logging must be enabled at both the global level and the ACL level. The global setting takes precedence.

To optimize logging, when multiple packets from the same flow match an ACL, only the first packet's details are logged, and the counter is incremented for every packet that belongs to the same flow. A flow is defined as a set of packets that have the same values for the source IP address, destination IP address, source port, destination port, and protocol parameters. To avoid flooding of log messages, the Citrix ADC performs internal rate limiting so that packets belonging to the same flow are not repeatedly logged. The total number of different flows that can be logged at any given time is limited to 10,000.

Note: You must apply ACLs after you enable logging.

CLI procedures

To configure extended ACL Logging by using the CLI:

At the command prompt, type the following commands to configure logging and verify the configuration:

- **set ns acl** <aclName> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
- **apply acls**
- **show ns acl** [<aclName>]

GUI procedures

To configure extended ACL Logging by using the GUI:

1. Navigate to **System > Network > ACLs** and, on the **Extended ACLs** tab, open the extended ACL.
2. Set the following parameters:
   - **Log State**—Enable or disable logging of events related to the extended ACL rule. The log messages are stored in the configured `syslog or auditlog` server.
   - **Log Rate Limit**—Maximum number of log messages to be generated per second. If you set this parameter, you must enable the Log State parameter.

Sample configuration


```
> set ns acl restrict -logstate ENABLED -ratelimit 120
Warning: ACL modified, apply ACLs to activate change

> apply ns acls
Done
<!--NeedCopy-->
```

Logging extended ACL6s

You can configure the Citrix ADC appliance to log details for packets that match an extended ACL6 rule. In addition to the ACL6 name, the logged details include packet-specific information, such as the source and destination IP addresses. The information is stored either in a syslog or `nslog` file, depending on the type of logging (`syslog or nslog`) that you have configured in the Citrix ADC appliance.

To optimize logging, when multiple packets from the same flow match an ACL6, only the first packet's details are logged. The counter is incremented for every other packet that belongs to the same flow. A flow is defined as a set of packets that have the same values for the following parameters:

- Source IP
- Destination IP
- Source port
- Destination port
- Protocol (TCP or UDP)

If an incoming packet is not from the same flow, a new flow is created. The total number of different flows that can be logged at any given time is limited to 10,000.

CLI procedures

To configure logging for an extended ACl6 rule by using the CLI:

- To configure logging while adding the extended ACL6 rule, at the command prompt, type:
  - **add acl6** <acl6Name> <acl6action> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
  - **apply acls6**
  - **show acl6** [<acl6Name>]
- To configure logging for an existing extended ACL6 rule, at the command prompt, type:
  - **set acl6** <acl6Name> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
  - **show acl6** [<acl6Name>]
  - **apply acls6**

GUI procedures

To configure extended ACL6 Logging by using the GUI:

1. Navigate to **System** > **Network** > **ACLs** and, then click the **Extended ACL6s** tab.
2. Set the following parameters while adding, or modifying an existing extended ACL6 rule.
   - **Log State** — Enable or disable logging of events related to the extended ACL6s rule. The log messages are stored in the configured syslog or `auditlog` server.
   - **Log Rate Limit**—Maximum number of log messages to be generated per second. If you set this parameter, you must enable the **Log State** parameter.

Sample configuration

```
> set acl6 ACL6-1 -logstate ENABLED -ratelimit 120
Done

> apply acls6
Done
<!--NeedCopy-->
```

Displaying extended ACLs and extended ACL6s statistics

You can display statistics of extended ACLs and ACL6s.

The following table lists the statistics associated with extended ACLs and ACL6s, and their descriptions.

Statistic	Specifies
Allow ACL matches	Packets matching ACLs with processing mode set to ALLOW. The Citrix ADC processes these packets.
NAT ACL matches	Packets matching a NAT ACL, resulting in a NAT session.
Deny ACL matches	Packets dropped because they match ACLs with processing mode set to DENY.
Bridge ACL matches	Packets matching a bridge ACL, which in transparent mode bypasses service processing.
ACL matches	Packets matching an ACL.
ACL misses	Packets not matching any ACL.
ACL Count	Total number of ACL rules configured by users.

Statistic	Specifies
Effective ACL Count	Total number of effective ACL configured internally. For an extended ACL with a range of IP addresses, the Citrix ADC appliance internally creates an extended ACL for each IP address. For example, for an extended ACL with 1000 IPv4 addresses (range or dataset), the Citrix ADC internally created 1000 extended ACLs.

CLI procedures

To display the statistics of all extended ACLs by using the CLI:

At the command prompt, type:

- **stat ns acl**

To display the statistics of all extended ACL6s by using the CLI:

At the command prompt, type:

- **stat ns acl6**

GUI procedures

To display the statistics of an extended ACL by using the GUI:

- Navigate to **System > Network > ACLs**, on the **Extended ACLs** tab, select the extended ACL, and click **Statistics**.

To display the statistics of an extended ACL6 by using the GUI:

- Navigate to **System > Network > ACLs**, on the **Extended ACL6s** tab, select the extended ACL, and click **Statistics**.

Stateful ACLs

A stateful ACL rule creates a session when a request matches the rule and allows the resulting responses even if these responses match a deny ACL rule in the Citrix ADC appliance. A stateful ACL offloads the work of creating more ACL rules/forwarding session rules for allowing these specific responses.

Stateful ACLs can be best used in an edge firewall deployment of a Citrix ADC appliance having the following requirements:

- The Citrix ADC appliance must allow requests initiated from internal clients and the related responses from the Internet.
- The appliance must drop the packets from the Internet that are not related to any client connections.

Before you begin

Before you configure stateful ACL rules, note the following points:

- The Citrix ADC appliance supports stateful ACL rules and stateful ACL6 rules.
- In a high availability setup, the sessions for a stateful ACL rule are not synchronized to the secondary node.
- You cannot configure an ACL rule as stateful if the rule is bound to any Citrix ADC NAT configuration. Some examples of Citrix ADC NAT configurations are:
  - RNAT
  - Large Scale NAT (large scale NAT44, DS-Lite, large scale NAT64)
  - NAT64
  - Forwarding session
- You cannot configure an ACL rule as stateful if TTL and Established parameters are set for this ACL rule.
- The sessions created for a stateful ACL rule continue to exist until time out irrespective of the following ACL operations:
  - Remove ACL
  - Disable ACL
  - Clear ACL
- Stateful ACLs are not supported for the following protocols:
  - Active FTP
  - TFTP

Configure stateful IPv4 ACL rules

Configuring a stateful ACL rule consists of enabling the stateful parameter of an ACL rule.

To enable the stateful parameter of an ACL rule by using the CLI:

- To enable the stateful parameter while adding an ACL rule, at the command prompt, type:
  - **add acl** <lname> ALLOW -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>
- To enable the stateful parameter of an existing ACL rule, at the command prompt, type:
  - **set acl** <name> -**stateful** (ENABLED | DISABLED)

- **apply acls**
- **show acl** <name>

To enable the stateful parameter of an ACL rule by using the GUI:

1. Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab.
2. Enable the **Stateful** parameter while adding, or modifying an existing ACL rule.

Sample configuration

```
> add acl ACL-1 allow -srciP 1.1.1.1 -stateful Yes

Done

> apply acls

Done

> show acl

1)         Name: ACL-1

    Action: ALLOW                          Hits: 0

    srcIP = 1.1.1.1

    destIP

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO

    TTL:

    Log Status: DISABLED

    Forward Session: NO

    Stateful: YES
```



```
<!--NeedCopy-->
```

Configure stateful ACL6 rules

Configuring a stateful ACL6 rule consists of enabling the stateful parameter of an ACL6 rule.

To enable the stateful parameter of an ACL6 rule by using the CLI:

- To enable the stateful parameter while adding an ACL6 rule, at the command prompt, type:
  - **add acl6** <name> ALLOW -stateful (ENABLED | DISABLD)
  - **apply acls6**
  - **show acl6** <name>
- To enable the stateful parameter of an existing ACL6 rule, at the command prompt, type:
  - **set acl6** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls6**
  - **show acl6** <name>

To enable the stateful parameter of an ACL6 rule by using the GUI:

1. Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab.
2. Enable the **Stateful** parameter while adding, or modifying an existing ACL6 rule.

Sample configuration

```
>  add acl6 ACL6-1 allow -srcipv6 1000::1  - stateful Yes

Done

>  apply acls6

Done

> show acl6

1)    Name: ACL6-1

    Action: ALLOW                          Hits: 0

    srcIPv6 = 1000::1

    destIPv6

    srcMac:
```



```
20
21     Protocol:
22
23     Vlan:                                  Interface:
24
25     Active Status: ENABLED                  Applied Status: NOTAPPLIED
26
27     Priority: 10                            NAT: NO
28
29     TTL:
30
31     Forward Session: NO
32
33     Stateful: YES
34 <!--NeedCopy-->
```

Dataset based extended ACLs

Many ACLs are required in an enterprise. Configuring and managing many ACLs is difficult and cumbersome when they require frequent changes.

A Citrix ADC appliance supports datasets in extended ACLs. Dataset is an existing feature of a Citrix ADC appliance. A dataset is an array of indexed patterns of types: number (integer), IPv4 address, or IPv6 address.

Dataset support in extended ACLs is useful for creating multiple ACL rules, which require common ACL parameters.

While creating an ACL rule, instead of specifying the common parameters, you can specify a dataset, which includes these common parameters.

Any changes made in the dataset are automatically reflected in the ACL rules that are using this dataset. ACLs with datasets are easier to configure and manage. They are also smaller and easier to read than the conventional ACLs.

Currently, the Citrix ADC appliance supports only the IPv4 address type dataset for extended ACLs.

Before you begin

Before configuring dataset based extended ACL rules, note the following points:

- Make sure that you are familiar with the dataset feature of a Citrix ADC appliance. For more information about datasets, see Pattern sets and data sets.
- The Citrix ADC appliance supports datasets only for IPv4 extended ACLs.

- The Citrix ADC appliance supports only the IPv4 type datasets for extended ACLs.
- The Citrix ADC appliance supports Dataset based extended ACLs for all set ups: standalone, high availability, and cluster.
- For an extended ACL with a range of IP addresses, the Citrix ADC appliance internally creates an extended ACL for each IP address. For example, for an IPv4 dataset based extended ACL with 1000 IPv4 addresses bound to the dataset, the Citrix ADC appliance internally created 1000 extended ACLs.
  - The Citrix ADC appliance supports a maximum of 10K extended ACLs. For an IPv4 dataset based extended ACL with a range of IP addresses bound to the dataset, the Citrix ADC appliance stops creating internal ACLs once the total number of extended ACLs reaches the maximum limit.
  - The following counters are present as part of the extended ACL statistics:
    * **ACL count**. Total number of ACL rules configured by users.
    * **Effective ACL count**. Total number of effective ACL rules that the Citrix ADC appliance configures internally.

    For more information, see Displaying extended ACL and extended ACL6s Statistics.
- The Citrix ADC appliance does not support `set` and `unset` operations for associating/dissociating datasets with the parameters of an extended ACL. You can set the ACL parameters to a dataset only during the `add` operation.

Configure dataset based extended ACLs

Configuring a dataset based extended ACL rule consists of the following tasks:

- **Add a dataset**. A dataset is an array of indexed patterns of types: number (integer), IPv4 address, or IPv6 address. In this task, you create a type of dataset, for example, a dataset of type IPv4.
- **Bind values to the dataset**. Specify a value or a range of values to the dataset. The specified values must be of the same type as the dataset type. For example, you can specify an IPv4 address or a range of IPv4 addresses to the dataset of type IPv4.
- **Add an extended ACL and set ACL paramters to the dataset**. Add an extended ACL and set the required ACL parameters to the dataset. This setting results in the parameters set to the values specified in the dataset.
- **Apply extended ACLs**. Apply the ACLs to activate any new or modified extended ACLs.

To add a policy dataset by using the CLI:

At the command prompt, type:

- **add policy dataset** <name> <type>
- **show policy dataset**

To bind a pattern to the data set by using the CLI:

At the command prompt, type:

- **bind policy dataset** <name> <value> [-endRange <string>]
- **show policy dataset**

To add an extended ACL and set the ACL paramters to the dataset by using the CLI:

At the command prompt, type:

- **add ns acl** <aclname> <aclaction> [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] …
- **show acls**

To apply extended ACLs by using the CLI:

At the command prompt, type:

- **apply acls**

Sample configuration

In the following sample configuration of a dataset based extended ACL, an IPv4 dataset DATASET-ACL-1 is created. Two IPv4 addresses: 192.0.2.30 and 192.0.2.60, and two IPv4 address ranges: (198.51.100.15 - 45) and (203.0.113.60-90) are bound to DATASET-ACL-1. DATASET-ACL-1 is then specified to the srcIP and destIP parameters of the extended ACL ACL-1.

```
add policy dataset DATASET-ACL-1 IPV4

bind dataset DATASET-ACL-1 192.0.2.30

bind dataset DATASET-ACL-1 192.0.2.60

bind dataset DATASET-ACL-1 198.51.100.15 -endrange 198.51.100.45

bind dataset DATASET-ACL-1 203.0.113.60 -endrange 203.0.113.90

add ns acl ACL-1 ALLOW -srcIP DATASET-ACL-1 -destIP DATASET-ACL-1

apply acls
<!--NeedCopy-->
```


MAC Address Wildcard Mask for ACLs

September 14, 2021

A wildcard mask parameter has been introduced for extended ACLs and ACL6s and is used with the source MAC address parameter to define a range of MAC addresses to be match against the source MAC address of incoming packets.

Wild card masks specify which hexadecimal digits of the MAC address are used and which hexadecimal digits are ignored. The wildcard mask parameter specifies a series of ones and zeroes and has a length of 12 digits. Each digit is a mask for the corresponding hexadecimal digit of the MAC address. A zero digit in the wildcard mask indicates that the corresponding hexadecimal digit of the MAC address must be considered and a one digit indicates that the corresponding hexadecimal digit to be ignored.

The wildcard mask must meet the following conditions:

- Has only one series of zeroes
- Has only one series of ones
- Start with a series of zeroes

The following are some of the examples of valid wildcard masks:

- 000000111111
- 000000011111
- 000011111111

The following are some of the examples of invalid wildcard masks:

- 000000111100
- 111110000000
- 010101010101

For an ACL, a wildcard mask of 000000111111 for MAC address 96:fa:95:1d:67:4a defines the MAC address range 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF. This MAC address range is matched against the source MAC address of the incoming packets.

To specify a range of source MAC addresses in an ACL rule by using the CLI:

At the command prompt, type:

```
1 -  add ns acl <name> <action> -srcMac <mac_addr> -srcMacMask <string>
2 -  show ns acl <aclname>
3 <!--NeedCopy-->
```

Example:

```
1 add ns acl ACL-1 ALLOW  - protocol TCP  - srcport 2000-3000 -srcMac 96:fa
     :95:1d:67:4a
```



```
  - srcMacMask 000000111111
Done
<!--NeedCopy-->
```

To specify a range of source MAC addresses in an ACL6 rule by using the CLI:

At the command prompt, type:

```
-  add ns acl6 <name> <action> -srcMac <mac_addr> -srcMacMask <string>
-  show ns acl6 <acl6name>
<!--NeedCopy-->
```

Example:

```
> add ns acl6 ACL6-1 ALLOW -destIPv6 2001::45 -srcMac 96:fa:90:1d:67:4a
 - srcMacMask 000000001111
Done
<!--NeedCopy-->
```

Blocking Traffic on Internal Ports

September 14, 2021

By default, a Citrix ADC appliance does not block some type of internal traffic even using ACL rules.

The following table lists the internal traffic types that a Citrix ADC appliance does not block even using ACL rules:

Citrix ADC Setup	Protocol	Destination Port	Destination IP address
All	TCP	3008–3011	NSIP or SNIP
All	TCP	179	NSIP or SNIP
All	UDP	520	NSIP or SNIP
High availability	UDP	3003	NSIP
High availability	TCP	4001	NSIP
High availability	TCP	22	NSIP
Cluster	UDP	7000	NSIP

This feature of not blocking the earlier mentioned types of traffic is specified by the default setting of

the global Layer-3 `Implicit ACL Allow` (`implicitACLAllow`) parameter.

You can disable this parameter if you want to block the earlier mentioned traffic types using the ACL rules. An appliance in a high availability setup makes an exception for its partner (primary or secondary) node. It does not block traffic from that node.

To disable or enable this parameter by using the CLI:

At the command prompt, type:

set l3param -implicitACLAllow [ENABLED DISABLED]

-
- **sh l3param**

Note: The parameter implicitACLAllow is enabled by default.

Example:

```
1 > set l3param -implicitACLAllow DISABLED
2 Done
3 <!--NeedCopy-->
```

IP Routing

September 14, 2021

Citrix ADC appliances support both dynamic and static routing. Because simple routing is not the primary role of a Citrix ADC, the main objective of running dynamic routing protocols is to enable route health injection (RHI), so that an upstream router can choose the best among multiple routes to a topographically distributed virtual server.

Most Citrix ADC implementations use some static routes to reduce routing overhead. You can create backup static routes and monitor routes to enable automatic switchover in the event that a static route goes down. You can also assign weights to facilitate load balancing among static routes, create null routes to prevent routing loops, and configure IPv6 static routes. You can configure policy based routes (PBRs), for which routing decisions are based on criteria that you specify.

Configuring Dynamic Routes

September 14, 2021

When a dynamic routing protocol is enabled, the corresponding routing process monitors route updates and advertises routes. Routing protocols enable an upstream router to use the equal cost multipath (ECMP) technique to load balance traffic to identical virtual servers hosted on two standalone Citrix ADC appliances. Dynamic routing on a Citrix ADC appliance uses three routing tables. In a high-availability setup, the routing tables on the secondary appliance mirror those on the primary.

For command reference guides and unsupported commands on dynamic routing protocol, see Dynamic Routing Protocol Command Reference Guides and Unsupported Commands.

The Citrix ADC supports the following protocols:

- Routing Information Protocol (RIP) version 2
- Open Shortest Path First (OSPF) version 2
- Border Gateway Protocol (BGP)
- Routing Information Protocol next generation (RIPng) for IPv6
- Open Shortest Path First (OSPF) version 3 for IPv6
- ISIS Protocol

You can enable more than one protocol simultaneously.

Routing Tables in Citrix ADC

In a Citrix ADC appliance, the Citrix ADC kernel routing table, the FreeBSD kernel routing table, and the NSM FIB routing table each hold a different set of routes and serve a different purpose. They communicate with each other by using UNIX routing sockets. Route updates are not automatically propagated from one routing table to another. You must configure propagation of route updates for each routing table.

NS Kernel Routing Table

The NS kernel routing table holds subnet routes corresponding to the NSIP and to each SNIP and MIP. Usually, no routes corresponding to VIPs are present in the NS kernel routing table. The exception is a VIP added by using the add ns ip command and configured with a subnet mask other than 255.255.255.255. If there are multiple IP addresses belonging to the same subnet, they are abstracted as a single subnet route. In addition, this table holds a route to the loopback network (127.0.0.0) and any static routes added through the CLI (CLI). The entries in this table are used by the Citrix ADC in packet forwarding. From the CLI, they can be inspected with the show route command.

FreeBSD Routing Table

The sole purpose of the FreeBSD routing table is to facilitate initiation and termination of management traffic (telnet, ssh, etc.). In a Citrix ADC appliance, these applications are tightly coupled to FreeBSD,

and it is imperative for FreeBSD to have the necessary information to handle traffic to and from these applications. This routing table contains a route to the NSIP subnet and a default route. In addition, FreeBSD adds routes of type WasCloned(W) when the Citrix ADC establishes connections to hosts on local networks. Because of the highly specialized utility of the entries in this routing table, all other route updates from the NS kernel and NSM FIB routing tables bypass the FreeBSD routing table. Do not modify it with the route command. The FreeBSD routing table can be inspected by using the netstat command from any UNIX shell.

Network Services Module (NSM) FIB

The NSM FIB routing table contains the advertisable routes that are distributed by the dynamic routing protocols to their peers in the network. It may contain:

- **Connected routes**. IP subnets that are directly reachable from the Citrix ADC. Typically, routes corresponding to the NSIP subnet and subnets over which routing protocols are enabled are present in NSM FIB as connected routes.
- **Kernel routes**. All the VIP addresses on which the -hostRoute option is enabled are present in NSM FIB as kernel routes if they satisfy the required RHI Levels. In addition, NSM FIB contains any static routes configured on the CLI that have the - advertise option enabled. Alternatively, if the Citrix ADC is operating in Static Route Advertisement (SRADV) mode, all static routes configured on the CLI are present in NSM FIB. These static routes are marked as kernel routes in NSM FIB, because they actually belong to the NS kernel routing table.
- **Static routes**. Normally, any static route configured in VTYSH is present in NSM FIB. If administrative distances of protocols are modified, this may not always be the case. An important point to note is that these routes can never get into the NS kernel routing table.
- **Learned routes**. If the Citrix ADC is configured to learn routes dynamically, the NSM FIB contains routes learned by the various dynamic routing protocols. Routes learned by OSPF, however, need special processing. They are downloaded to FIB only if the fib-install option is enabled for the OSPF process. This can be done from the router-config view in VTYSH.

Dynamic Routing in a High Availability Setup

In a high availability setup, the primary node runs the routing process and propagates routing table updates to the secondary node. The routing table of the secondary node mirrors the routing table on the primary node.

Non-Stop Forwarding

After failover, the secondary node takes some time to start the protocol, learn the routes, and update its routing table. But this does not affect routing, because the routing table on the secondary node

is identical to the routing table on the primary node. This mode of operation is known as non-stop forwarding.

Black Hole Avoidance Mechanism

After failover, the new primary node injects all its VIP routes into the upstream router. However, that router retains the old primary node's routes for 180 seconds. Because the router is not aware of the failover, it attempts to load balance traffic between the two nodes. During the 180 seconds before the old routes expire, the router sends half the traffic to the old, inactive primary node, which is, in effect, a black hole.

To prevent this, the new primary node, when injecting a route, assigns it a metric that is slightly lower than the one specified by the old primary node.

Interfaces for Configuring Dynamic Routing

To configure dynamic routing, you can use either the GUI or a command-line interface. The Citrix ADC supports two independent command-line interfaces: the CLI and the Virtual Teletype Shell (VTYSH). The CLI is the appliance's native shell. VTYSH is exposed by ZebOS. The Citrix ADC routing suite is based on ZebOS, the commercial version of GNU Zebra.

> **Note:**
>
> Citrix recommends that you use VTYSH for all commands except those that can be configured only on the CLI. Use of the CLI should generally be limited to commands for enabling the routing protocols, configuring host route advertisement, and adding static routes for packet forwarding.

Dynamic Routing Protocol Command Reference Guides and Unsupported Commands

The following table lists command reference guide links, for various dynamic routing protocols, and unsupported commands on the Citrix ADC appliance: Dynamic routing protocol reference guides and unsupported commands.

Configuring RIP

September 14, 2021

Routing Information Protocol (RIP) is a Distance Vector protocol. The Citrix ADC supports RIP as defined in RFC 1058 and RFC 2453. RIP can run on any subnet.

After enabling RIP, you need to configure advertisement of RIP routes. For troubleshooting, you can limit RIP propagation. You can display RIP settings to verify the configuration.

Enabling and Disabling RIP

Use either of the following procedures to enable or disable RIP. After you enable RIP, the Citrix ADC appliance starts the RIP process. After you disable RIP, the appliance stops the RIP process.

To enable or disable RIP routing by using the CLI:

At the command prompt, enter one of the following commands to enable or disable RIP:

- **enable ns feature RIP**
- **disable ns feature RIP**

To enable or disable RIP routing by using the GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features** group, click **Change advanced features**.
2. Select or clear the **RIP Routing** option.

Advertising Routes

RIP enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone Citrix ADC appliances. Route advertisement enables an upstream router to track network entities located behind the Citrix ADC.

To configure RIP to advertise routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```



```
<!--NeedCopy-->
```

Limiting RIP Propagations

If you need to troubleshoot your configuration, you can configure listen-only mode on any given interface.

To limit RIP propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the RIP Configuration

You can display the routing table and other RIP settings.

To view the RIP settings by using the VTYSH command line:

At the command prompt, type the following commands in the following order:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh rip	Display updated RIP routing table.

Command	Specifies
sh rip interface <vlan_name>	Displays RIP information for the specified VLAN.

Example:

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
<!--NeedCopy-->
```

Configuring OSPF

October 28, 2021

The Citrix ADC supports Open Shortest Path First (OSPF) Version 2 (RFC 2328). The features of OSPF on the Citrix ADC are:

- If a vserver is active, the host routes to the vserver can be injected into the routing protocols.
- OSPF can run on any subnet.
- Route learning advertised by neighboring OSPF routers can be disabled on the Citrix ADC.
- The Citrix ADC can advertise Type-1 or Type-2 external metrics for all routes.
- The Citrix ADC can advertise user-specified metric settings for VIP routes. For example, you can configure a metric per VIP without special route maps.
- You can specify the OSPF area ID for the Citrix ADC.
- The Citrix ADC supports not-so-stubby-areas (NSSAs). An NSSA is similar to an OSPF stub area but allows injection of external routes in a limited fashion into the stub area. To support NSSAs, a new option bit (the N bit) and a new type (Type 7) of Link State Advertisement (LSA) area have been defined. Type 7 LSAs support external route information within an NSSA. An NSSA area border router (ABR) translates a type 7 LSA into a type 5 LSA that is propagated into the OSPF domain. The OSPF specification defines only the following general classes of area configuration:
  - Type 5 LSA: Originated by routers internal to the area are flooded into the domain by AS boarder routers (ASBRs).
  - Stub: Allows no type 5 LSAs to be propagated into/throughout the area and instead depends on default routing to external destinations.

After enabling OSPF, you need to configure advertisement of OSPF routes. For troubleshooting, you can limit OSPF propagation. You can display OSPF settings to verify the configuration.

Enabling and Disabling OSPF

To enable or disable OSPF, you must use either the CLI or the GUI. When OSPF is enabled, the Citrix ADC starts the OSPF process. When OSPF is disabled, the Citrix ADC stops the OSPF routing process.

To enable or disable OSPF routing by using the CLI:

At the command prompt, type one of the following commands:

1. **enable ns feature OSPF**
2. **disable ns feature OSPF**

To enable or disable OSPF routing by using the GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features** group, click **Change advanced features**.
2. Select or clear the **OSPF Routing** option.

Advertising OSPF Routes

OSPF enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone Citrix ADC appliances. Route advertising enables an upstream router to track network entities located behind the Citrix ADC.

To configure OSPF to advertise routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enters global configuration mode.
router OSPF	Start OSPF routing process and enter configuration mode for the routing process.
network A.B.C.D/M area <0-4294967295>	Enable routing on an IP network.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
```



```
NS(config-router)# network 10.102.29.0/24 area 0
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

Limiting OSPF Propagations

If you need to troubleshoot your configuration, you can configure listen-only mode on any given VLAN.

To limit OSPF propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router OSPF	Start OSPF routing process and enters configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the OSPF Configuration

You can display current OSPF neighbors, and OSPF routes.

To view the OSPF settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.

Command	Specifies
sh OSPF neighbor	Displays current neighbors.
sh OSPF route	Displays OSPF routes.

Example:

```
>VTYSH
NS# sh ip OSPF neighbor
NS# sh ip OSPF route
<!--NeedCopy-->
```

Configuring Graceful Restart for OSPF

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neighbor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped.

Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

> **Note:**
>
> Graceful restart is not supported for high availability setups in INC mode.

To configure graceful restart for OSPF by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.

Command	Example	Command Description
router-id <id>	NS(config)# router-id 1.1.1.1	Sets a router identifier for the Citrix ADC appliance. This identifier is set for all the dynamic routing protocols. The same ID must be specified in the other node in a high availability set up for graceful restart to work properly in the HA setup.
ospf restart grace-period <1-1800>	NS(config)# ospf restart grace-period 170	Specifies the grace period, in seconds, for which the routes are to be preserved in the helper devices. Default value: 120 seconds.
ospf restart helper max-grace-period <1-1800>	NS(config)# ospf restart helper max-grace-period 180	This is an optional command to limit the maximum grace period for which the Citrix ADC appliance will be in the helper mode. If the Citrix ADC appliance receives an opaque LSA with grace-period greater than the set helper max-grace-period, the LSA is discarded and the Citrix ADC is not placed in helper mode.
router ospf	NS(config)# router ospf	Starts OSPF routing process and enter configuration mode for the routing process.
network A.B.C.D/M area <0-4294967295>	NS(config-router)# network 192.0.2.0/24 area 0	Enables routing on an IP network.
capability restart graceful	NS(config-router)# capability restart graceful	Enables graceful restart on the OSPF routing process.
redistribute kernel	NS(config-router)# redistribute kernel	Redistributes kernel routes.

Configuring BGP

October 28, 2021

The Citrix ADC appliance supports BGP (RFC 4271). The features of BGP on the Citrix ADC are:

- The Citrix ADC advertises routes to BGP peers.
- The Citrix ADC injects host routes to virtual IP addresses (VIPs), as determined by the health of the underlying virtual servers.
- The Citrix ADC generates configuration files for running BGP on the secondary node after failover in an HA configuration.
- This protocol supports IPv6 route exchanges.
- As-Override Support in Border Gateway Protocol

After enabling BGP, you need to configure advertisement of BGP routes. For troubleshooting, you can limit BGP propagation. You can display BGP settings to verify the configuration.

Prerequisites for IPv6 BGP

Before you begin configuring IPv6 BGP, do the following:

- Make sure that you understand the IPv6 BGP protocol.
- Enable the IPv6 feature.

Enabling and Disabling BGP

To enable or disable BGP, you must use either the CLI or the GUI. When BGP is enabled, the Citrix ADC appliance starts the BGP process. When BGP is disabled, the appliance stops the BGP process.

To enable or disable BGP routing by using the CLI:

At the command prompt, type one of the following commands:

- enable ns feature BGP
- disable ns feature BGP

To enable or disable BGP routing by using the GUI:

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the BGP Routing option.

Advertising IPv4 Routes

You can configure the Citrix ADC appliance to advertise host routes to VIPs and to advertise routes to downstream networks.

To configure BGP to advertise IPv4 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv4 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the link local IPv4 address of the neighbor in the specified autonomous system.
Address-family ipv4	Enter address family configuration mode.
Neighbor < IPv4 address> activate	Exchange prefixes for the IPv4 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor 10.102.29.170 remote-as 100
NS(config-router)# Address-family ipv4
NS(config-router-af)# Neighbor 10.102.29.170 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

Prerequisites for IPv6 BGP

Before you begin configuring IPv6 BGP, do the following:

- Make sure that you understand the IPv6 BGP protocol.
- Enable the IPv6 feature.

Advertising IPv6 BGP Routes

Border Gateway Protocol (BGP) enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone Citrix ADC appliances. Route advertising enables an upstream router to track network entities located behind the Citrix ADC.

To configure BGP to advertise IPv6 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv6 address> remote-as < as-number>	Update the IPv6 BGP neighbor table with the link local IPv6 address of the neighbor in the specified autonomous system.
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> activate	Exchange prefixes for the IPv6 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```


Verifying the BGP Configuration

You can use VTYSH to display BGP settings.

To view the BGP settings using the VTYSH command line

At the command prompt, type:

```
VTYSH
You are now in the VTYSH command prompt. An output similar to the
   following appears:
NS170#
At the VTYSH command prompt, type:
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>
<!--NeedCopy-->
```

As-Override Support in Border Gateway Protocol

As a part of BGP loop prevention functionality, if a router receives a BGP packet containing the router's Autonomous System Number (ASN) in the Autonomous Systems (AS) path, the router drops the packet. The assumption is that the packet originated from the router and has reached the place from where it originated.

If an enterprise has several sites with a same ASN, BGP loop prevention causes the sites with an identical ASN to not get linked by another ASN. Routing updates (BGP packets) are dropped when another site receives them.

To solve this issue, BGP AS-Override functionality has been added to the ZebOS BGP routing module of the Citrix ADC.

With AS-Override enabled for a peer device, when the Citrix ADC appliance receives a BGP packet for forwarding to the peer, and the ASN of the packet matches that of the peer, the appliance replaces the ASN of the BGP packet with its own ASN number before forwarding the packet.

You can enable AS-Override for a specific neighbor or a group of neighbors (peer group) by using the VTYSH command line.

To configure BGP AS-Override for a IPv4 neighbor by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor < IPv4 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <IPv4 address> as-override	Enable BGP as-override for the specified neighbor.

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# Neighbor 192.0.2.100 remote-as 100
4      NS(config-router)# Neighbor 10.102.29.100 as-override
5  <!--NeedCopy-->
```

To configure BGP AS-Override for a IPv4 BGP peer group by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor <peer group name> **peer-group**	Create a BGP peer group.
Neighbot <IPv4 address> **peer-group** <peer group name>	Associate neighbors to the specified peer group.
Neighbor <peer group name> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# neighbor external-peers-1 peer-group
```



```
4     NS(config-router)# neighbor 192.0.2.101 peer-group external-peers-1
5     NS(config-router)# neighbor 192.0.2.102 peer-group external-peers-1
6     NS(config-router)# neighbor 192.0.2.103 peer-group external-peers-1
7     NS(config-router)# Neighbor external-peers-1 remote-as 100
8     NS(config-router)# Neighbor external-peers-1 as-override
9  <!--NeedCopy-->
```

To configure BGP AS-Override for an IPv6 neighbor by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor < IPv6 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <IPv6 address> as-override	Enable BGP as-override for the specified neighbor.
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> activate	Exchange prefixes for the IPv6 router family between the specified neighbor and the Citrix ADC by using the link local address.
Neighbor <IPv6 address> as-override	Enable BGP as-override for the specified neighbor.

```
1     > VTYSH NS# configure terminal
2     NS(config)# router BGP 200
3     NS(config-router)# Neighbor a1bc::102 remote-as 100
4     NS(config-router)# Neighbor a1bc::102 as-override
5     NS(config-router)# Address-family ipv6
6     NS(config-router-af)# Neighbor a1bc::102 activate
7     NS(config-router)# Neighbor a1bc::102 as-override
8  <!--NeedCopy-->
```

To configure BGP AS-Override for IPv6 peer group by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor <peer group name> **peer-group**	Create a BGP peer group.
Neighbor <IPv6 address> **peer-group** <peer group name>	Associate a neighbor with the specified peer group.
Neighbor <peer group name> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.
Address-family ipv6	Enter address family configuration mode.
Neighbor <peer group name> activate	Exchange prefixes for the IPv6 router family between the neighbors of the specified peer group and the Citrix ADC by using the link local address.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# neighbor external-peers-2 peer-group
4      NS(config-router)# neighbor 2001::1 peer-group external-peers-2
5      NS(config-router)# neighbor 2001::2 peer-group external-peers-2
6      NS(config-router)# Neighbor external-peers-2 remote-as 100
7      NS(config-router)# Neighbor external-peers-2 as-override
8      NS(config-router)# Address-family ipv6
9      NS(config-router-af)# Neighbor external-peers-2 activate
10     NS(config-router)# Neighbor external-peers-2 as-override
11 <!--NeedCopy-->
```


Graceful Restart

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neighbor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped.

Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

> **Note:**
>
> Graceful restart is not supported for high availability setups in INC mode.

Configuring Graceful Restart for BGP

To configure graceful restart for BGP by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.
router-id <ID>	NS(config)# router-id 1.1.1.1	A router identifier for the Citrix ADC appliance. This identifier is set for all the dynamic routing protocols. The same identifier must be specified on the other node in a high availability setup for graceful restart to work properly.
router bgp <AS-number>	NS(config)# router bgp 5	Enters BGP configuration mode.
bgp graceful-restart	NS(config)# bgp graceful-restart	Enables graceful restart on the BGP routing process.

Command	Example	Command Description
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	Specifies the grace period, in seconds, that the helper routers waits for a TCP connection from the new primary node after a failover. For this amount of time, the helper routers preserve the routes.
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	Specifies the time, in seconds, that the Citrix ADC appliance in helper mode retains the stale routes for restarting neighbor routers. The default value is 360 seconds.
neighbor <IPv4 address of the peer router> remote-as <AS-number>	NS(config-router)# neighbor 192.0.2.30 remote-as 2	Establishes BGP peering with the specified neighbor router device.
neighbor <IPv4 address of the peer router> capability graceful-restart	NS(config-router)# neighbor 192.0.2.30 capability graceful-restart	Enables graceful restart with the specified neighbor.
redistribute kernel	NS(config-router)# redistribute kernel	Redistributes kernel routes.

Configuring Graceful Restart for IPv6 BGP

To configure graceful restart for IPv6 BGP by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.

Command	Example	Command Description
router-id <id>	NS(config)# router-id 1.1.1.1	Sets a router identifier for the Citrix ADC appliance. This identifier is set for all the dynamic routing protocols. The same ID must be specified in the other node in a high availability setup for graceful restart to work properly.
router bgp <AS-number>	NS(config)# router bgp 5	Enters configuration mode for BGP protocol.
bgp graceful-restart	NS(config)# bgp graceful-restart	Enables graceful restart on the BGP routing process.
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	Specifies the grace period, in seconds, that the helper routers waits for a TCP connection from the new primary node after a failover. For this amount of time, the helper routers preserve the routes. The default value is 360 seconds.
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	Specifies the time, in seconds, that the Citrix ADC appliance in helper mode retains the stale routes for restarting neighbor routers. The default value is 360 seconds.
neighbor <IPv6 address> remote-as <AS-number>	NS(config-router)# neighbor 2001:db8::10 remote-as 2	Establishes BGP peering with the specified neighbor router device.
address-family ipv6	NS(config-router)#address-family ipv6	Enters address family configuration mode.

Command	Example	Command Description
neighbor <IPv6 address of the neighbor> activate	NS(config-router-af)#neighbor 2001:db8::10 activate	Enables the exchange of address family routes with the specified neighbor router device.
neighbor <IPv6 address of the neighbor> capability graceful-restart	NS(config-router-af)#neighbor 2001:db8::10 capability graceful-restart	Enables graceful restart with the specified neighbor router device.
redistribute kernel	NS(config-router-af)#redistribute kernel	Redistributes kernel routes.
exit-address-family	NS(config-router-af)#exit-address-family	Exits address family configuration mode.

Configuring MD5 Authentication for IPv4 BGP

The Citrix ADC appliance supports MD5 authentication for Border Gateway Protocol (BGP). When authentication is enabled, any TCP segment belonging to BGP exchanged between the Citrix ADC appliance and its peer device is verified and accepted only if authentication is successful. For authentication to be successful, both the peers must be configured with the same MD5 password. If authentication fails, the BGP neighbor relationship is not being established. MD5 authentication support for BGP in the Citrix ADC appliance is compliant with RFC 2385.

Before you Begin

Before you start configuring BGP MD5 authentication, consider the following points:

- Make sure that you understand the different components of BGP MD5 authentication, described in RFC 2385.
- BGP MD5 authentication is not supported for Citrix ADC admin partitions.
- BGP MD5 authentication is not supported for IPv6 BGP configurations.
- BGP MD5 authentication is supported for Citrix ADC cluster configurations as well as for high availability configurations.
- Because of the following issue in FreeBSD, Citrix recommends to set a low keep-live and hold-time values (for example, 5 and 15) and configure graceful restart for a BGP session in a Layer 2 high availability configuration. Otherwise, with MD5 authentication enabled, BGP might take a longer time to re-establish a connection with the neighbour after a failover.
  - Last ACK from FreeBSD does not contain md5 digest:
    * https://forums.freebsd.org/threads/11170/

* http://support.pfsense.narkive.com/povrH5HI/bgp-md5-weird-behavior-when-connection-closes

Configuring MD5 Authentication for IPv4 BGP

To configure MD5 authentication for IPv4 BGP by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Displays VTYSH command prompt.
configure terminal	Enters global configuration mode.
router bgp <AS-number>	Enters configuration mode for BGP protocol. <AS-number> is a BGP autonomous system number and is a required parameter.
neighbor <neighbour IPv4 address> remote-as < AS-number >	Updates the IPv4 BGP table with the IPv4 address of the neighbor in the specified autonomous system.
neighbor < neighbour IPv4 address > password < password in double quotes>	Configures MD5 authentication for the specified neighbour with the specified MD5 password. For MD5 authentication to be successful, you must configure the same MD5 password on the Citrix ADC appliance and the neighbour appliance.

```
> vtysh

ns# configure terminal

ns(config)#router bgp 5

ns(config-router)#neighbor 20.20.20.138 remote-as 1

ns(config-router)#neighbor 20.20.20.138 password  “secret”

ns(config-router)#redistribute kernel

ns(config-router)#exit

<!--NeedCopy-->
```


Configuring IPv6 RIP

September 14, 2021

IPv6 Routing Information Protocol (RIP) or RIPng is a Distance Vector protocol. This protocol is an extension of RIP to support IPv6. After enabling IPv6 RIP, you need to configure advertisement of IPv6 RIP routes. For troubleshooting, you can limit IPv6 RIP propagation. You can display IPv6 RIP settings to verify the configuration.

Prerequisites for IPv6 RIP

Before you begin configuring IPv6 RIP, do the following:

- Make sure that you understand the IPv6 RIP protocol.
- Install the IPv6PT license on the Citrix ADC appliance.
- Enable the IPv6 feature.

Advertising IPv6 RIP Routes

IPv6 RIP enables an upstream router to load balance traffic between two identical vservers hosted on two standalone Citrix ADC devices. Route advertisement enables an upstream router to track network entities located behind the Citrix ADC.

To configure IPv6 RIP to advertise IPv6 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router ipv6 rip
```



```
4 NS(config-router)# redistribute static
5 NS(config-router)# redistribute kernel
6 <!--NeedCopy-->
```

Limiting IPv6 RIP Propagations

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given interface.

To limit IPv6 RIP propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router ipv6 rip
4 NS(config-router)# passive-interface VLAN0
5 <!--NeedCopy-->
```

Verifying the IPv6 RIP Configuration

You can use VTYSH to display the IPv6 RIP routing table and IPv6 RIP information for a specified VLAN.

To view the IPv6 RIP settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.

Commands	Specifies
sh ipv6 rip	Display updated IPv6 RIP routing table.
sh ipv6 rip interface <vlan_name>	Display IPv6 RIP information for the specified VLAN.

Example:

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
<!--NeedCopy-->
```

Configuring IPv6 OSPF

October 28, 2021

IPv6 OSPF or OSPF version 3 (OSPF v3) is a link state protocol that is used to exchange IPv6 routing information. After enabling IPv6 OSPF, you need to configure advertisement of IPv6 OSPF routes. For troubleshooting, you can limit IPv6 OSPF propagation. You can display IPv6 OSPF settings to verify the configuration.

Prerequisites for IPv6 OSPF

Before you begin configuring IPv6 OSPF, do the following:

- Make sure that you understand the IPv6 OSPF protocol.
- Install the IPv6PT license on the Citrix ADC appliance.
- Enable the IPv6 feature.

Advertising IPv6 Routes

IPv6 OSPF enables an upstream router to load balance traffic between two identical vservers hosted on two standalone Citrix ADC devices. Route advertising enables an upstream router to track network entities located behind the Citrix ADC.

To configure IPv6 OSPF to advertise IPv6 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

Limiting IPv6 OSPF Propagations

If you need to troubleshoot your configuration, you use VTYSH to configure listen-only mode on any given VLAN.

To limit IPv6 OSPF propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
passive-interface < vlan_name >	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
```



```
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the IPv6 OSPF Configuration

You use VTYSH to display IPv6 OSPF current neighbors and IPv6 OSPF routes.

To view the IPv6 OSPF settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh ipv6 OSPF neighbor	Display current neighbors.
sh ipv6 OSPF route	Display IPv6 OSPF routes.

Example:

```
>VTYSH
NS# sh ipv6 OSPF neighbor
NS# sh ipv6 OSPF route
<!--NeedCopy-->
```

OSPFv3 Authentication

To ensure the integrity, data origin authentication, and data confidentiality of OSPFv3 packets, OSPFv3 authentication must be configured on OSPFv3 peers.

The Citrix ADC appliance supports OSPFv3 authentication and is partially compliant with RFC 4552. OSPFv3 authentication is based on the two IPSec protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). The Citrix ADC appliance supports only the AH protocol for OSPFv3 authentication.

OSPFv3 authentication uses manually defined IPSec Security Associations (SAs) between the OSPFv3 peers and does not rely on IKE protocol for forming dynamic SAs. Manual SAs define the security parameter Index (SPI) values, algorithms, and keys to be used between the peers. Manual SAs require no negotiation between the peers; therefore, the same SA must be defined on both the peers.

You can configure OSPFv3 authentication on a VLAN or for an OSPFv3 area. When you configure for a

VLAN, the settings are applied to all the interfaces that are members of the VLAN. When you configure OSPFv3 authentication for an OSPF area, the settings are applied to all the VLANs in that area. The settings are in turn applied to all the interfaces that are members of these VLANs. These settings do not apply to member VLANs on which you have configured OSPFv3 authentication directly.

Consider the following points and limitations before configuring OSPFv3 authentication on a Citrix ADC appliance:

- Make sure that you understand the different components of OSPFv3 authentication, described in RFC 4552.
- Only Authentication Header protocol is supported for OSPFv3 authentication. Encapsulating Security Payload (ESP) is not supported.
- You must define an SA with the same setting on the peer interface.
- Rekeying of manual keys is not supported.

To configure OSPFv3 authentication on a VLAN by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown: OSPFv3 authentication VLAN commands.

Example:

```
1 > VTYSH NS# configure terminal
2 NS(config)# interface vlan2
3 NS(config-if)# ipv6 ospf authentication ipsec spi 256 md5 123456789
     ABCDEF0123456789ABCDEF0
4 <!--NeedCopy-->
```

To configure OSPFv3 authentication on an OSPF area by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown: OSPFv3 authentication OSPF area commands.

Example:

```
1 > VTYSH NS# configure terminal
2 ns(config)#router ipv6 ospf 30
3 ns(config-router)# area 1 authentication ipsec spi 256
     md5123456789ABCDEF0123456789ABCDEF0
4 <!--NeedCopy-->
```

Configuring Graceful Restart for IPv6 OSPF

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neigh-

bor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped.

Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

> **Note:**
>
> Graceful restart is not supported for high availability setups in INC mode.

To configure graceful restart for IPv6 OSPF by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	> VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.
router-id id>	NS(config)#router-id 1.1.1.1	Sets a router identifier for the Citrix ADC appliance. This identifier is set for all the dynamic routing protocols. The same ID must be specified in the other node in a high availability set up for graceful restart to work properly in the HA set up.
IPv6ospf restart grace-period <1-1800>	NS(config)# IPv6ospf restart grace-period 170	Specifies the grace period, in seconds, for which the routes are to be preserved in the helper devices. Default value: 120 seconds.

Command	Example	Command Description
IPv6 ospf restart helper max-grace-period <1-1800>	NS(config)# IPv6 ospf restart helper max-grace-period 180	This is an optional command to limit the maximum grace period for which the Citrix ADC appliance will be in the helper mode. If the Citrix ADC appliance receives an opaque LSA with grace-period greater than the set helper max-grace-period, the LSA is discarded and the Citrix ADC is not placed in helper mode.
interface <VLANID>	NS(config)#interface vlan3	Enters VLAN configuration mode.
ipv6 router ospf area <area_id> tag <tag_id>	NS(config-if)#ipv6 router ospf area 0 tag 1	Starts IPv6 OSPF routing process on a VLAN.
exit	NS(config-if)#exit	Exits VLAN configuration mode.
router ipv6 ospf	NS(config)# router ipv6 ospf 1	Starts IPv6 OSPF routing process and enters configuration mode for the routing process.
capability restart graceful	NS(config-router)#capability restart graceful	Enables graceful restart on the IPv6 OSPF routing process.
redistribute kernel	NS(config-router)# redistribute kernel	Redistributes kernel routes.

Configuring ISIS

September 14, 2021

The Citrix ADC appliance supports the Intermediate System-to-Intermediate System (IS-IS or ISIS) dynamic routing protocol. This protocol supports IPv4 as well as IPv6 route exchanges. IS-IS is a link state protocol and is therefore less prone to routing loops. With the advantages of faster convergence and the ability to support larger networks, ISIS can be very useful in Internet Service Provider (ISP)

networks.

Prerequisites for configuring ISIS

Before you begin configuring ISIS, do the following:

- Make sure that you understand the ISIS protocol.
- For IPV6 routes, enable:
  - IPv6 protocol translation feature.
  - IPv6 Dynamic Routing option on the VLANs on which you want to run ISIS protocol.

Enabling ISIS

Use either of the following procedures to enable the ISIS routing feature on the Citrix ADC appliance.

To enable ISIS routing by using the CLI:

At the command prompt, type:

enable ns feature ISIS

To enable ISIS routing by using the GUI:

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the ISIS Routing option.

Creating an ISIS Routing Process and Starting It on a VLAN

To create an ISIS routing process, you must use the VTYSH command line.

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router ISIS [tag]	Creates an ISIS routing process and configuration mode for the routing process.
net XX….XXXX.YYYY.YYYY.YYYY.00	Specifies a NET value for the routing process, where: ·**XX. .. .XXXX** is the Area Address (can be 1-13 bytes), ·**YYYY.YYYY.YYYY** is the System ID (6 bytes), ·**00** is the N-selector (1 byte).
is-type (level-1|level-1-2|level-2-only)	Sets the ISIS routing process to the specified level of routing. Default: level-1-2.

Command	Description
ns IPv6-routing	Starts the IPv6 dynamic routing daemon.
interface <vlan_name>	Enters the VLAN configuration mode.
ip router ISIS	Enables the ISIS routing process on the VLAN for IPv4 route exchanges.
ipv6 router ISIS	Enables the ISIS routing process on the VLAN for IPv6 route exchanges.

Example:

```
> VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# net 15.aabb.ccdd.0097.00
NS(config-router)# is-type level-1
NS(config-router)# exit
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)#  ip router isis 11
NS(config-if)# ipv6 router isis 11
<!--NeedCopy-->
```

Advertising Routes

Route advertisement enables an upstream router to track network entities located behind the Citrix ADC appliance.

To configure ISIS to advertise routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays the VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router ISIS [tag]	Starts the ISIS routing instance and enter configuration mode for the routing process.

Command	Description
redistribute connected (level-1 or level-1-2 or level-2)	Redistributes connected routes, where: **level-1**: Redistribute connected routes into Level-1, **level-1-2**: Redistribute connected routes into Level-1 and Level-2, **level-2**: Redistribute connected routes into Level-2.
redistribute kernel (level-1 or level-1-2 or level-2)	Redistributes kernel routes, where: **level-1**: Redistribute kernel routes into Level-1, **level-1-2**: Redistribute kernel routes into Level-1 and Level-2, **level-2**: Redistribute kernel routes into Level-2.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# redistribute connected level-1
NS(config-router)# redistribute kernel level-1
<!--NeedCopy-->
```

Limiting ISIS Propagations

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given VLAN.

To limit ISIS propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays the VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router isis [tag]	Enters the configuration mode for the routing process.
passive-interface <vlan_name>	Suppresses routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the ISIS Configuration

You can use VTYSH to display the ISIS routing table and ISIS information for a specified VLAN.

To view the ISIS settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Description
VTYSH	Displays the VTYSH command prompt.
show ip isis route	Displays updated IPv4 ISIS routing table.
show ipv6 isis route	Displays updated IPv6 ISIS routing table.
sh isis interface <vlan_name>	Displays IPv6 ISIS information for the specified VLAN.

Example:

```
NS# VTYSH
NS#  show ip isis route
NS#  show ipv6 isis route
NS# sh isis interface VLAN0
<!--NeedCopy-->
```

Install Routes to the Citrix ADC Routing Table

September 14, 2021

The Citrix ADC appliance can use routes learned by various routing protocols after you install the routes in the appliance's routing table.

To install various routes to the internal routing table by using the VTYSH command line:

At the CLI, type the following commands as appropriate for the routes that you want to install:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns route-install Default	Install IPv4 default routes to the internal routing table.
ns route-install RIP	Install IPv4 RIP specific routes to the internal routing table.
ns route-install BGP	Install IPv4 BGP specific routes to the internal routing table.
ns route-install OSPF	Install IPv4 OSPF specific routes to the internal routing table.
ns route-install IPv6 Default	Install IPv6 default routes to the internal routing table.
ns route-install IPv6 RIP	Install IPv6 RIP specific routes to the internal routing table.
ns route-install IPv6 BGP	Install IPv6 BGP specific routes to the internal routing table.
ns route-install IPv6 OSPF	Install IPv6 OSPF specific routes to the internal routing table.

Example:

```
>VTYSH
NS# configure terminal
NS# ns route-install Default
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
NS(config)# ns route-install IPv6 BGP
NS(config)# ns route-install IPv6 OSPF
<!--NeedCopy-->
```


Advertisement of SNIP and VIP Routes to Selective Areas

September 14, 2021

To advertise some SNIP addresses to selective areas, enabling DRADV mode or redistribute connect ZebOS operations cannot be used. This is because these operations send all the connected routes to ZebOS. Also, adding dummy static routes in ZebOS for the required subnets, or adding ACLs in ZebOS to filter unwanted connected routes, is a cumbersome and tedious task.

The Network Route and the Tag options address this issue. You can enable the Network Route option for only one SNIP address per subnet. The connected route for that SNIP address is sent as a kernel route to ZebOS.

For VIP and SNIP addresses, Tag, can be assigned an integer from 1 to 4294967295. This parameter can be set only when Host Route or Network Route is enabled for VIP or SNIP addresses. The tag value associated with VIP and SNIP addresses are also sent along with their routes to ZebOS. Tags with different values can be set for VIP and SNIP routes. These tag values can then be matched in route maps in ZebOS and advertised to selective areas.

Advertise SNIP Routes to Selective Areas

To configure the network route and tag parameters of a SNIP address by using the CLI:

At the command prompt, type:

- If adding a new SNIP address:
  - **add ns ip** <IPAddress>@ <netmask> -**type SNIP** -**networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- If reconfiguring an existing SNIP address:
  - **set ns ip** <IPAddress>@ <netmask> -**type SNIP** - **networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

To configure the network route and tag parameters of a SNIP address by using GUI:

1. Navigate to **System**> **Network** > **IPs** > **IPV4s**.
2. Set the **Network Route** and **Tag** parameters while adding a Subnet IP (SNIP) address or modifying an existing Subnet IP address.

Advertise VIP Routes to Selective Areas

To configure the host route and tag parameters of a VIP address by using the CLI:

At the command prompt, type one of the following sets of commands.

- If adding a new VIP address:
  - **add ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- If reconfiguring an existing VIP address:
  - **set ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

To configure the network route and tag parameters of a VIP address by using the GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Set the **Host Route** and **Tag** parameters while adding a VIP address or modifying an existing VIP address.

Configuring Bidirectional Forwarding Detection

September 14, 2021

Bidirectional Forwarding Detection (BFD) protocol is a mechanism for fast detection of failures of forwarding paths. BFD detects path failures in the order of milliseconds. BFD is used with dynamic routing protocols.

In BFD operation, routing peers exchange BFD packets at a negotiated interval. If a packet is not received from a peer within the negotiated interval plus grace interval, the peer is considered to be dead and a notification will be sent to the set of registered routing protocols. In turn, the routing protocols recalculate the best path and reprogram the routing table. BFD supports smaller time interval, when compared to the timers provided by the routing protocols, thus resulting in faster detection of failures.

The Citrix ADC appliance supports BFD for the following routing protocols: BGP (IPv4 and IPv6), OSPFv2 (IPv4), and OSPFv3 (IPv6). BFD support in the Citrix ADC appliance is compliant with RFCs 5880, 5881, and 5883.

Points to Consider for Configuring Bidirectional Forwarding Detection

Before you start configuring BFD, consider the following points:

- Make sure that you understand the different components of BFD, described in RFCs 5880, 5881, and 5883.
- BFD on a Citrix ADC appliance is supported for the following routing protocols:

  - BGP (IPv4 and IPv6)
  - OSPFv2 (IPv4)
  - OSPFv3 (IPv6)
- BFD on a Citrix ADC appliance is not supported for the following routing protocols:
  - ISIS
  - RIP (IPv4)
  - RIPng (IPv6)
- The following BFD features are not supported on a Citrix ADC appliance:
  - BFD Echo mode
  - BFD Authentication
  - BFD Demand asynchronous mode
- The minimum values for BFD interval and BFD Rx timers are 100 milliseconds.
- When BFD is used in a topology with shared IP addresses (for example, Layer 2 high availability setup with SNIP addresses or a cluster setup with striped IP addresses), BFD brings down the active sessions during a failover because the BFD failure detection time (order of milliseconds) is lesser than the HA failover detection interval (3–4 seconds). Therefore, Citrix recommends usage of Graceful restart in layer-2 HA topologies as the routes are retained during the failover process.

Configuration Steps

Configuring BFD on a Citrix ADC appliance consists of the following tasks:

- Configure BFD Parameters
- Configure BFD Support for Dynamic Routing Protocols

Configure BFD Parameters

The Citrix ADC appliance provides separate BFD session parameters for single hop sessions, IPv4 multiple hop sessions, and IPv6 multiple hop sessions. If you do not configure BFD parameters for a type of session, the default values are applied for that session.

The default value of each BFD parameter is same for single hop sessions, IPv4 multiple hop sessions, and IPv6 multiple hop sessions. The following table displays the default value of each BFD parameter.

BFD Parameter Name	Default Value
Interval	750 milliseconds
Minimum Rx	500 milliseconds
Multiplier	3

> **IMPORTANT:**
>
> Mellanox NICs in a Citrix appliance take around 1500 ms to initialize. You must set the BFD timers to more than 1500 ms for a Citrix ADC appliance with Mellanox NICs. Citrix recommends setting the BFD timers to 3000 ms:
>
> - Interval Tx = 600 ms
> - Minimum Rx = 600 ms
> - Multipler = 5

Configuring BFD Parameters for a Single Hop Session

To configure BFD parameters for a single hop session by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
`vtysh`	Display VTYSH command prompt.
`configure terminal`	Enter global configuration mode.
`interface vlan ID>`	Enter the interface configuration mode.
`bfd singlehop-peer interval <num> minrx <num> multiplier <num>`	Configure the BFD parameters on the specified interface.

Sample configuration:

```
> vtysh

ns# configure terminal

ns(config)# interface vlan3

ns(config-if)# bfd singlehop-peer interval 200 minrx 200 multiplier 5

ns(config-if)# exit
<!--NeedCopy-->
```

Configuring BFD Parameters for IPv4 Multiple Hop Sessions

To configure BFD parameters for IPv4 multiple hop sessions by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
`vtysh`	Display VTYSH command prompt.
`configure terminal`	Enter global configuration mode.
`bfd multihop-peer <ipv4addr> interval <num> minrx <num> multiplier <num>`	Configure the BFD parameters for IPv4 multiple hops sessions.

Sample configuration:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)# bfd multihop-peer 20.20.20.138 interval 300 minrx 300
           multiplier 5
6
7      ns(config)# exit
8  <!--NeedCopy-->
```

Configuring BFD Parameters for IPv6 Multiple Hop Sessions

To configure BFD parameters for IPv6 multiple hop sessions by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
`vtysh`	Display VTYSH command prompt.
`configure terminal`	Enter global configuration mode.
`bfd multihop-peer ipv6 <ipv6addr> interval <num> minrx <num> multiplier <num>`	Configure the BFD parameters for IPv6 multiple hops sessions.

Sample configuration:

```
1      > vtysh
2
3      ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx
           500 multiplier 5
```



```
4
5     ns(config)# exit
6  <!--NeedCopy-->
```

Configure BFD Support for Dynamic Routing Protocols

You can enable BFD for a dynamic routing protocol for a type of session with a peer. For example, single hop and multiple hops. The Citrix ADC appliance applies the relevant BFD parameter settings to the session.

Configuring BFD for an IPv4 BGP Single Hop Session

To configure BFD for an IPv4 BGP single hop session by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router bgp <asnumber>	BGP autonomous system. asnumber is a required parameter.
neighbor <ipv4addr> remote-as <num>	Update the IPv4 BGP table with the IPv4 address of the neighbor in the specified autonomous system.
neighbor <ipv4addr> fall-over bfd	Enable BFD for the specified neighbor.

Sample configuration:

```
1     > vtysh
2
3     ns# configure terminal
4
5     ns(config)#router bgp 1
6
7     ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9     ns(config-router)#neighbor 20.20.20.138 fall-over bfd
10
11    ns(config-router)#redistribute kernel
12
```



```
13      ns(config-router)#exit
14  <!--NeedCopy-->
```

Configuring BFD for an IPv4 BGP Multiple Hop Session

To configure BFD for an IPv4 BGP multiple hop session by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router bgp <asnumber>	BGP autonomous system. asnumber is a required parameter.
neighbor <ipv4addr> remote-as <num>	Update the IPv4 BGP table with the IPv4 address of the neighbor in the specified autonomous system.
neighbor <ipv4addr> fall-over bfd multihop	Enable BFD for the specified neighbor.

Sample configuration:

```
1       > vtysh
2
3       ns# configure terminal
4
5       ns(config)#router bgp 1
6
7       ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9       ns(config-router)#neighbor 20.20.20.138 fall-over bfd multihop
10
11      ns(config-router)#redistribute kernel
12
13      ns(config-router)#exit
14  <!--NeedCopy-->
```


Configuring BFD for an IPv6 BGP Single Hop Session

To configure BFD for an IPv6 BGP single hop session by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router bgp <asnumber>	BGP autonomous system. asnumber is a required parameter.
neighbor <ipv6addr> remote-as <num>	Update the IPv6 BGP table with the link local IPv6 address of the neighbor in the specified autonomous system.
neighbor <ipv6addr> fall-over bfd	Enable BFD for the specified neighbor.
address-family ipv6	Enter address family configuration mode.
neighbor <ipv6addr> activate	Exchange prefixes for the IPv6 router family between the peer and the local node by using the link local address.

Sample configuration:

```
> vtysh

ns# configure terminal ns(config)#router bgp 1

ns(config-router)#neighbor 30fe:123::124 remote-as 1

ns(config-router)#neighbor 30fe:123::124 fall-over bfd

ns(config-router)#address-family ipv6

ns(config-router-af)#neighbor 30fe:123::124 activate

ns(config-router-af)#redistribute kernel

ns(config-router-af)#exit

<!--NeedCopy-->
```


Configuring BFD for an IPv6 BGP Multiple Hop Session

To configure BFD for an IPv6 BGP multiple hop session by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router bgp <asnumber>	BGP autonomous system. asnumber is a required parameter.
neighbor <ipv6addr> remote-as <num>	Update the IPv6 BGP table with the link local IPv6 address of the neighbor in the specified autonomous system.
neighbor <ipv6addr> fall-over bfd multihop	Enable BFD for the specified neighbor.
address-family ipv6	Enter address family configuration mode.
neighbor <ipv6addr> activate	Exchange prefixes for the IPv6 router family between the peer and the local node by using the link-local address.

Sample configuration:

```
> vtysh

ns# configure terminal

ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx 500
    multiplier 5

ns(config)#router bgp 1

ns(config-router)#neighbor 20fe:125::138 remote-as 1

ns(config-router)#neighbor 20fe:125::138 fall-over bfd multihop

ns(config-router)#address-family ipv6

ns(config-router-af)#neighbor 20fe:125::138 activate

ns(config-router-af)#redistribute kernel
```



```
18
19 ns(config-router-af)#end
20
21 <!--NeedCopy-->
```

Configuring BFD for OSPFv2 (IPv4) on Interfaces

You can enable BFD on all or on a specific interface that uses the OSPFv2 protocol.

To configure BFD for OSPFv2 on all interfaces by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ospf <process tag>	Enter OSPFv2 configuration mode.
bfd all-interfaces	Enable BFD on all interfaces that use OSPFv2.

Sample configuration:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router ospf 1
6
7      ns(config-router)#bfd all-interfaces
8
9      ns(config-router)#redistribute kernel
10
11     ns(config-router)#exit
12 <!--NeedCopy-->
```

To configure BFD for OSPFv2 on a specific interface by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
vtysh	Display VTYSH command prompt.

Command	Specifies
`configure terminal`	Enter global configuration mode.
`interface` <vlan ID>	Enter the interface configuration mode.
`ip ospf bfd`	Enable BFD on the specified interface that uses OSPFv2.

Sample configuration:

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)# interface vlan5
6
7      ns(config-if)# ip ospf bfd
8
9      ns(config-if)# exit
10 <!--NeedCopy-->
```

Configuring BFD for OSPFv3 (IPv6) on Interfaces

You can enable BFD on all or on a specific interface that uses the OSPFv3 protocol.

To configure BFD for OSPFv3 on all interfaces by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
`vtysh`	Display VTYSH command prompt.
`configure terminal`	Enter global configuration mode.
`router ipv6 ospf <process tag>`	Enter OSPFv3 configuration mode.
`bfd all-interfaces`	Enable BFD on all interfaces that use OSPFv3.

Sample configuration:

```
1      > vtysh
2
3      ns# configure terminal
```



```
4
5     ns(config)#router ipv6 ospf 10
6
7     ns(config-router)#bfd all-interfaces
8
9     ns(config-router)#redistribute kernel
10
11    ns(config-router)#exit
12 <!--NeedCopy-->
```

To configure BFD for OSPFv3 on a specific interface by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
`vtysh`	Display VTYSH command prompt.
`configure terminal`	Enter global configuration mode.
`interface` <vlan ID>	Enter the interface configuration mode.
`ipv6 ospf bfd`	Enable BFD on the specified interface that uses OSPFv3.

Sample configuration:

```
1     > vtysh
2
3     ns# configure terminal
4
5     ns(config)# interface vlan15
6
7     ns(config-if)# ipv6 ospf bfd
8
9     ns(config-if)# exit
10 <!--NeedCopy-->
```

Configuring Static Routes

September 14, 2021

Static routes are manually created to improve the performance of your network. You can monitor

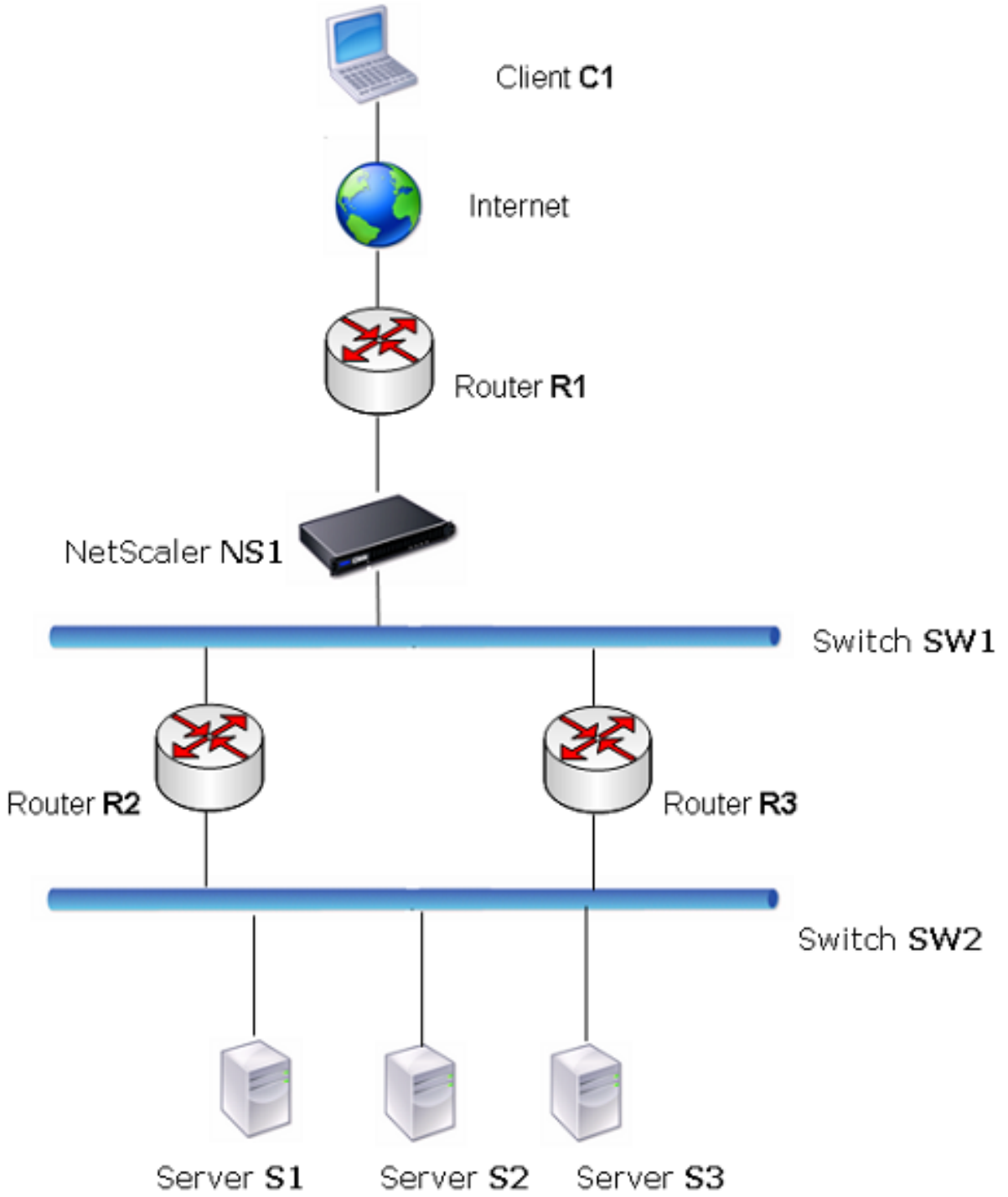
static routes to avoid service disruptions. Also, you can assign weights to ECMP routes, and you can create null routes to prevent routing loops.

Monitored Static Routes. If a manually created (static) route goes down, a backup route is not automatically activated. You must manually delete the inactive primary static route. However, if you configure the static route as a monitored route, the Citrix ADC appliance can automatically activate a backup route.

Static route monitoring can also be based on the accessibility of the subnet. A subnet is usually connected to a single interface, but it can be logically accessed through other interfaces. Subnets bound to a VLAN are accessible only if the VLAN is up. VLANs are logical interfaces through which packets are transmitted and received by the Citrix ADC. A static route is marked as DOWN if the next hop resides on a subnet that is unreachable.

Note: In a high availability (HA) setup, the default value for monitored state routes (MSRs) on the secondary node is UP. The value is set to avoid a state transition gap upon failover, which could result in dropping packets on those routes.

Consider the following simple topology, in which a Citrix ADC is load balancing traffic to a site across multiple servers.



Router R1 moves traffic between the client and the Citrix ADC appliance. The appliance can reach servers S1 and S2 through routers R2 or R3. It has two static routes through which to reach the servers'

subnet, one with R2 as the gateway and another with R3 as the gateway. Both these routes have monitoring enabled. The administrative distance of the static route with gateway R2 is lower than that of the static route with gateway R3. Therefore, R2 is preferred over R3 to forward traffic to the servers. Also, the default route on the Citrix ADC points to R1 so that all Internet traffic exits properly.

If R2 fails while monitoring is enabled on the static route, which uses R2 as the gateway, the Citrix ADC marks it as DOWN. The Citrix ADC now uses the static route with R3 as the gateway and forwards the traffic to the servers through R3.

The Citrix ADC supports monitoring of IPv4 and IPv6 static routes. You can configure the Citrix ADC to monitor an IPv4 static route either by creating a new ARP or PING monitor or by using existing ARP or PING monitors. You can configure the Citrix ADC to monitor an IPv6 static route either by creating a new Neighbor discovery for IPv6 (ND6) or PING monitor or by using the existing ND6 or PING monitors.

Weighted Static Routes. When the Citrix ADC appliance makes routing decisions involving routes with equal distance and cost, that is, Equal Cost Multi-Path (ECMP) routes, it balances the load between them by using a hashing mechanism based on the source and destination IP addresses. For an ECMP route, however, you can configure a weight value. The Citrix ADC then uses both the weight and the hashed value for balancing the load.

Null Routes. If the route chosen in a routing decision is inactive, the Citrix ADC appliance chooses a backup route. If all the backup routes become inaccessible, the appliance might reroute the packet to the sender, which could result in a routing loop leading to network congestion. To prevent this situation, you can create a null route, which adds a null interface as a gateway. The null route is never the preferred route, because it has a higher administrative distance than the other static routes. But it is selected if the other static routes become inaccessible. In that case, the appliance drops the packet and prevents a routing loop.

Configuring IPv4 Static Routes

You can add a simple static route or a null route by setting a few parameters, or you can set additional parameters to configure a monitored or monitored and weighted static route. You can change the parameters of a static route. For example, you might want to assign a weight to an unweighted route, or you might want to disable monitoring on a monitored route.

CLI procedures

To create a static route by using the CLI:

At the command prompt, type:

- add route <network> <netmask> <gateway>[-cost <positive_integer>] [-advertise (DISABLED | ENABLED)]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

Example:

```
> add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise
   ENABLED
 Done
<!--NeedCopy-->
```

To create a monitored static route by using the CLI:

At the command prompt, type the following commands to create a monitored static route and verify the configuration:

- add route <network> <netmask> <gateway> [-distance <positive_integer>] [-weight <positive_integer>][-msr (ENABLED | DISABLED) [-monitor <string>]]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

Example:

```
> add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6
    -msr ENBLED -monitor PING
 Done
<!--NeedCopy-->
```

To create a null route by using the CLI:

At the command prompt type:

- add route <network> <netmask> null
- show route <network> <netmask>

Example:

```
> add route 10.102.29.0 255.255.255.0 null
 Done
<!--NeedCopy-->
```

To remove a static route by using the CLI:

At the command prompt, type:

rm route <network> <netmask> <gateway>

Example:

```
> rm route 10.102.29.0 255.255.255.0 10.102.29.3
 Done
<!--NeedCopy-->
```


GUI procedures

To configure a static route by using the GUI:

Navigate to System > Network > Routes and, on the Basic tab, add a new static route, or edit an existing static route.

To remove a route by using the GUI:

Navigate to System > Network > Routes and, on the Basic tab, delete the static route.

Configuring IPv6 Static Routes

You can configure a maximum of six default IPv6 static routes. IPv6 routes are selected on the basis of whether the MAC address of the destination device is reachable. This can be determined by using the IPv6 Neighbor Discovery feature. Routes are load balanced and only source/destination-based hash mechanisms are used. Therefore, route selection mechanisms such as round robin are not supported. The next hop address in the default route need not belong to the NSIP subnet.

CLI procedures

To create an IPv6 route by using the CLI:

At the command prompt, type the following commands to create an IPv6 route and verify the configuration:

- add route6 <network> <gateway> [-vlan <positive_integer>]
- show route6 [<network> [<gateway>]

Example:

```
1 > add route6 ::/0 FE80::67 -vlan 5
2  Done
3 <!--NeedCopy-->
```

To create a monitored IPv6 static route by using the CLI:

At the command prompt, type the following commands to create a monitored IPv6 static route and verify the configuration:

- add route6 <network> <gateway> [-msr (ENABLED | DISABLED) [-monitor <string>]
- show route6 [<network> [<gateway>]

Example:

```
1 > add route6 ::/0 2004::1 -msr ENABLED -monitor PING
2  Done
3 <!--NeedCopy-->
```


To remove an IPv6 route by using the CLI:

At the command prompt, type:

rm route6 <network> <gateway>

Example:

```
> rm route6 ::/0 FE80::67
 Done
<!--NeedCopy-->
```

GUI procedures

To configure an IPv6 route by using the GUI:

Navigate to System > Network > Routes and, on the IPV6 tab, add a new IPv6 route, or edit an existing IPv6 route.

To remove an IPv6 route by using the GUI:

Navigate to System > Network > Routes and, on the IPV6 tab, delete the IPv6 route.

Route Health Injection Based on Virtual Server Settings

September 14, 2021

The following option and parameter are introduced for controlling the Route Health Injection (RHI) functionality of the Citrix ADC appliance for advertising the route of a VIP address.

- **VSVR_CNTRLD**. It is an option for the (Vserver RHI Level) parameter of a VIP address. When this option is set to the Vserver RHI Level parameter, the RHI behavior for advertising the route of the VIP address depends on the RHI STATE parameter setting on all the associated virtual servers of the VIP address along with their states.
- **RHI STATE**. It is a parameter of virtual server. You can set the RHI STATE parameter to either PASSIVE or ACTIVE. By default, the RHI STATE parameter is set to PASSIVE.

For a VIP address, when RHI (Vserver RHI Level) parameter is set to VSVR_CNTRLD, the following are different RHI behaviours for the VIP address on the basis of RHI STATE settings on the virtual servers associated with the VIP address:

- If you set RHI STATE to PASSIVE on all virtual servers, the Citrix ADC always advertises the route for the VIP address.
- If you set RHI STATE to ACTIVE on all virtual servers, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state.

- If you set RHI STATE to ACTIVE on some and PASSIVE on others, the Citrix ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state.

Following table displays the sample RHI behaviour for a VIP address on the basis of RHI STATE settings on the virtual servers associated with the VIP address. The Citrix ADC appliance has two virtual servers V1 and V2 associated with the VIP address:

Associated virtual servers for a VIP	State 1	State 2	State 3	State 4
RHI State set to PASSIVE on all virtual servers				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	Yes	Yes
RHI State set to ACTIVE on all virtual servers				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	Yes	No
RHI State set to ACTIVE on one virtual server and PASSIVE on the other				
V1 (RHI State = ACTIVE)	UP	UP	DOWN	DOWN
V2 (RHI State = PASSIVE)	UP	DOWN	UP	DOWN

Associated virtual servers for a VIP	State 1	State 2	State 3	State 4
Advertise the route for this VIP address?	Yes	Yes	No	No

To configure RHI for a VIP address, to be based on the RHI (RHI State) parameter setting of the associated virtual servers, perform the following steps:

- Set the RHI (Vserver RHI Level) parameter to VSVR_CNTRLD for the VIP address.
- Set the RHI State parameter for each virtual server associated with the VIP address.

To set the vServer RHI Level for a VIP address by using the CLI:

At the command prompt, type:

- **set ns ip** <IPAddress> [-**vserverRHILevel** <vserverRHILevel>]

To set the RHI State parameter of a virtual server by using the CLI:

At the command prompt, type:

- **set lb vserver** <name> [-**RHIstate** (**PASSIVE** | **ACTIVE**)]

To set the vServer RHI Level for a VIP address by using GUI

1. Navigate to **System** > **Network** > **IPs**.
2. Select a VIP address, and then click **Edit**.
3. Set the **Vserver RHI Level** parameter to **VSVR_CNTRLD**, and then click **OK**.

To set the RHI State parameter of a virtual server by using GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Select a load balancing virtual server, and then click **Edit**.
3. Set the **RHI State** parameter, and then click **OK**.

Configuring Policy-Based Routes

September 14, 2021

Policy-based routing bases routing decisions on criteria that you specify. A policy-based route (PBR) specifies criteria for selecting packets and, typically, a next hop to which to send the selected packets. For example, you can configure the Citrix ADC appliance to route outgoing packets from a specific

IP address or range to a particular next hop router. Each packet is matched against each configured PBR, in the order determined by the specified priorities, until a match is found. If no match is found, or if the matching PBR specifies a DENY action, the Citrix ADC applies the routing table for normal destination-based routing.

A PBR bases routing decisions for the data packets on parameters such as source IP address, source port, destination IP address, destination port, protocol, and source MAC address. A PBR defines the conditions that a packet must satisfy for the Citrix ADC to route the packet. These actions are known as "processing modes." The processing modes are:

- **ALLOW.** The appliance sends the packet to the designated next-hop router.
- **DENY.** The Citrix ADC applies the routing table for normal destination-based routing.

You can create PBRs for outgoing IPv4 and IPv6 traffic.

Many users begin by creating PBRs and then modifying them. To activate a new PBR, you must apply it. To deactivate a PBR, you can either remove or disable it. You can change the priority number of a PBR to give it a higher or lower precedence.

Policy-Based Routes (PBR) for IPv4 Traffic

September 14, 2021

Configuring PBRs involves the following tasks:

- Create a PBR.
- Apply PBRs.
- (Optional) Disable or enable a PBR.
- (Optional) Renumber the priority of the PBR.

Creating or Modifying a PBR

You cannot create two PBRs with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR. The priority (an integer value) defines the order in which the Citrix ADC appliance evaluates PBRs. When you create a PBR without specifying a priority, the Citrix ADC automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR, the Citrix ADC performs an action. If the packet does not match the condition defined by the PBR, the Citrix ADC compares the packet against the PBR with the next highest priority.

Instead of sending the selected packets to a next hop router, you can configure the PBR to send them to a link load balancing virtual server to which you have bound multiple next hops. This configuration can provide a backup if a next hop link fails.

Consider the following example. Two PBRs, p1 and p2, are configured on the Citrix ADC and automatically assigned priorities 20 and 30. You need to add a third PBR, p3, to be evaluated immediately after the first PBR, p1. The new PBR, p3, must have a priority between 20 and 30. In this case, you can specify the priority as 25.

CLI procedures

To create a PBR by using the CLI:

At the command prompt, type:

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr

Example:

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -
   nexthop 10.102.29.77
 Done
<!--NeedCopy-->
```

To modify the priority of a PBR by using the CLI:

At the command prompt, type the following commands to modify the priority and verify the configuration:

- set ns pbr <name> [-action (ALLOW | DENY)] [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> | -protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr [<name>]

Example:

```
> set ns pbr pbr1 -priority 23
 Done
```



```
<!--NeedCopy-->
```

To remove one or all PBRs by using the CLI:

At the command prompt, type one of the following commands:

- rm ns pbr <name>
- clear ns pbrs

Example:

```
> rm ns pbr pbr1
 Done

> clear ns PBRs
 Done
<!--NeedCopy-->
```

GUI procedures

To create a PBR by using the GUI:

Navigate to System > Network > PBRs, on the PBRs tab, add a new PBR, or edit an existing PBR.

To remove one or all PBRs by using the GUI:

Navigate to System > Network > PBRs, on the PBRs tab, delete the PBR.

Applying a PBR

You must apply a PBR to activate it. The following procedure reapplies all PBRs that you have not disabled. The PBRs constitute a memory tree (lookup table). For example, if you create 10 PBRs (p1 - p10), and then you create another PBR (p11) and apply it, all of the PBRs (p1 - p11) are freshly applied and a new lookup table is created. If a session has a DENY PBR related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR. For example, you must follow this procedure after disabling a PBR.

Note: PBRs created on the Citrix ADC appliance do not work until they are applied.

To apply a PBR by using the CLI:

At the command prompt, type:

apply ns PBRs

To apply a PBR by using the GUI:

1. Navigate to System > Network > PBRs.

2. On the PBRs tab, select the PBR, in the Action list, select Apply.

Enabling or Disabling PBRs

By default, the PBRs are enabled. This means that when PBRs are applied, the Citrix ADC appliance automatically compares incoming packets against the configured PBRs. If a PBR is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBRs are applied. After the PBRs are applied, the Citrix ADC does not compare incoming packets against disabled PBRs.

To enable or disable a PBR by using the CLI:

At the command prompt, type one of the following commands:

- enable ns pbr <name>
- disable ns pbr <name>

Example:

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
        Vlan:                                   Interface:
        Active Status: ENABLED                  Applied Status: APPLIED
        Priority: 10
        NextHop: 10.102.29.77

 Done

> disable ns PBR pbr1
Warning: PBR modified, use 'apply pbrs' to commit this operation

> apply pbrs
 Done

> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
```



```
28          Vlan:                                   Interface:
29          Active Status: DISABLED                 Applied Status:
               NOTAPPLIED
30          Priority: 10
31          NextHop: 10.102.29.77
32  Done
33  <!--NeedCopy-->
```

To enable or disable a PBR by using the GUI:

1. Navigate to System > Network > PBRs.
2. On the PBRs tab, select the PBR, in the Action list, select Enable or Disable.

Renumbering PBRs

You can automatically renumber the PBRs to set their priorities to multiples of 10.

To renumber PBRs by using the CLI:

At the command prompt, type:

- renumber ns pbrs

To renumber PBRs by using the GUI:

Navigate to System > Network > PBRs, on the PBRs tab, in the Action list, select Renumber Priority (s).

Use Case - PBR with Multiple Hops

Consider a scenario in which two PBRs, PBR1 and PBR2, are configured on Citrix ADC appliance NS1. PBR1 routes all the outgoing packets, with source IP address as 10.102.29.30, to next hop router R1. PBR2 routes all the outgoing packets, with source IP address as 10.102.29.90, to next hop router R2. R3 is another next hop router connected to NS1.

If router R1 fails, all the outgoing packets that matched against PBR1 are dropped. To avoid this situation, you can specify a link load balancing (LLB) virtual server in the next hop field while creating or modifying a PBR. Multiple next hops are bound to the LLB virtual server as services (for example R1, R2, and R3). Now, if R1 fails, all the packets that matched against PBR1 are routed to R2 or R3 as determined by the LB method configured on the LLB virtual server.

The Citrix ADC appliance throws an error if you attempt to create a PBR with an LLB virtual server as the next hop in the following cases:

- Adding another PBR with the same LLB virtual server.
- Specifying a nonexistent LLB virtual server.
- Specifying an LLB virtual server for which the bound services are not next hops.

- Specifying an LLB virtual server for which the LB method is not set to one of the following:
  - ROUNDROBIN
  - DESTINATIONIPHASH
  - SOURCEIPHASH
  - SRCIPDESTIPHASH
  - LEASTPACKETS
  - LEASTBANDWIDTH
  - LTRM
  - CALLIDHASH
  - CUSTOM LOAD
- Specifying an LLB virtual server for which the LB persistence type is not set to one of the following:
  - DESTIP
  - SOURCEIP
  - SRCDSTIP

The following table lists the names and values of the entities configured on the Citrix ADC appliance:

Entity Type	Name	IP Address
Link load balancing virtual server	LLB1	NA
Services (next hops)	Router1	1.1.1.254
	Router2	2.2.2.254
	Router3	3.3.3.254
PBRs	PBR1	NA
	PBR2	NA

Table 1. Sample Values for Creating Entities

To implement the configuration described above, you need to:

1. Create services Router1, Router2, and Router3 that represent next hop routers R1, R2, and R3.
2. Create link load balancing virtual server LLB1 and bind services Router1, Router2, and Router3 to it.
3. Create PBRs PBR1 and PBR2, with next hop fields set as LLB1 and 2.2.2.254 (IP address of the router R2), respectively.

To create a service by using the CLI:

At the command prompt, type:

- add service <name> <IP> <serviceType> <port>
- show service <name>

Example:

```
> add service Router1 1.1.1.254 ANY *
 Done
> add service Router2 2.2.2.254 ANY *
 Done
> add service Router3 3.3.3.254 ANY *
 Done
<!--NeedCopy-->
```

To create a service by using the GUI:

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create a link load balancing virtual server and bind a service by using the CLI:

At the command prompt, type:

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

Example:

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
<!--NeedCopy-->
```

To create a link load balancing virtual server and bind a service by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server for link load balancing. Specify **ANY** in the **Protocol** field.
   Note: Make sure that
   Directly Addressable is unchecked.
2. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.

To create a PBR by using the CLI:

At the command prompt, type:

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-nextHop <nextHopVal>]
- show ns pbr

Example:

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
<!--NeedCopy-->
```

To create a PBR by using the GUI:

Navigate to System > Network > PBRs, on the PBRs tab, add a new PBR.

Policy-Based Routes (PBR6) for IPv6 Traffic

September 14, 2021

Configuring PBR6s involves the following tasks:

- Create a PBR6.
- Apply PBR6s.
- (Optional) Disable or enable a PBR6.
- (Optional) Renumber the priority of the PBR6.

Creating or Modifying a PBR6

You cannot create two PBR6s with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR6. The priority (an integer value) defines the order in which the Citrix ADC appliance evaluates PBR6s. When you create a PBR6 without specifying a priority, the Citrix ADC automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR6, the Citrix ADC performs an action. If the packet does not match the condition defined by the PBR6, the Citrix ADC compares the packet against the PBR6 with the next highest priority.

CLI procedures

To create a PBR6 by using the CLI:

At the command prompt, type:

- **add ns pbr6** <name> <action> [-srcIPv6 [<operator>] <srcIPv6Val>] [-srcPort [<operator>] <srcPortVal>] [-destIPv6 [<operator>] <destIPv6Val>] [-destPort [<operator>] <destPortVal>]

[-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-state (ENABLED | DISABLED)] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-nextHop <nextHopVal>] [-nextHopVlan <positive_integer>]

- show ns pbr

To modify or remove a PBR6 by using the CLI:

To modify a PBR6, type the **set pbr6** <name> command and the parameters to be changed, with their new values.

To remove one or all PBR6s by using the CLI:

At the command prompt, type one of the following commands:

- **rm ns pbr6** <name>
- **clear ns pbr6**

GUI procedures

To create or modify a PBR6 by using the GUI:

Navigate to System > Network > PBRs and, on the PBR6s tab, add a new PBR6, or edit an existing PBR6.

To remove one or all PBR6s by using the GUI:

Navigate to System > Network > PBRs and, on the PBR6s tab, delete the PBR6.

Applying PBR6s

You must apply a PBR6 to activate it. The following procedure reapplies all PBR6s that you have not disabled. The PBR6s constitute a memory tree (lookup table). For example, if you create 10 PBR6s (p6_1 - p6_10), and then you create another PBR6 (p6_11) and apply it, all of the PBR6s (p6_1 - p6_11) are freshly applied and a new lookup table is created. If a session has a DENY PBR6 related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR6. For example, you must follow this procedure after disabling a PBR6.

Note: PBR6s created on the Citrix ADC appliance do not work until they are applied.

To apply PBR6s by using the CLI:

At the command prompt, type:

- **apply ns PBR6**

To apply PBR6s by using the GUI:

1. Navigate to System > Network > PBRs.
2. On the PBR6s tab, select the PBR6, in the Action list, select Apply.

Enabling or Disabling a PBR6

By default, the PBR6s are enabled. This means that when PBR6s are applied, the Citrix ADC appliance automatically compares outgoing IPv6 packets against the configured PBR6s. If a PBR6 is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBR6s are applied. After the PBR6s are applied, the Citrix ADC does not compare incoming packets against disabled PBR6s.

To enable or disable a PBR6 by using the CLI:

At the command prompt, type one of the following commands:

- **enable ns pbr** <name>
- **disable ns pbr** <name>

To enable or disable a PBR6 by using the GUI:

1. Navigate to System > Network > PBRs.
2. On the PBR6s tab, select the PBR6, in the Action list, select Enable or Disable.

Renumbering PBR6s

You can automatically renumber the PBR6s to set their priorities to multiples of 10.

To renumber PBR6s by using the CLI:

At the command prompt, type:

- **renumber ns pbr6**

To renumber PBR6s by using the GUI:

Navigate to System > Network > PBRs, on the PBR6s tab, in the Action list, select Renumber Priority (s).

MAC Address Wildcard Mask for PBRs

September 14, 2021

A wildcard mask parameter has been introduced for extended PBRs and PBR6s and is used with the source MAC address parameter to define a range of MAC addresses to be match against the source MAC address of outgoing packets.

Wild card masks specify which hexadecimal digits of the MAC address are used and which hexadecimal digits are ignored. The wildcard mask parameter specifies a series of ones and zeroes and has a length of 12 digits. Each digit is a mask for the corresponding hexadecimal digit of the MAC address. A zero digit in the wildcard mask indicates that the corresponding hexadecimal digit of the MAC address must be considered and a one digit indicates that the corresponding hexadecimal digit to be ignored.

The wildcard mask should meet the following conditions:

- Has only one series of zeroes
- Has only one series of ones
- Start with a series of zeroes

The following are some of the examples of valid wildcard masks:

- 000000111111
- 000000011111
- 000011111111

The following are some of the examples of invalid wildcard masks:

- 000000111100
- 111110000000
- 010101010101

For an PBR rule, a wildcard mask of 000000111111 for MAC address 96:fa:95:1d:67:4a defines the MAC address range 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF. This MAC address range is matched against the source MAC address of the outgoing packets.

To specify a range of source MAC addresses in a PBR rule by using the CLI:

At the command prompt, type:

- **add ns pbr** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns pbr** <pbrname>

Example:

```
1 > add ns pbr PBR-1 ALLOW -srcip 192.0.2.34 -srcMac 96:fa:95:1d:67:4a
      - srcMacMask 000000111111 -nexthop 198.51.100.1
2
3  Done
```

To specify a range of source MAC addresses in an PBR6 rule by using the CLI:

At the command prompt, type:

- **add ns pbr6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show pbr6** <pbr6name>

Example:


```
> add ns pbr6 PBR6-1 ALLOW  -srcipv6 2001:db8:0::7 -srcMac 96:fa:95:1d
   :67:4a   -srcMacMask 000000001111 -nexthop 2001:db8:0::1
 Done
```

Using NULL Policy Based Routes to Drop Outgoing Packets

September 14, 2021

Some situations might demand that the Citrix ADC appliance drops specific outgoing packets instead of routing them, for example, in testing cases and during deployment migration.

NULL policy based routes can be used to drop specific outgoing packets. A NULL PBR is a type of PBR that has the nexthop parameter set to NULL. The Citrix ADC appliance drops outgoing packets that match a NULL PBR.

Configuring NULL PBRs for IPv4 Packets

To create a NULL PBR by using the CLI:

At the command prompt, type:

- **add ns pbr** <name> ALLOW [**-td** <positive_integer>] [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] (-**nextHop NULL**) [**srcMac** <mac_addr> [-**srcMacMask** <string>]] [-**protocol** <protocol> | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer> | -**vxlan** <positive_integer>] [-**interface** <interface_name>] [-**priority** <positive_integer>] [-**msr** (**ENABLED** | **DISABLED**) [-**monitor** <string>]] [-**state** (**ENABLED** | **DISABLED**)[-**ownerGroup** <string>]
- **apply ns pbrs**
- **show ns pbr**<id>

To configure a NULL PBR by using the GUI:

Navigate to **System** > **Network**> **PBRs**, on the **PBRs** tab, add a **new NULL PBR**, or edit an existing NULL PBR.

Sample configuration

In the following sample configuration, NULL PBR6 PBR6-NULL-EXAMPLE-1 is configured for dropping any outgoing IPv6 packets from interface 1/5.


```
> add ns pbr PBR6-NULL-EXAMPLE-1 ALLOW  -nextHop NULL -interface 1/5
 Done

> apply ns pbr6
 Done
```

Traffic distribution in multiple routes based on five tuples information

September 14, 2021

In a load balancing setup, a Citrix ADC appliance can have multiple routes to send a packet to its destination. For example: to a server, and to a client.

A Citrix ADC appliance uses a hashing algorithm to select a route for sending the packet to its destination.

The hashing algorithm uses the following two tuples of a packet to calculate a hash, based on which the Citrix ADC appliance selects a route for the packet.

- Source IP address
- Destination IP address

The selection of routes based on two tuples information can cause an uneven distribution of traffic on the available routes. This uneven distribution of traffic leads to overloading of traffic in some routes.

To resolve this issue, from build 13.0 71.x, the Citrix ADC appliance uses the following five tuples information of a packet in the hashing algorithm to select a route for the packet:

- Source IP address (Client IP)
- Source Port (Client Port)
- Destination IP address(Service IP)
- Destination Port (Service Port)
- Protocol number

The selection of routes based on five tuples information ensures even distribution of traffic on the available routes. This even distribution of traffic prevents overloading of traffic in a route.

Consider an example of a load balancing setup where a client sends a request to the VIP address. The Citrix ADC appliance uses the following five tuples information to select a route to send the request packet to the load balanced server:

- Source IP address (Client IP address)
- Source Port (Client port)
- Destination IP address (Service IP address)

- Destination Port (Service port number)
- Protocol Number

Precedence regarding other route selection based Citrix ADC features

This section talks about the precedence of the route selection based on five tuples feature and other route selection related features in a Citrix ADC appliance.

- **Policy Based Routes (PBR)**. PBR rules always take precedence over route selection based on five tuples.
- **Mac Based Forwarding (MBF)**. In a load balancing configuration, MBF or route selection based on five tuples takes precedence in the following cases:
  - For a client initiated traffic to the VIP address of the load balancing configuration in the Citrix ADC appliance:
    * Request traffic destined to a load balanced server. Route selection based on five tuples takes preference over MBF.
    * Response Traffic destined to the client. MBF takes preference over route selection based on five tuples.
  - For a server initiated traffic to the SNIP address in the Citrix ADC appliance:
    * Response Traffic destined to the client. Route selection based on five tuples takes preference over MBF.
    * Request traffic destined to a load balanced server. MBF takes preference over route selection based on five tuples.

Troubleshooting Routing Issues

September 14, 2021

To make your troubleshooting process as efficient as possible, begin by gathering information about your network. You need to obtain the following information about the Citrix ADC appliance and other systems in the Network:

- Complete Topology diagram, including interface connectivity and intermediate switch details.
- Running Configuration. You can use the show running command to get the running configuration for ns.conf and ZebOS.conf.
- Output of the History command, to determine whether any configuration changes were made when the issue arose.
- Output of the Top and ps -ax commands, to determine whether any routing daemon is over utilizing the CPU or is misbehaving.

- Any routing related core files in /var/core - nsm, bgpd, ospfd, or ripd. Check the time stamp to see if they are relevant.
- dr_error.log and dr_info.log files from /var/log.
- Output of the date command and time details for all relevant systems. Print dates across all devices one after another, so that the times on the log messages can be correlated with various events.
- Relevant ns.log, newnslog files.
- Configuration files, log files and command history details from upstream and downstream routers.

Generic Routing FAQs

September 14, 2021

Users typically have the following questions about how to troubleshoot generic routing issues:

- How do I save the config files?

  The write command from VTYSH saves only ZebOS.conf. Run the save ns config command from CLI to save both ns.conf and ZebOS.conf files.

- If I have configured both a static default route and a dynamically learned default route, which is the preferred default route?

  The dynamically learned route is the preferred default route. This behavior is unique to default routes. However, in case of the Network Services Module (NSM), unless the administrative distances are modified, a statically configured route in the RIB is preferred over a dynamic route. The route that is downloaded to the NSM FIB is the static route.

- How do I block the advertisement of default routes?

  The default route is not injected into ZebOS.

- How do I view the debug output of networking daemons?

  You can write debugging output from networking daemons to a file by entering the following log file command from the global configuration view in VTYSH:

  ```
  ns(config)# log file /var/ZebOS.log
  <!--NeedCopy-->
  ```

  You can direct debug output to the console by entering the terminal monitor command from VTYSH user view:

  ```
  ns# terminal monitor
  ```



```
<!--NeedCopy-->
```

- How do I collect cores of running daemons?

  You can use the gcore utility to collect cores of running daemons for processing by gdb. This might be helpful in debugging misbehaving daemons without bringing the whole routing operation to a standstill.

```
gcore [-s] [-c core] [executable] pid
<!--NeedCopy-->
```

  The -s option temporarily stops the daemon while gathering the core image. This is a recommended option, because it guarantees that the resulting image shows the core in a consistent state.

```
root@ns#gcore -s -c nsm.core /netscaler/nsm 342
<!--NeedCopy-->
```

- How do I run a batch of ZebOS commands?

  You can run a batch of ZebOS commands from a file by entering the VTYSH -f <file-name> command. This does not replace the running configuration, but appends to it. However, by including commands to delete the existing configuration in the batch file and then add those for the new, desired configuration, you can use this mechanism to replace a specific configuration:

```
!
router bgp 234
network 1.1.1.1 255.255.255.0
!
route-map bgp-out2 permit 10
 set metric 9900
 set community 8602:300
!
<!--NeedCopy-->
```

Troubleshooting OSPF-Specific Issues

September 14, 2021

Before you start debugging any OSPF specific issue, you must collect information from the Citrix ADC appliance and all systems in the affected LAN, including upstream and downstream routers. To begin, enter the following commands:

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route
6. show ip ospf database summary
    - If there are only few LSAs in the database, then enter show ip ospf database router, show ip ospf database A. network, show ip ospf database external, and other commands to get the full details of LSAs.
    - If there are a large number of LSAs in the database, enter the show ip ospf database self-originated command.
7. show ip ospf
8. show ns ip. This ensures that the details of all VIPs of interest are included.
9. Get the logs from peering devices and run the following command:

    ```
    gcore -s -c xyz.core /netscaler/ospfd <pid>
    ```

    Note: The gcore command is non-disruptive.

Collect additional information from the Citrix ADC as follows:

1. Enable logging of error messages by entering the following command from the global configuration view in VTYSH:

    ```
    ns(config)# log file /var/ospf.log
    <!--NeedCopy-->
    ```

2. Enable debugging ospf events and log them by using the following command:

    ```
    ns(config) #log file /var/ospf.log
    <!--NeedCopy-->
    ```

    Enable debug ospf lsa packet only if the number of LSAs in the database is relatively small (< 500).

Internet Protocol version 6 (IPv6)

September 14, 2021

A Citrix ADC appliance supports both server-side and client-side IPv6 and can therefore function as an IPv6 node. It can accept connections from IPv6 nodes (both hosts and routers) and from IPv4 nodes, and can perform Protocol Translation (RFC 2765) before sending traffic to the services.

The following table lists some of the IPv6 features that the Citrix ADC appliance supports.

Table 1. Some Supported IPv6 Features

IPv6 features
IPv6 addresses for SNIPs (NSIP6, VIP6, and SNIP6)
Neighbor Discovery (Address Resolution, Duplicated Address Detection, Neighbor Unreachability Detection, Router Discovery)
Management Applications (ping6, telnet6, ssh6)
Static Routing and Dynamic routing (OSPF, BGP, RIPng, and ISIS)
Port Based VLANs
Access Control Lists for IPv6 addresses (ACL6)
IPv6 Protocols (TCP6, UDP6, ICMP6)
Server Side Support (IPv6 addresses for vservers, services)
USIP (Use source IP) and DSR (Direct Server Return) for IPv6
SNMP and CVPN for IPv6
HA with native IPv6 node address
IPv6 addresses for MIPs
Path-MTU discovery for IPv6

Implementing IPv6 Support

You must enable IPv6 feature on a Citrix ADC appliance before you can use or configure it. If IPv6 is disabled, the Citrix ADC does not process IPv6 packets. It displays the following warning when you run an unsupported command:

```
"Warning: Feature(s) not enabled [IPv6PT]"
<!--NeedCopy-->
```

Use either of the following procedures to enable or disable IPv6.

CLI procedures

To enable or disable IPv6 by using the CLI:

At the command prompt, type one of the following commands:

- enable ns feature ipv6pt
- disable ns feature ipv6pt

GUI procedures

To enable or disable IPv6 by using the GUI:

1. Navigate to **System** > **Settings**, in the **Modes and Features** group, click **Configure Advanced Features**.
2. Select or clear the **IPv6 Protocol Translation** option.

VLAN Support

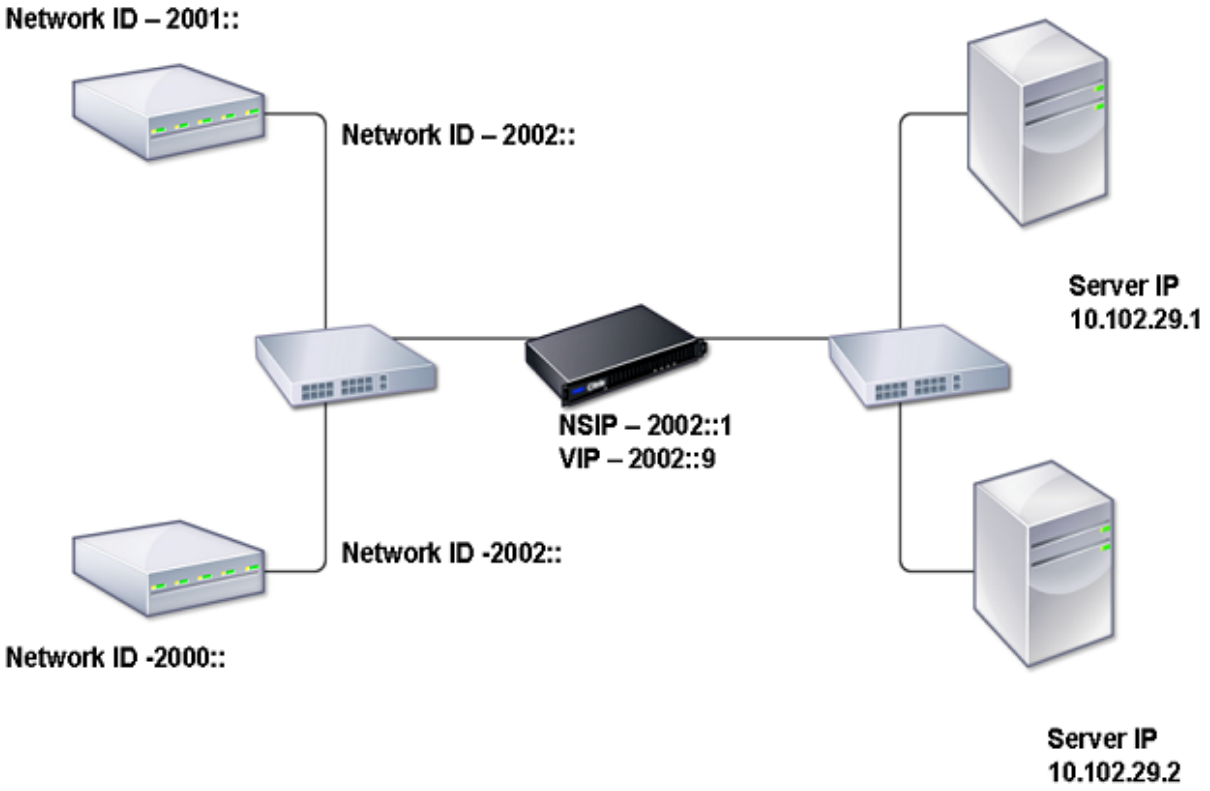
If you need to send broadcast or multicast packets without identifying the VLAN (for example, during DAD for NSIP, or ND6 for the next hop of the route), you can configure the Citrix ADC appliance to send the packet on all the interfaces with appropriate tagging. The VLAN is identified by ND6, and a data packet is sent only on the VLAN. For more information about ND6 and VLANs, see Configuring Neighbor Discovery.

Port-based VLANs are common for IPv4 and IPv6. Prefix-based VLANs are supported for IPv6.

Simple Deployment Scenario

Following is an example of a simple load balancing set-up consisting of an IPv6 vserver and IPv4 services, as illustrated in the following topology diagram.

Figure 1. IPv6 Sample Topology



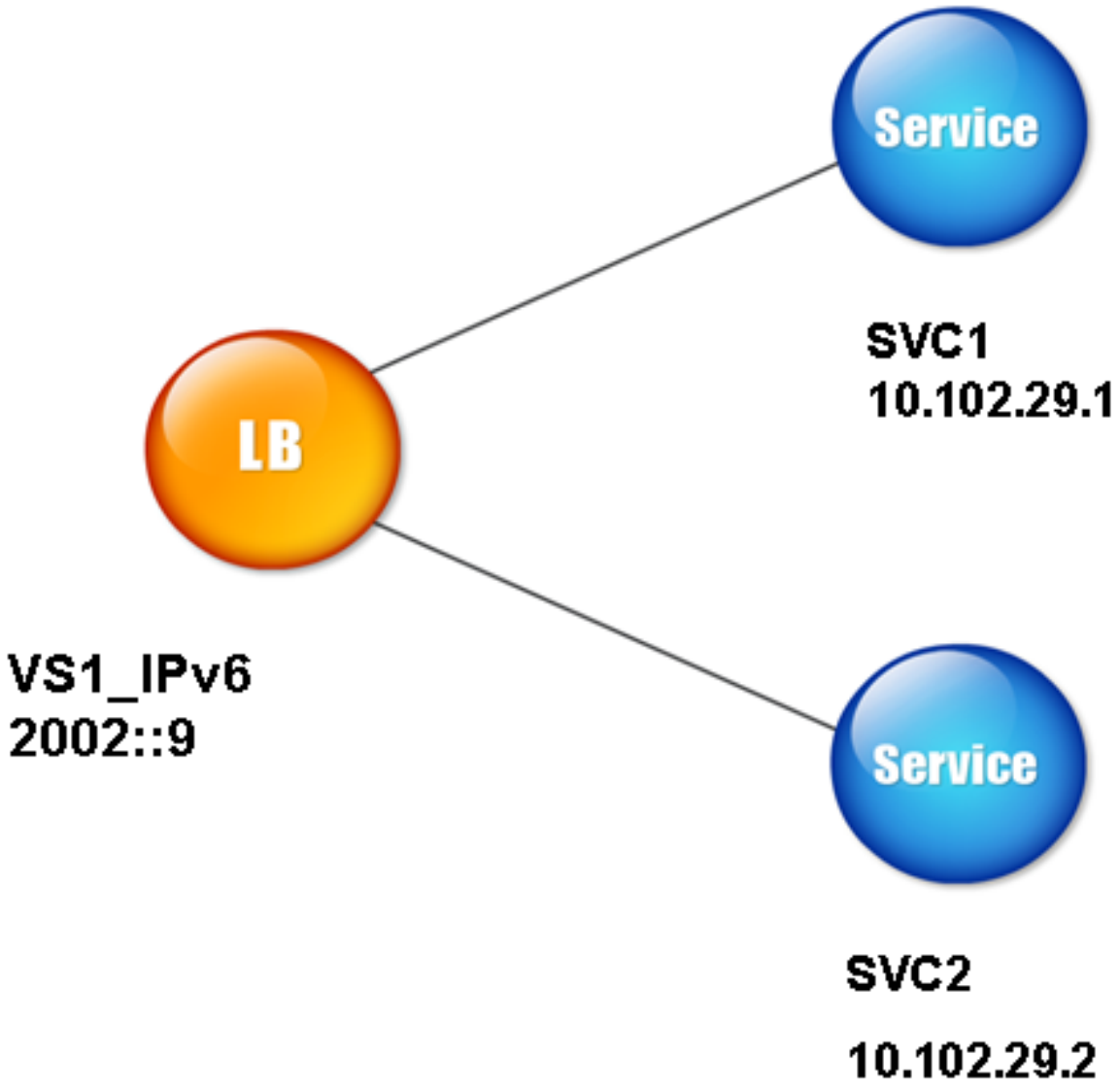
The following table summarizes the names and values of the entities that must be configured on the Citrix ADC.

Table 2. Sample Values for Creating Entities

Entity type	Name	Value
LB Vserver	VS1_IPv6	2002::9
Services	SVC1	10.102.29.1
	SVC2	10.102.29.2

The following figure shows the entities and values of the parameters to be configured on the Citrix ADC.

Figure 2. IPv6 Entity Diagram



To configure this deployment scenario, you need to do the following:

1. Create an IPv6 service.
2. Create an IPv6 LB vserver.
3. Bind the services to the vserver.

CLI procedures

To create IPv4 services by using the CLI:

At the command prompt, type:

- **add service** <Name> <IPAddress> <Protocol> <Port>
- **sh service** <Name>

Example:

```
> add service SVC1 10.102.29.1 HTTP 80
Done

>add service SVC2 10.102.29.2 HTTP 80
Done
<!--NeedCopy-->
```

To create IPv6 vserver by using the CLI:

At the command prompt, type:

- **add lb vserver** <Name> <IPAddress> <Protocol> <Port>
- **sh lb vserver** <Name>

Example:

```
> add lb vserver VS1_IPv6 2002::9 HTTP 80
Done
<!--NeedCopy-->
```

To bind a service to an LB vserver by using the CLI:

At the command prompt, type:

- **bind lb vserver** <name> <service>
- **sh lb vserver** <name>

Example:

```
> bind lb vserver VS1_IPv6 SVC1
Done
<!--NeedCopy-->
```

GUI procedures

To create IPv4 services by using the GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Services**, click **Add**, and then set the following parameters:

- Service Name
- IP Address
- Protocol

- Port

To create IPv6 vserver by using the GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, click **Add**, and select the **IPv6** check box.
2. Set the following parameters:
   - Name
   - Protocol
   - IP Address Type
   - IP Address
   - Port

To bind a service to an LB vserver by using the GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** page, select the vserver for which you want to bind the service (for example, VS1_IPv6).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box corresponding to the service that you want to bind to the vserver (for example, SVC1).
5. Click **OK**.
6. Repeat steps 1-4 to bind the service (for example, SVC2 to the vserver).

Host Header Modification

When an HTTP request has an IPv6 address in the host header, and the server does not understand the IPv6 address, you must map the IPv6 address to an IPv4 address. The IPv4 address is then used in the host header of the HTTP request sent to the vserver.

CLI procedures

To change the IPv6 address in the host header to an IPv4 address by using the CLI:

At the command prompt, type:

- **set ns ip6** <IPv6Address> -**map** <IPAddress>
- **sh ns ip6** <IPv6Address>

Example:

```
> set ns ip6 2002::9 -map 200.200.200.200
Done
<!--NeedCopy-->
```


GUI procedures

To change the IPv6 address in the host header to an IPv4 address by using the GUI:

1. Navigate to **System** > **Network** > **IPs** and, on the **IPV6s** tab, select the IP address for which you want to configure a mapped IP address, for example, 2002:0:0:0:0:0:0:9, and click Edit.
2. In the **Mapped IP** text box, type the mapped IP address that you want to configure, for example, 200.200.200.200.

VIP Insertion

If an IPv6 address is sent to an IPv4-based server, the server may not understand the IP address in the HTTP header, and may generate an error. To avoid this, you can map an IPv4 address to the IPv6 VIP. Then, you can enable VIP insertion to enable insertion of the IPv4 VIP address and port number in the HTTP requests sent to the servers.

CLI procedures

To configure a map IPv6 address by using the CLI:

At the command prompt, type:

set ns ip6 <IPv6Address> -**map** <IPAddress>

Example:

```
> set ns ip6 2002::9 -map 200.200.200.200
 Done
<!--NeedCopy-->
```

To enable VIP insertion by using the CLI:

At the command prompt, type:

- **set lb vserver** <name> -**insertVserverIPPort** <Value>
- **sh lb vserver** <name>

Example:

```
> set lb vserver VS1_IPv6 -insertVserverIPPort ON
 Done

<!--NeedCopy-->
```


GUI procedures

To configure a map IPv6 address by using the GUI:

1. Navigate to **System** > **Network** > **IPs**, on the **IPV6s** tab, select the IP address for which you want to configure a map IP address, for example, 2002:0:0:0:0:0:0:9, and click **Edit**.
2. In the **Mapped IP** text box, type the map IP address that you want to configure, for example, 200.200.200.200.

To enable VIP insertion by using the GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select the virtual server that you want to enable port insertion, and click **Edit**.
2. In the **Advanced** tab, under **Traffic Settings**, in the **Vserver IP Port Insertion** drop-down list box, select **VIPADDR**.
3. In the **Vserver IP Port Insertion** text box, type the vip header.

Traffic Domains

September 14, 2021

> **Warning**
>
> Citrix recommends you to use Admin Partitions instead of using Traffic Domains. For more information, see Admin Partitioning page.

Traffic domains are a way to segment network traffic for different applications. You can use traffic domains to create multiple isolated environments within a Citrix ADC appliance. An application belonging to a specific traffic domain communicates with entities and processes traffic within that domain. The traffic belonging to one traffic domain cannot cross the boundary of another traffic domain.

Benefits of using Traffic Domains

The main benefits of using traffic domains on a Citrix ADC appliance are the following:

- **Use of duplicate IP addresses in a Network**. Traffic domains allow you to use duplicate IP address on the network. You can assign the same IP address or network address to multiple devices on a network, or multiple entities on a Citrix ADC appliance, as long as each of the duplicate address belongs to a different traffic domain.
- **Use of Duplicate entities on the Citrix ADC appliance**. Traffic domains also allow you to use duplicate Citrix ADC feature entities on the appliance. You can create entities with the same settings as long as each entity is assigned to a separate traffic domain.
  Note: Duplicate entities with same name are not supported.

- **Multitenancy**. Using traffic domains, you can provide hosting services for multiple customers by isolating each customer's type of application traffic within a defined address space on the network.

A traffic domain is uniquely identified by an identifier, which is an integer value. Each traffic domain needs a VLAN or a set of VLANs. The isolation functionality of the traffic domain depends on the VLANs bound to the traffic domain. More than one VLAN can be bound to a traffic domain, but the same VLAN cannot be a part of multiple traffic domains. Therefore, the maximum number of traffic domains that can be created depends on the number of VLANS configured on the appliance.

Default Traffic Domain

A Citrix ADC appliance has a preconfigured traffic domain, called the *default traffic domain*, which has an ID of 0. All factory settings and configurations are part of the default traffic domain. You can create other traffic domains and then segment traffic between the default traffic domain and each of the other traffic domains. You cannot remove the default traffic domain from the Citrix ADC appliance. Any feature entity that you create without setting the traffic domain ID is automatically associated with the default traffic domain.

Note: Some features and configurations are supported only in the default traffic domain. They do not work in nondefault traffic domains. For a list of the features supported in all traffic domains, see Supported Citrix ADC Features in Traffic Domains.

How Traffic Domains Work

As an illustration of traffic domains, consider an example in which two traffic domains, with IDs 1 and 2, are configured on Citrix ADC appliance NS1.

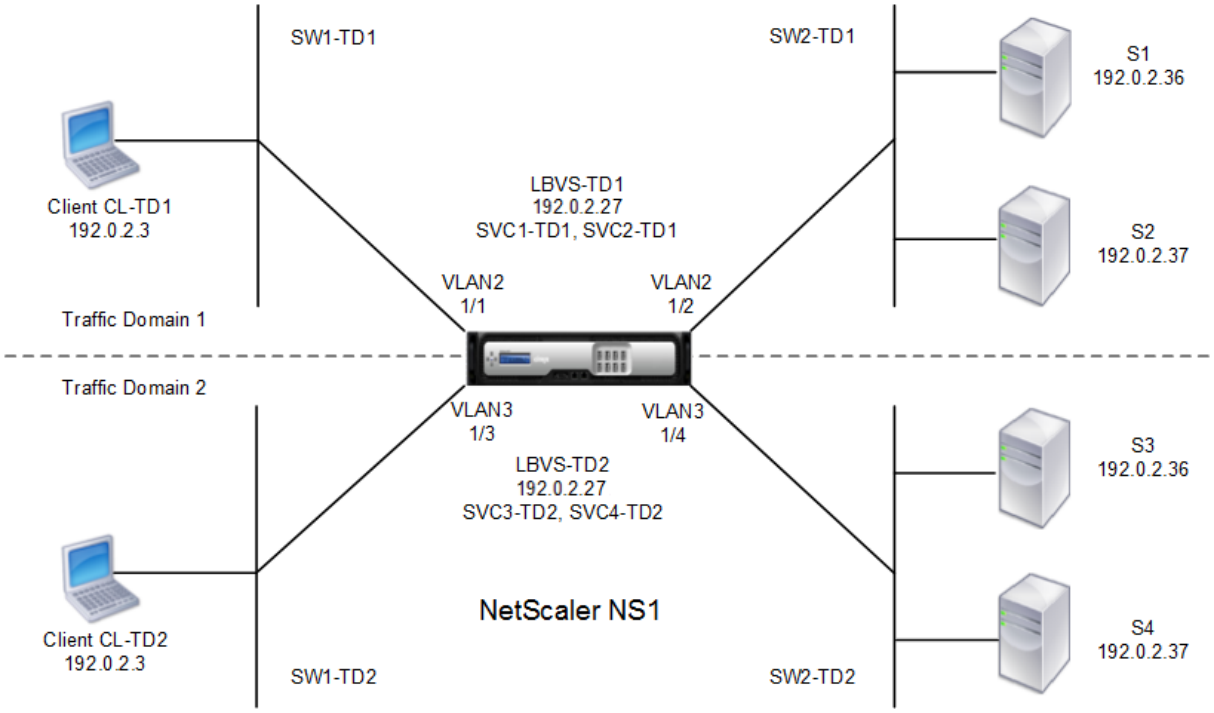
In traffic domain 1, load balancing virtual server LBVS-TD1 is configured to load balance traffic across servers S1 and S2. On the Citrix ADC appliance, servers S1 and S2 are represented by services SVC1-TD1 and SVC2-TD1, respectively. Servers S1 and S2 are connected to NS1 through L2 switch SW2-TD1. Client CL-TD1 is on a private network connected to NS1 through L2 switch SW1-TD1. SW1-TD1 and SW2-TD1 are connected to VLAN 2 of NS1. VLAN 2 is bound to traffic domain 1, which means that client CL-TD1 and servers S1 and S2 are part of traffic domain 1.

Similarly in traffic domain 2, load balancing virtual server LBVS-TD2 is configured to load balance traffic across S3 and S4. On the Citrix ADC appliance, servers S3 and S4 are represented by services SVC3-TD2 and SVC4-TD2, respectively. Servers S3 and S4 are connected to NS1 through L2 switch SW2-TD2. Client CL-TD2 is on a private network connected to NS1 through L2 switch SW1-TD2. SW1-TD2 and SW2-TD2 are connected to VLAN 3 of NS1. VLAN 3 is bound to traffic domain 2, which means that client CL-TD2 and servers S3 and S4 are part of traffic domain 2.

On the Citrix ADC appliance, entities LBVS-TD1 and LBVS-TD2 share the same settings, including the IP address. The same is true for SVC1-TD1 and SVC3-TD2, and for SVC2-TD1 and SVC4-TD2. This is possible because these entities are in different traffic domains.

Similarly, servers S1 and S3, S2 and S4 share the same IP address, and clients CL-TD1 and CL-TD2 each have the same IP address.

Figure 1. How traffic domains work



The following table lists the settings used in the example.

Entity	Name	Details
Settings in traffic domain 1		
VLANs bound to traffic domain 1	VLAN 2	VLAN Id: 2 Interfaces bound: 1/1, 1/2
Client connected to TD1	CL-TD1 (for reference purposes only)	IP address: 192.0.2.3
Load balancing virtual server in TD1	LBVS-TD1	IP address: 192.0.2.27
Service bound to virtual server LBVS-TD1	SVC1-TD1	IP address: 192.0.2.36
Service bound to virtual server LBVS-TD1	SVC2-TD1	IP address: 192.0.2.37

Entity	Name	Details
SNIP	SNIP-TD1 (for reference purposes only)	IP address: 192.0.2.27
Settings in traffic domain 2		
VLAN bound to traffic domain 2	VLAN 3	VLAN Id: 3 Interfaces bound: 1/3, 1/4
Client connected to TD2	CL-TD2 (for reference purposes only)	IP address: 192.0.2.3
Load balancing virtual server in TD2	LBVS-TD2	IP address: 192.0.2.27
Service bound to virtual server LBVS-TD2	SVC3-TD2	IP address: 192.0.2.36
Service bound to virtual server LBVS-TD2	SVC4-TD2	IP address: 192.0.2.37
SNIP in TD2	SNIP-TD2 (for reference purposes only)	IP address: 192.0.2.29

Following is the traffic flow in traffic domain 1:

1. Client CL-TD1 broadcasts an ARP request for the IP address of 192.0.2.27 through L2 switch SW1-TD1.
2. The ARP request reaches NS1 on interface 1/1, which is bound to VLAN 2. Because VLAN 2 is bound to traffic domain 1, NS1 updates the ARP table of traffic domain 1 for the IP address of client CL-TD1.
3. Because the ARP request is received on traffic domain 1, NS1 looks for an entity configured on traffic domain 1 that has an IP address of 192.0.2.27. NS1 finds that a load balancing virtual server LBVS-TD1 is configured on traffic domain 1 and has the IP address 192.0.2.27.
4. NS1 sends an ARP response with the MAC address of interface 1/1.
5. The ARP reply reaches CL-TD1. CL-TD1 updates its ARP table for the IP address of LBVS-TD1 with the MAC address of interface 1/1 of NS1.
6. Client CL-TD1 sends a request to 192.0.2.27. The request is received by LBVS-TD1 on port 1/1 of NS1.
7. LBVS-TD1's load balancing algorithm selects server S2, and NS1 opens a connection between a SNIP in traffic domain 1 (192.0.2.27) and S2.
8. S2 replies to SNIP 192.0.2.27 on NS1.
9. NS1 sends S2's reply to client CL-TD1.

Following is the traffic flow in traffic domain 2:

1. Client CL-TD2 broadcasts an ARP request for the IP address of 192.0.2.27 through L2 switch SW1-TD2.
2. The ARP request reaches NS1 on interface 1/3, which is bound to VLAN 3. Because VLAN 3 is bound to traffic domain 2, NS1 updates traffic-domain 2's ARP-table entry for the IP address of client CL-TD2, even though an ARP entry for the same IP address (CL-TD1) is already present in the ARP table of traffic domain 1.
3. Because the ARP request is received in traffic domain 2, NS1 searches traffic domain 2 for an entity that has an IP address of 192.0.2.27. NS1 finds that load balancing virtual server LBVS-TD2 is configured in traffic domain 2 and has the IP address 192.0.2.27. NS1 ignores LBVS-TD1 in traffic domain 1, even though it has the same IP address as LBVS-TD2.
4. NS1 sends an ARP response with the MAC address of interface 1/3.
5. The ARP reply reaches CL-TD2. CL-TD2 updates its ARP table entry for the IP address of LBVS-TD2 with the MAC address of interface 1/3 of NS1.
6. Client CL-TD2 sends a request to 192.0.2.27. The request is received by LBVS-TD2 on interface 1/3 of NS1.
7. LBVS-TD2's load balancing algorithm selects server S3, and NS1 opens a connection between a SNIP in traffic domain 2 (192.0.2.29) and S3.
8. S2 replies to SNIP 192.0.2.29 on NS1.
9. NS1 sends S2's reply to client CL-TD2.

Supported Citrix ADC Features in Traffic Domains

The Citrix ADC features in the following list are supported in all traffic domains.

> **Important**
>
> Any Citrix ADC feature not listed below is supported only in the default traffic domain.

- ARP table
- ND6 table
- Bridge table
- All types of IPv4 and IPv6 addresses
- IPv4 and IPv6 routes
- ACL and ACL6
- PBR & PBR6
- INAT
- RNAT
- RNAT6
- MSR
- MSR6
- Net profiles

- SNMP MIBs
- Fragmentation
- Monitors (Scriptable monitors are not supported)
- Content Switching
- Cache Redirection
- Persistency (Persistency groups are not supported)
- Service (Domain based services are not supported)
- Service group (Domain based service groups are not supported)
- Policies (*)
- PING
- TRACEROUTE
- PMTU
- High Availability (connection mirroring is not supported)
- Cluster (Supported on L2 clusters. Not supported on L3 clusters)
- Cookie Persistency
- MSS
- Logging (Syslog is not supported)
- Priority Queuing
- Surge Protection
- HTTP DOSP (**)
- Load balancing (The following types are not supported:)
  - TFTP
  - RTSP
  - Diameter
  - SIP
  - SMPP
- NAT46
- NAT64
- DNS64
- Forwarding Session Rules
- SNMP

> **Note**
>
> - * Policies do not have global binding points for traffic domains. However, policies can be bound to a specific load balancing virtual server of a traffic domain.
> - ** HTTP DOSP policies do not have global binding points for traffic domains. However, HTTP DOSP policies can be bound to a specific load balancing service of a traffic domain.
> - Global Server Loading Balancing (GSLB) and ADNS features in Citrix ADC are not aware of

> Traffic Domains. If the GSLB configuration needs to be shared across all traffic domains then GSLB methods Static Proximity and Round Trip Time (RTT) do not work. As a workaround in this scenario you can use GSLB methods other than RTT and Static Proximity. For more information, see http://support.citrix.com/article/CTX202277.

Configuring Traffic Domains

Configuring a traffic domain on the Citrix ADC appliance consists of the following tasks:

- **Add VLANs.** Create VLANs and bind specified interfaces to them.
- **Create a traffic domain entity and bind VLANs to it.** This involves the following two tasks:
  - Create a traffic domain entity uniquely identified by an ID, which is an integer value.
  - Bind the specified VLANs to the traffic domain entity. All the interfaces that are bound to the specified VLANs are associated with the traffic domain. More than one VLAN can be bound to a traffic domain, but a VLAN cannot be a part of multiple traffic domains.
- **Create feature entities on the traffic domain**. Create the required feature entities in the traffic domain. The CLI commands and configuration dialog boxes of all the supported features in a nondefault traffic domain include a parameter called a *traffic domain identifier* (td). When configuring a feature entity, if you want the entity to be associated with a particular traffic domain, you must specify the td. Any feature entity that you create without setting the td is automatically associated with the default traffic domain.

To give you an idea of how feature entities are associated with a traffic domain, this topic covers the procedures for configuring all the entities mentioned in the figure titled How Traffic Domains Work.

The CLI has two commands for these two tasks, but the GUI combines them in a single dialog box.

CLI procedures

To create a VLAN and bind interfaces to it by using the CLI:

At the command prompt, type:

- **add vlan** <id>
- **bind vlan** <id> -**ifnum** <slot/port>
- **show vlan** <id>

To create a traffic domain entity and bind VLANs to it by using the CLI:

At the command prompt, type:

- **add ns trafficdomain** <td>
- **bind ns trafficdomain** <td> -**vlan** <id>
- **show ns trafficdomain** <td>

To create a service by using the CLI:

At the command prompt, type:

- **add service** <name> <IP> <serviceType> <port> -**td** <id>
- **show service** <name>

To create a load balancing virtual server and bind services to it by using the CLI:

At the command prompt, type:

- **add lb vserver** <name> <serviceType> <IPAddress> <port> -**td** <id>
- **bind lb vserver** <name> <serviceName>
- **show lb vserver** <name>

GUI procedures

To create a VLAN by using the GUI:

Navigate to **System** > **Network** > **VLANs**, click **Add**, and set the parameters.

To create a traffic domain entity by using the GUI:

Navigate to **System** > **Network** > **Traffic Domains**, click **Add**, and in the **Create Traffic Domain** dialog box, set the parameters.

To create a service by using the GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Services**, click **Add**, and set the parameters.

To create a load balancing virtual server by using the GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, click **Add**, and set the parameters.

Inter Traffic Domain Entity Bindings

September 14, 2021

You can bind services in one traffic domain to a virtual server in another traffic domain. All the services to be bound to a virtual server in a different traffic domain must reside in the same traffic domain.

You configure this support by using the existing bind lb vserver command or the related GUI procedure.

This capability can facilitate interaction between different traffic domains. In an enterprise, servers can be grouped in different traffic domains. Virtual servers are created in a traffic domain that faces the internet. A virtual server from this traffic domain can be configured to load balance servers in

another traffic domain. This virtual server receives connection requests from the Internet to be forwarded to the bound servers.

When a Citrix ADC is used in a cloud infrastructure, each tenant can be assigned a separate traffic domain, and all the resources (including servers) for a tenant can be grouped together in the tenant's traffic domain. For each tenant, a virtual server is created for load balancing servers in its traffic domain. All of these virtual servers are grouped together in a single traffic domain that faces the Internet.

Consider an example of in which cloud service provider Example-Cloud-A has three traffic domains, with IDs 10, 20, and 30, configured on Citrix ADC appliance NS1.

Example-Org-A and Example-Org-B are tenants of Example-Cloud-A. Tenant A is assigned traffic domain 20, and tenant B is assigned domain 30. Servers S1 and S2 reside in traffic domain 20 and servers S3 and S4 reside in traffic domain 30.

Traffic domain 10 faces the internet. Virtual servers LBVS-1 and LBVS-2 are created in traffic domain 10. LBVS-1, in traffic domain 10, is configured to load balance servers S1 and S2, which are in traffic domain 20. LBVS-2, in traffic domain 10, is configured to load balance servers S3 and S4, which are in traffic domain 30.

Therefore, these virtual servers accept Internet connection requests for servers that are in a different traffic domain than that of the virtual servers.

virtual MAC Based Traffic Domains

September 14, 2021

You can associate a traffic domain with a virtual MAC address instead of with VLANs. The Citrix ADC then sends the traffic domain's virtual MAC address in all responses to ARP queries for network entities in that domain. As a result, the ADC can segregate subsequent incoming traffic for different traffic domains on the basis of the destination MAC address, because the destination MAC address is the virtual MAC address of a traffic domain. After creating entities on a traffic domain, you can easily manage and monitor them by performing traffic domain level operations.

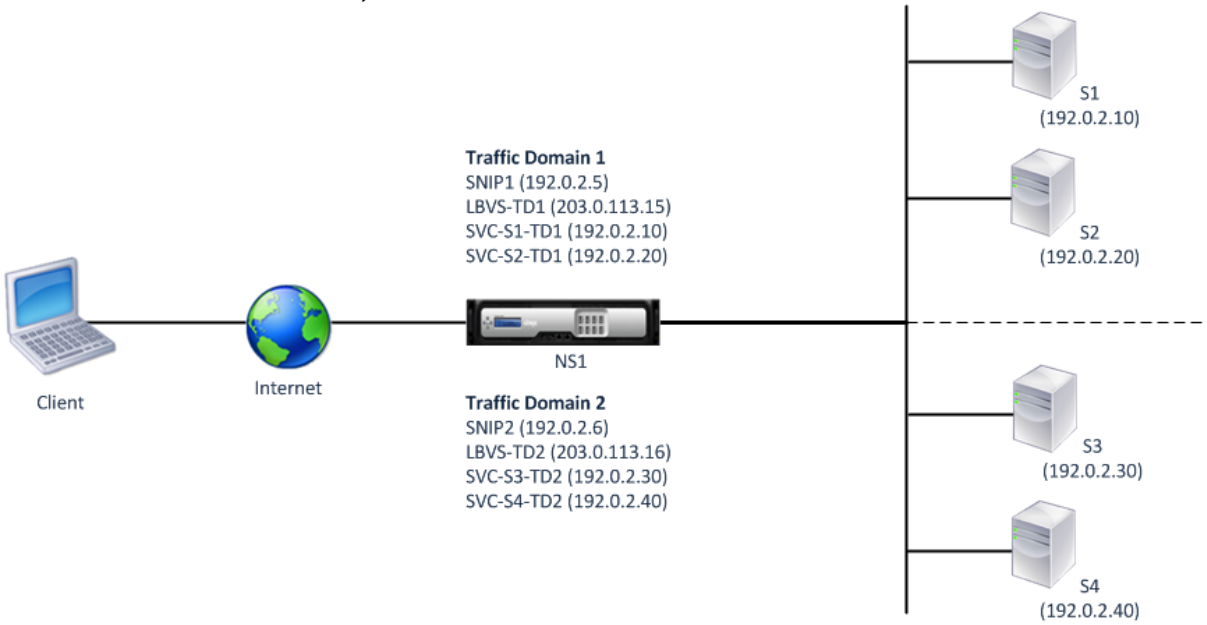
Consider an example in which two traffic domains, with IDs 1 and 2, are configured on Citrix ADC appliance NS1. The Citrix ADC creates a virtual MAC address virtual MAC1 and associates it with traffic domain 1. Similarly, the Citrix ADC created another virtual MAC address virtual MAC2 and associates with traffic domain 2.

In traffic domain 1, load balancing virtual server LBVS-TD1 is configured to load balance traffic across servers S1 and S2. On the Citrix ADC appliance, servers S1 and S2 are represented by services SVC1-TD1 and SVC2-TD1, respectively. A subnet IP address (SNIP) SNIP1 is configured for enabling the Citrix ADC to communicate with S1 and S2. Because virtual MAC1 is associated with traffic domain 1, the

appliance sends virtual MAC1 as the MAC address in all ARP announcements and ARP responses for LBVS-TD1 and SNIP1.

Similarly in traffic domain 2, load balancing virtual server LBVS-TD2 is configured to load balance traffic across S3 and S4. On the Citrix ADC appliance, servers S3 and S4 are represented by services SVC3-TD2 and SVC4-TD2, respectively. A SNIP address SNIP2 is configured for enabling the Citrix ADC to communicate with S3 and S4. Because virtual MAC2 is associated with traffic domain 2, the appliance sends virtual MAC2 as the MAC address in all ARP announcements and ARP responses for LBVS-TD2 and SNIP2.

The Citrix ADC segregate subsequent incoming traffic for traffic domains 1 or 2 on the basis of the destination MAC address, if the destination MAC address is virtual MAC1 or virtual MAC2.



The following table lists the settings used in the example: Virtual MAC based traffic domain example settings.

Before you Begin

Following are points to consider before you configure virtual MAC based traffic domain:

1. virtual MAC based traffic domains are easiest way to achieve network traffic segregation.
2. Because virtual MAC based traffic domains segregate network traffic based on virtual MAC addresses and not VLANS, you cannot create duplicate IP addresses on different virtual MAC based traffic domains on a Citrix ADC.
3. virtual MAC based traffic domains do not work when the Citrix ADC is deployed only in L2 Mode.
4. Both VLAN and virtual MAC based traffic domains can coexist on a Citrix ADC. virtual MAC based traffic domains actually runs on all VLANs that are not bound to any VLAN based traffic domain.

Configuration Steps

Configuring a virtual MAC based traffic domain on a Citrix ADC appliance consists of the following tasks:

- Create a traffic domain entity and enable the virtual MAC option. Create a traffic domain entity uniquely identified by an ID, which is an integer value, and then enable the virtual MAC option. After creating the traffic domain entity, the Citrix ADC creates a virtual MAC address and then associates it to the traffic domain entity.
- Create feature entities on the traffic domain. Create the required feature entities in the traffic domain by specifying the traffic domain identifier (td) when configuring these feature entities. Citrix ADC owned network entities created in a virtual MAC based traffic domain are associated with the virtual MAC address, which is associated with the traffic domain. The Citrix ADC then sends the traffic domain's virtual MAC address in ARP announcements and ARP responses for these network entities.

CLI procedures

To create a virtual MAC based traffic domain by using the CLI:

At the command prompt, type:

```
add ns trafficDomain <td> [-vmac ( ENABLED        DISABLED )]
```

-
- show ns trafficdomain <td>

To configure a SNIP address by using the CLI:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP –td <id>
- show ns ip <IPAddress> -td <id>

To create a service by using the CLI:

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name> -td <id>

To create a load balancing virtual server and bind services to it by using the CLI:

At the command prompt, type:

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>

- bind lb vserver <name> <serviceName>
- show lb vserver <name> -td <id>

Example:

```
> add ns trafficDomain 1 -vmac ENABLED
 Done
> add ns trafficDomain 2 -vmac ENABLED
 Done

> add ns ip 192.0.2.5 255.255.255.0 -type -SNIP -td 1
 Done
> add service  SVC-S1-TD1  192.0.2.10 HTTP 80 -td 1
 Done
> add service  SVC-S2-TD1  192.0.2.20 HTTP 80 -td 1
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.15 80 -td 1
 Done
> bind lb vserver  LBVS-TD1 SVC-S1-TD1
 Done
> bind lb vserver  LBVS-TD1 SVC-S2-TD1
 Done

> add ns ip 192.0.2.6 255.255.255.0 -type -SNIP -td 2
 Done
> add service  SVC-S3-TD2  192.0.2.30 HTTP 80 -td 2
 Done
> add service  SVC-S4-TD2  192.0.2.40 HTTP 80 -td 2
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.16 80 -td 1
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
<!--NeedCopy-->
```

GUI procedures

To create a virtual MAC based traffic domain by using the GUI:

1. Navigate to System > Network > Interfaces.
2. In the details pane, click Add.
3. On the Create Traffic Domain page, set the following parameters:

- Traffic Domain ID*
- Enable Mac

4. Click Create.

To configure a SNIP address by using the GUI:

1. Navigate to System > Network > IPs > IPv4
2. Navigate to Network > IPs > IPv4
3. In the details pane, click Add
4. In the Create IP page, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - IP Address
   - Netmask
   - IP Type
   - Traffic Domain ID
5. Click Create.

To create a service by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Basic Settings Page, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - Service Name
   - Server
   - Protocol
   - Port
   - Traffic Domain ID
4. Click Continue, and click Done.
5. Repeat steps 2-4 to create another service.
6. Click Close.

To create a load balancing virtual server and bind services to it by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the Load Balancing Virtual Servers pane, click Add.
3. In the Create Virtual Servers (Load Balancing) dialog box, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - Name
   - IP Address
   - Protocol
   - Port
   - Traffic Domain ID
4. Click Continue, on the Service Pane, click >.

5. On the Service page, click Insert, and then select the check box for the services that you want to bind to the virtual server.
6. Click Continue, and click Done.
7. Repeat steps 2-5 to create another virtual server

VXLAN

September 14, 2021

Citrix ADC appliances support Virtual eXtensible Local Area Networks (VXLANs). A VXLAN overlays Layer 2 networks onto a layer 3 infrastructure by encapsulating Layer-2 frames in UDP packets. Each overlay network is known as a VXLAN Segment and is identified by a unique 24-bit identifier called the VXLAN Network Identifier (VNI). Only network devices within the same VXLAN can communicate with each other.

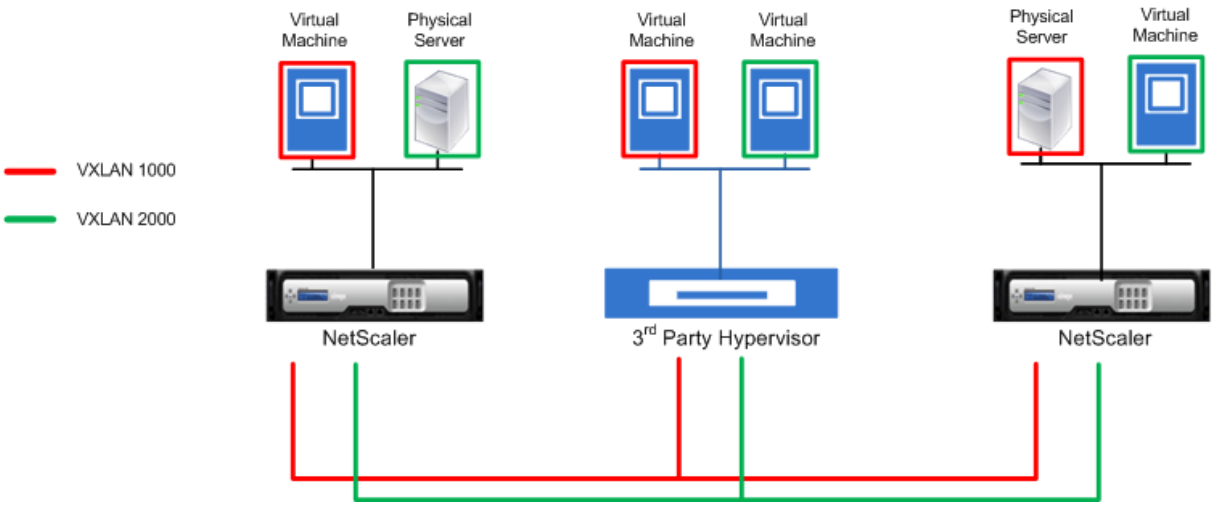
VXLANs provide the same Ethernet Layer 2 network services that VLANs do, but with greater extensibility and flexibility. The two main benefits of using VXLANs are the following:

- **Higher scalability.** Server virtualization and cloud computing architectures have dramatically increased the demand for isolated Layer 2 networks in a datacenter. The VLAN specification uses a 12-bit VLAN ID to identify a Layer 2 network, so you cannot scale beyond 4094 VLANs. That number can be inadequate when the requirement is for thousands of isolated Layer 2 networks. The 24-bit VNI accommodates up to 16 million VXLAN segments in the same administrative domain.
- **Higher flexibility.** Because VXLAN carries Layer 2 data frames over Layer 3 packets, VXLANs extend L2 networks across different parts of a datacenter and across geographically separated datacenters. Applications that are hosted in different parts of a datacenter and in different datacenters but are part of the same VXLAN appear as one contiguous network.

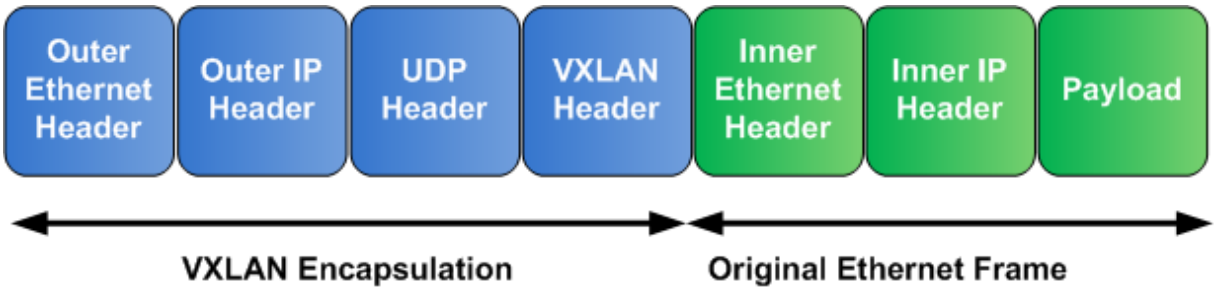
How VXLANs Work

VXLAN Segments are created between VXLAN Tunnel End Points (VTEPs). VTEPs support the VXLAN protocol and perform VXLAN encapsulation and decapsulation. You can think of a VXLAN segment as a tunnel between two VTEPs, where one VTEP encapsulates a Layer2 frame with a UDP header and an IP header and sends it through the tunnel. The other VTEP receives and decapsulates the packet to get the Layer 2 frame. A Citrix ADC is one example of a VTEP. Other examples are third-party hypervisors, VXLAN aware virtual machines, and VXLAN capable switches.

The following illustration displays virtual machines and physical servers connected through VXLAN tunnels.



The following illustration displays the format of a VXLAN packet.



VXLANs on a Citrix ADC use a Layer 2 mechanism for sending broadcast, multicast, and unknown unicast frames. A VXLAN supports the following modes for sending these L2 frames.

- **Unicast mode**: In this mode, you specify the IP addresses of VTEPs while configuring a VXLAN on a Citrix ADC. The Citrix ADC sends broadcast, multicast, and unknown unicast frames over Layer 3 to all VTEPs of this VXLAN.
- **Multicast mode**: In this mode, you specify a multicast group IP address while configuring a VXLAN on a Citrix ADC. Citrix ADCs do not support Internet Group Management Protocol (IGMP) protocol. Citrix ADCs rely on the upstream router to join a multicast group, which shares a common multicast group IP address. The Citrix ADC sends broadcast, multicast, and unknown unicast frames over Layer 3 to the multicast group IP address of this VXLAN.

Similar to a Layer 2 bridge table, Citrix ADCs maintain VXLAN mapping tables based on the inner and outer header of the received VXLAN packets. This table maps of remote host MAC addresses to VTEP IP addresses for a particular VXLAN. The Citrix ADC uses the VXLAN mapping table to look up the destination MAC address of a Layer 2 frame. If an entry for this MAC address is present in the VXLAN table, the Citrix ADC sends the Layer 2 frame over Layer 3, using the VXLAN protocol, to the mapped VTEP IP address specified in the mapping entry for a VXLAN.

Because VXLANs function similarly to VLANs, most of the Citrix ADC features that support VLAN as a classification parameter support VXLAN. These features include an optional VXLAN parameter setting,

which specifies the VXLAN VNI.

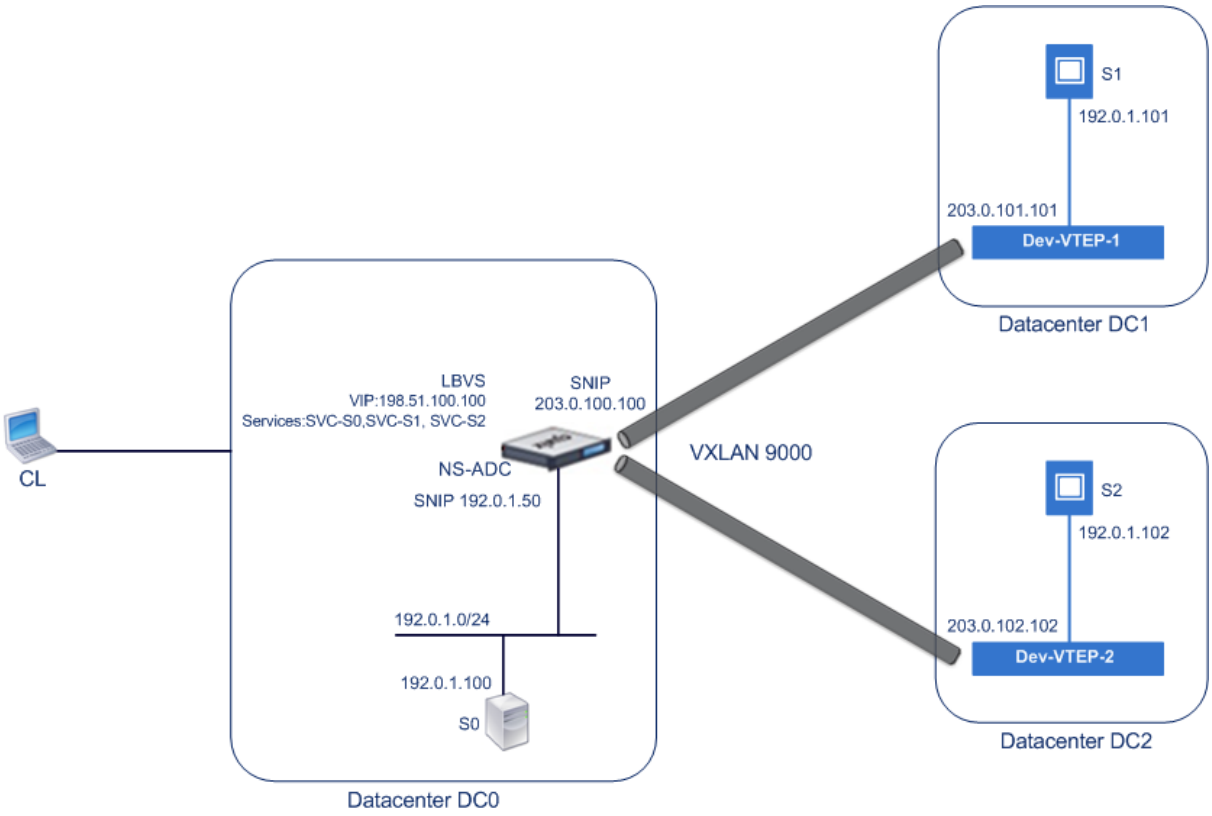
In a high availability (HA) configuration, the VXLAN configuration is propagated or synchronized to the secondary node.

VXLAN Use Case: Load Balancing across Datacenters

To understand the VXLAN functionality of a Citrix ADC, consider an example in which Example Corp hosts a site at www.example.com. To ensure application availability, the site is hosted on three servers, S0, S1, and S2. A load balancing virtual server, LBVS, on Citrix ADC NS-ADC is used to load balance these servers. S0, S1, and S2 reside in datacenters DC0, DC1, and DC2, respectively. In DC0, server S0 is connected to NS-ADC.

S0 is a physical server, and S1 and S2 are virtual machines (VMs). S1 runs on virtualization host device Dev-VTEP-1 in datacenter DC1, and S2 runs on host device Dev-VTEP-2 in DC2. NS-ADC, Dev-VTEP-1, and Dev-VTEP-2 support the VXLAN protocol.

S0, S1, and S2 are part of the same private subnet, 192.0.1.0/24. S0, S1, and S2 be part of a common broadcast domain, VXLAN 9000 is configured on NS-ADC, Dev-VTEP-1, and Dev-VTEP-2. Servers S1 and S2 are made part of VXLAN9000 on Dev-VTEP-1 and Dev-VTEP-2, respectively.



The following table lists the settings used in this example:
VXLAN settings.

Services SVC-S0, SVC-S1, and SVC-S2 on NS-ADC represent S0, S1, and S2. As soon as these services are configured, NS-ADC broadcasts ARP requests for S0, S1, and S2 to resolve IP-to-MAC mapping. These ARP requests are also sent over VXLAN 9000 to Dev-VTEP-1 and Dev-VTEP-2.

Following is the traffic flow for resolving the ARP request for S2:

1. NS-ADC broadcasts an ARP request for S2 to resolve IP-to-MAC mapping. This packet has:
    - Sourced IP address = Subnet IP address SNIP-for-Servers (192.0.1.50)
    - Source MAC address = MAC address of the NS-ADC's interface from which the packet is sent out = NS-MAC-1
2. NS-ADC prepares the ARP packet to be sent over the VXLAN 9000 by encapsulating the packet with following headers:
    - VXLAN header with an ID (VNI) of 9000
    - Standard UDP header, UDP checksum set to 0×0000, and destination port set to 4789.
3. NS-ADC sends the resulting encapsulated packet to Dev-VTEP-1 and Dev-VTEP-2 on VXLAN-9000. The encapsulated packet has:
    - Source IP address = SNIP-VTEP-0 (203.0.100.100).
4. Dev-VTEP-2 receives the UDP packet and decapsulates the UDP header, from which Dev-VTEP-2 learns that the packet is a VXLAN related packet. Dev-VTEP-2 then decapsulates the VXLAN header and learns the VXLAN ID of the packet. The resulting packet is the ARP request packet for S2, which is same as in step 1.
5. From the inner and outer header of the VXLAN packet, Dev-VTEP-2 makes an entry in its VXLAN mapping table that shows the mapping of MAC address (NS-MAC-1) and SNIP-VTEP-0 (203.0.100.100) for VXLAN9000.
6. Dev-VTEP-2 sends the ARP packet to S2. S2's response packet reaches Dev-VTEP-2. Dev-VTEP-2 performs a lookup in its VXLAN mapping table and gets a match for the destination MAC address NS-MAC-1. The Dev-VTEP-2 now knows that NS-MAC-1 is reachable through SNIP-VTEP-0 (203.0.100.100) over VXLAN 9000.
7. S2 responds with its MAC address (MAC-S2). The ARP response packet has:
    - Destination IP address = Subnet IP address SNIP-for-Servers (192.0.1.50)
    - Destination MAC address = NS-MAC-1
8. S2's response packet reaches Dev-VTEP-2. Dev-VTEP-2 performs a lookup in its VXLAN mapping table and gets a match for the destination MAC address NS-MAC-1. The Dev-VTEP-2 now knows that NS-MAC-1 is reachable through SNIP-VTEP-0 (203.0.100.100) over VXLAN 9000. Dev-VTEP-2 encapsulates the ARP response with VXLAN and UDP headers, and sends the resultant packet to SNIP-VTEP-0 (203.0.100.100) of NS-ADC.
9. NS-ADC on receiving the packet, decapsulates the packet by removing the VXLAN and UDP headers. The resultant packet is S2's ARP response. NS-ADC updates it VXLAN mapping table for S2's MAC address (MAC-S2) with Dev-VTEP-2's IP address (203.0.102.102) for VXLAN 9000. NS-ADC also updates it ARP table for S2's IP address (192.0.1.102) with S2's MAC address (MAC-S2).

Following is the traffic flow for load balancing virtual server LBVS in this example:

1. Client CL sends a request packet to LBVS of NS-ADC. The request packet has:
   - Source IP address = IP address of client CL (198.51.100.90)
   - Destination IP address = IP address (VIP) of LBVS = 198.51.110.100
2. LBVS of NS-ADC receives the request packet, and its load balancing algorithm selects server S2 of datacenter DC2.
3. NS-ADC processes the request packet, changing its destination IP address to the IP address of S2 and its source IP address to one of the Subnet IP (SNIP) addresses configured on NS-ADC. The request packet has:
   - Source IP address = Subnet IP address on NS-ADC= SNIP-for-Servers (192.0.1.50)
   - Destination IP address = IP address of S2 (192.0.1.102)
4. NS-ADC finds a VXLAN mapping entry for S2 in its bridge table. This entry indicates that S2 is reachable through Dev-VTEP-2 over VXLAN 9000.
5. NS-ADC prepares the packet to be sent over the VXLAN 9000 by encapsulating the packet with following headers:
   - VXLAN header with an ID (VNI) of 9000
   - Standard UDP header, UDP checksum set to 0×0000, and destination port set to 4789.
6. NS-ADC sends the resulting encapsulated packet to Dev-VTEP-2. The request packet has:
   - Source IP address = SNIP address = SNIP-VTEP-0 (203.0.100.100)
   - Destination IP address = IP address of Dev-VTEP-2 (203.0.102.102)
7. Dev-VTEP-2 receives the UDP packet and decapsulates the UDP header, from which Dev-VTEP-2 learns that the packet is a VXLAN related packet. Dev-VTEP-2 then decapsulates the VXLAN header and learns the VXLAN ID of the packet. The resulting packet is the same packet as in step 3.
8. Dev-VTEP-2 then forwards the packet to S2.
9. S2 processes the request packet and sends the response to the SNIP address of NS-ADC. The response packet has:
   - Source IP address = IP address of S2 (192.0.1.102)
   - Destination IP address = Subnet IP address on NS-ADC= SNIP-for-Servers (192.0.1.50)
10. Dev-VTEP-2 encapsulates the response packet in the same way that NS-ADC encapsulated the request packet in steps 4 and 5. Dev-VTEP-2 then sends the encapsulated UDP packet to SNIP address SNIP-for-Servers (192.0.1.50) of NS-ADC.
11. NS-ADC, upon receiving the encapsulated UDP packet, decapsulates the packet by removing the UDP and VXLAN headers in the same way that Dev-VTEP-2 decapsulated the packet in step 7. The resultant packet is the same response packet as in step 9.
12. NS-ADC then uses the session table for load balancing virtual server LBVS, and forwards the response packet to client CL. The response packet has:
    - Source IP address = IP address of client CL (198.51.100.90)
    - Destination IP address = IP address (VIP) of LBVS (198.51.110.100)

Points to Consider for Configuring VXLANs

Consider the following points before configuring VXLANs on a Citrix ADC:

- A maximum of 2048 VXLANs can be configured on a Citrix ADC.
- VXLANs are not supported in a cluster.
- Link-local IPv6 addresses cannot be configured for each VXLAN.
- Citrix ADCs do not support Internet Group Management Protocol (IGMP) protocol to form a multicast group. Citrix ADCs rely on the IGMP protocol of its upstream router to join a multicast group, which share a common multicast group IP address. You can specify a multicast group IP address while creating VXLAN bridge table entries but the multicast group must be configured on the upstream router. The Citrix ADC sends broadcast, multicast, and unknown unicast frames over Layer 3 to the multicast group IP address of this VXLAN. The upstream router then forwards the packet to all the VTEPs that are part of the multicast group.
- VXLAN encapsulation adds an overhead of 50 bytes to each packet:

  Outer Ethernet Header (14) + UDP header (8) + IP header (20) + VXLAN header (8) = 50 bytes

  To avoid fragmentation and performance degradation, you must adjust the MTU settings of all network devices in a VXLAN pathway, including the VXLAN VTEP devices, to handle the 50 bytes of overhead in the VXLAN packets.

  Important: Jumbo frames are not supported on the Citrix ADC VPX virtual appliances, Citrix ADC SDX appliances, and Citrix ADC MPX 15000/17000 appliances. These appliances support an MTU size of only 1500 bytes and cannot be adjusted to handle the 50 bytes overhead of VXLAN packets. VXLAN traffic might be fragmented or suffer performance degradation, if one of these appliances is in the VXLAN pathway or acts as a VXLAN VTEP device.
- On Citrix ADC SDX appliances, VLAN filtering does not work for VXLAN packets.
- You cannot set a MTU value on a VXLAN.
- You cannot bind interfaces to a VXLAN.

Configuration Steps

Configuring a VXLAN on a Citrix ADC appliance consists of the following tasks.

- **Add a VXLAN entity**. Create a VXLAN entity uniquely identified by a positive integer, which is also called the VXLAN Network Identifier (VNI). In this step, you can also specify the destination UDP port of remote VTEP on which the VXLAN protocol is running. By default, the destination UDP port parameter is set to 4789 for the VXLAN entity. This UDP port setting must match the settings on all remote VTEPs for this VXLAN. You can also bind VLANs to this VXLAN. The traffic (which includes broadcasts, multicasts, unknown unicasts) of all bound VLANs are allowed over

this VXLAN. If no VLANs are bound to the VXLAN, the Citrix ADC allows traffic of all VLANs, on this VXLAN, that are not part of any other VXLANs.

- **Bind the local VTEP IP address and to the VXLAN entity**. Bind one of the configured SNIP address to the VXLAN to source outgoing VXLAN packets.
- **Add a bridgetable entry**. Add a bridgetable entry specifying the VXLAN ID and the remote VTEP IP address for the VXLAN to be created.
- **(Optional) Bind different feature entities to the configured VXLAN**. VXLANs function similarly to VLANs, most of the Citrix ADC features that support VLAN as a classification parameter also support VXLAN. These features include an optional VXLAN parameter setting, which specifies the VXLAN VNI.
- **(Optional) Display the VXLAN mapping table**. Display the VXLAN mapping table, which includes mapping entries for remote host MAC address to VTEP IP address for a particular VXLAN. In other words, a VXLAN mapping states that a host is reachable through the VTEP on a particular VXLAN. The Citrix ADC learns VXLAN mappings and updates its mapping table from the VXLAN packets it receives. The Citrix ADC uses the VXLAN mapping table to lookup for the destination MAC address of a Layer 2 frame. If an entry for this MAC address is present in the VXLAN table, the Citrix ADC sends the Layer 2 frame over Layer 3, using the VXLAN protocol, to the mapped VTEP IP address specified in the mapping entry for a VXLAN.

CLI procedures

To add a VXLAN entity by using CLI:

At the command prompt, type

- **add vxlan** <id>
- **show vxlan**<id>

To bind the local VTEP IP address to the VXLAN by using CLI:

At the command prompt, type

- **bind vxlan** <id> -**SrcIP** <IPaddress>
- **show vxlan** <id>

To add a bridgetable by using CLI:

At the command prompt, type

- **add bridgetable** -**mac** <macaddress> -**vxlan** <ID> -**vtep** <IPaddress>
- **show bridgetable**

To display the VXLAN forwarding table by using the command line:

At the command prompt, type:

- **show bridgetable**

GUI procedures

To add a VXLAN entity and bind a local VTEP IP address by using the GUI:

Navigate to **System** > **Network** > **VXLANs**, and add a new VXLAN entity or modify an existing VXLAN entity.

To add a bridgetable by using the GUI:

Navigate to **System** > **Network** > **Bridge Table**, set the following paramters while adding or modifying a VXLAN bridge table entry:

- MAC
- VTEP
- VXLAN ID

To display the VXLAN forwarding table by using the GUI:

Navigate to **System** > **Network** > **Bridge Table**.

```
Example
> add vxlan 9000
Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
   203.0.101.101

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
   203.0.102.102

Done
```

Support of IPv6 Dynamic Routing Protocols on VXLANs

The Citrix ADC appliance supports IPv6 dynamic routing protocols for VXLANs. You can configure various IPv6 Dynamic Routing protocols (for example, OSPFv3, RIPng, BGP) on VXLANs from the VTYSH command line. An option IPv6 Dynamic Routing Protocol has been added to VXLAN command set for enabling or disabling IPv6 dynamic routing protocols on a VXLAN. After enabling IPv6 dynamic routing protocols on a VXLAN, processes related to the IPv6 dynamic routing protocols are required to be started on the VXLAN by using the VTYSH command line.

To enable IPv6 Dynamic routing protocols on a VXLAN by using the CLI:

- **add vxlan** <ID> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]

- **show vxlan**

```
In the following sample configuration, VXLAN-9000 is created and has
    IPv6 dynamic routing protocols enabled on it. Then, using the VTYSH
    command line, process for the IPv6 OSPF protocol is started on the
    VXLAN.

> add vxlan 9000 -ipv6DynamicRouting ENABLED

Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface VXLAN-9000
NS(config-if)# ipv6 router OSPF area 3
```

Extending VLANs from Multiple Enterprises to a Cloud using VXLAN-VLAN Maps

CloudBridge Connector tunnels is used to extend an enterprise's VLAN to a cloud. VLANs extended from multiple enterprises can have overlapping VLAN IDs. You can isolate each enterprise's VLANs, by mapping them to a unique VXLAN in the cloud. On a Citrix ADC appliance, which is the CloudBridge connector endpoint in the cloud, you can configure a VXLAN-VLAN map that links an enterprise's VLANs to a unique VXLAN in the cloud. VXLANs support VLAN tagging for extending multiple VLANs of an enterprise from CloudBridge Connector to the same VXLAN.

Perform the following tasks for extending VLANs of multiple enterprises to a cloud:

1. Create a VXLAN-VLAN map.
2. Bind the VXLAN-VLAN map to a network bridge based or PBR based CloudBridge Connector tunnel configuration on the Citrix ADC appliance on cloud.
3. (Optional) Enable VLAN tagging in a VXLAN configuration.

CLI procedures

To add a VXLAN-VLAN map by using the CLI:

- **add vxlanVlanMap** <name>

- **show vxlanVlanMap** <name>

To bind a VXLAN and VLANS to a VXLAN-VLAN map by using the CLI:

- **bind vxlanVlanMap** <name> [-**vxlan** <positive_integer> -**vlan** <int[-int]> …]
- **show vxlanVlanMap** <name>

To bind a VXLAN-VLAN map to a network bridge based CloudBridge Connector tunnel by using the CLI:

At the command prompt, type one of the following sets of commands.

if adding a new network bridge:

- **add netbridge** <name> [-**vxlanVlanMap** <string>]
- **show netbridge** <name>

if reconfiguring an existing network bridge:

- **set netbridge** <name> [-**vxlanVlanMap** <string>]
- **show netbridge** <name>

To bind a VXLAN-VLAN map to a PBR based CloudBridge Connector tunnel by using the CLI:

At the command prompt, type one of the following sets of commands.

if adding a new PBR:

- **add pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** <name>])
- **show pbr** <name>

if reconfiguring an existing PBR:

- **set pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** <name>])
- **show pbr** <name>

To include VLAN tags in packets related to a VXLAN by using the CLI:

At the command prompt, type one of the following sets of commands.

if adding a new VXLAN:

- **add vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

if reconfiguring an existing VXLAN:

- **set vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

GUI procedures

To add a VXLAN-VLAN map by using the GUI:

Navigate to **System** > **Network** > **VXLAN VLAN Map**, add a VXLAN VLAN map.

To bind a VXLAN-VLAN map to a netbridge based CloudBridge Connector tunnel by using the GUI:

Navigate to **System** > **CloudBridge Connector** > **Network Bridge**, select a VXLAN-VLAN map from the **VXLAN VLAN** drop down list while adding a new network bridge, or reconfiguring an existing network bridge.

To bind a VXLAN-VLAN map to a PBR based CloudBridge Connector tunnel by by using the GUI:

Navigate to **System** > **Network** > **PBRs**, on the Policy Based Routing (PBRs) tab, select a **VXLAN-VLAN** map from the **VXLAN VLAN** drop down list while adding a new PBR, or reconfiguring an existing PBR.

To include VLAN tags in packets related to a VXLAN by using the GUI:

Navigate to **System** > **Network** > **VXLANs**, enable **Inner VLAN Tagging** while adding a new VXLAN, or reconfiguring an existing VXLAN.

```
1  > add vxlanVlanMap VXLANVLAN-DC1
2
3  Done
4
5  > bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3000 -vlan 3
6
7  Done
8
9  > bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3500 -vlan 4
10
11 Done
12
13 >add vxlanVlanMap VXLANVLAN-DC2
14
15 Done
16
17 > bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 8000 -vlan 3 4
18
19 Done
20
21 > set pbr PBR-CBC-DC-1-CLOUD ALLOW  -ipTunnel  CBC-DC-1-CLOUD -
      vxlanVlanMap VXLANVLAN-DC1
22
23 Done
24
25 > set pbr PBR-CBC-DC-2-CLOUD ALLOW  -ipTunnel  CBC-DC-2-CLOUD -
      vxlanVlanMap VXLANVLAN-DC2
26
27 Done
```

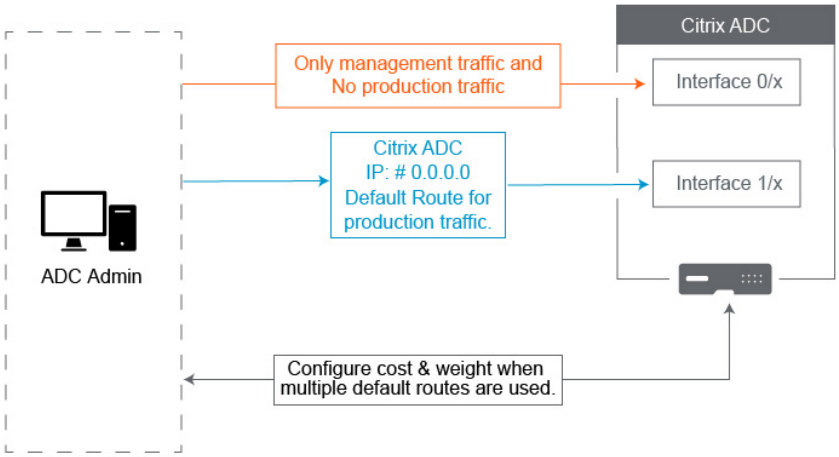

Best practices for network configurations

September 14, 2021

The following sections talk about some best practices for configuring networking features on a Citrix ADC appliance.

Routing and Default Routes

The following are some best practices for configuring Layer 3 features on a Citrix ADC appliance.



- **Interface `0/x` on a Citrix ADC appliance or Citrix SDX appliance must not be used for production traffic**. On an MPX or SDX, Interfaces named `0/x` are referred to the Management Interfaces. This does not mean you must use these interfaces for Management. What it means is that these interfaces are NOT designed for Production traffic. They do not have the hardware buffers and optimization necessary to achieve sustained 1 Gbps throughput. Therefore, If your Default Route is in the same Subnet as your NSIP, you must either change the default route or use a `1/x` interface for your Management network as the `1/x` interfaces are fully optimized for production 1 Gbps traffic.

  > **Note**:
  >
  > This does not apply to a Citrix ADC VPX appliance.

  - **Option 1**. Do not connect to Interfaces `0/x` – Disconnect the cable from interface `0/1`. NetScaler listens for the NSIP on the other interfaces. (NOTE: This is not an option for SDX, as the SVM and XenServer can only speak to `0/x` interfaces)
  - **Option 2**. change the default route to a different interface as detailed in the next section.

- **Default gateway (route 0.0.0.0) should be on a Production network, and not on any `0/x` interface**. When first setting up a NetScaler, it asks you for the NSIP, Subnet Mask, and Gateway address. The problem this creates for Administrators is that they just configured their default route to be on their Management Network using Interface 0/1.
  - To check what your routes are, run in CLI `show route` and your default gateway is the IP in the line where the Network and Netmask are 0.0.0.0. Here's an example where the Gateway is on Line 1:

```
1  > sh route
2          Network          Netmask          Gateway/OwnedIP
               State   Traffic Domain  Type
3          -------          -------          ---------------
               -----   --------------  ----
4  1)      0.0.0.0          0.0.0.0          10.25.213.65     UP
            0               STATIC
5  2)      127.0.0.0        255.0.0.0        127.0.0.1        UP
            0               PERMANENT
6  3)      10.25.213.64     255.255.255.192  10.25.213.68     UP
            0               DIRECT
7  4)      172.16.0.0       255.255.255.0    172.16.0.1       UP
            0               DIRECT
8
9  <!--NeedCopy-->
```

  - To check the Interface and VLAN used for your Default Gateway, check the ARP table using `sh arp` in CLI. You can also search for the specific IP using `show arp | grep 10.25.213.65`. Here's an example where you see the Gateway 10.25.213.65 is using Interface 1/1 and VLAN 1:

```
1  > sh arp
2          IP               MAC                Iface VLAN
               Origin     TTL     Traffic Domain
3          --               ---                ----- ----
               ------     ---     --------------
4  1)      127.0.0.1        02:00:18:a4:00:1e  LO/1  1
      PERMANENT  N/A    0
5  2)      10.25.213.70     02:00:0f:46:00:28  1/1   1
      DYNAMIC    967    0
6  3)      10.25.213.68     02:00:18:a4:00:1e  LO/1  1
      PERMANENT  N/A    0
7  4)      10.25.213.67     02:00:0f:46:00:28  1/1   1
      DYNAMIC    641    0
8  5)      10.25.213.65     00:08:e3:ff:fd:90  1/1   1
      DYNAMIC    483    0
```



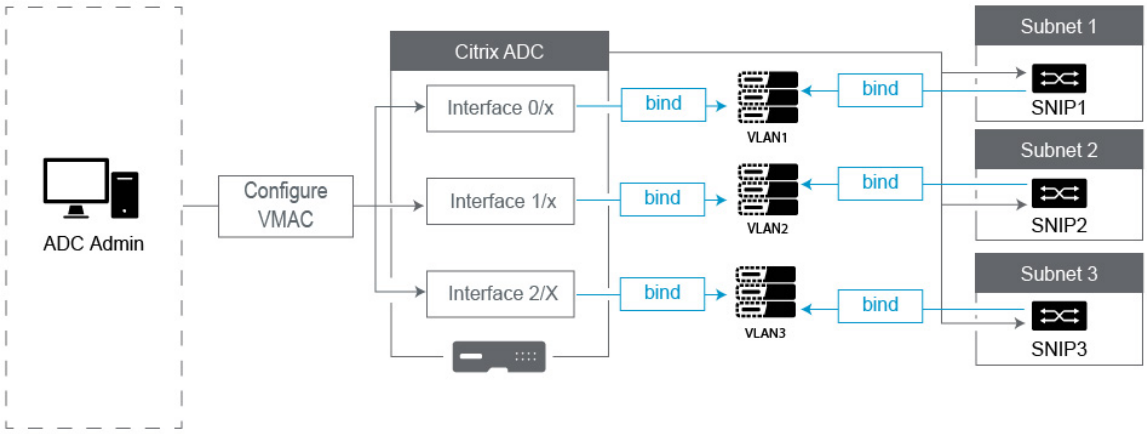
```
<!--NeedCopy-->
```

- Change the default route to use a Gateway on your Production Subnet and Interface. Assume your Management network is 10.0.0.0/24 with gateway 10.0.0.1, and Production network is 10.1.1.0/24 with gateway 10.1.1.1. Set up your config like this:
  - SNIP: (Management Access Disabled) 10.1.1.2
  - NSIP: (Management Access Enabled) 10.0.0.2
  - Default Route: 0.0.0.0 0.0.0.0 10.1.1.1 (System > Network > Routes). This uses a router on the SNIP network instead of the NSIP network.

    > **Note**:
    >
    > Changing the default gateway might break management traffic unless you configure static routes, a Policy Based Route, or enable MAC Based Forwarding.

Interfaces, channels, and VLANs

The following are some best practices for configuring Layer 2 features on a Citrix ADC appliance.



- **Do not connect multiple interfaces/channels to the same VLAN, including VLAN 1**:
  - If you do not properly configure your VLANs, it can cause some unexpected packet routing in your network and Layer 2 looping anytime there are more than one Active interface with the same VLAN (either Native or Tagged).
  - By Default, all Interfaces and Channels are on Native VLAN 1. This creates two possible problems:

        * The NetScaler thinks all traffic received is on the same network, so it uses any interface to send the traffic out on. If you have a different Native VLAN on the interface it sent data out on, then the traffic will not be routed as you expect.
        * If the NetScaler receives broadcast packets on one port, it may retransmit on another port. If both switchports are on the same VLAN, you just created a Layer 2 loop.
    - To remove an interface/channel from VLAN 1:
        * If you are not using Native VLANs on your switch interface/port channel. Change the Native VLAN on the NetScaler Interface/Channel to an unused VLAN number such as 999. You should not use the same unused VLAN number for multiple Channels or Interfaces, as it creates a layer 2 loop.
        * If you are using Native VLANs on your switch interface/port channel. Change the Native VLAN on the NetScaler Interface/Channel to match. However take care not to have multiple active Interfaces or Channels on the same VLAN as doing so creates Layer 2 loops.
        * You cannot remove the Native VLAN. Instead, you can change it or set TagAll for the interface or channel. If the switch port isn't configured with an untagged native VLAN, then enable tagall on the interface so High Availability heartbeat packets will be tagged.
    - To view the Native VLAN on an interface, run `sh interface` in CLI. This will also inform you if the interface is using the TAGALL option.

- **Bind an Interface to your VLAN** - The NetScaler, by default, does not attach a new VLAN to an interface. This means the VLAN is not going to be used until you bind it to an interface. When the new VLAN is not bound to an Interface and that VLAN is Tagged, the NetScaler drops all inbound traffic from that VLAN. Also, do not bind the same VLAN to more than one interface.
    - Bind Subnets to your VLANs. The NetScaler does not work like a typical Router. Most Routers attach IPs to Interfaces. On a NetScaler, the IPs float on any Interface unless configured otherwise. Therefore, any Subnet you want to ensure the NetScaler sends over a specific VLAN, especially when the NetScaler is Initiating that traffic, then you must bind a SNIP within that subnet to the VLAN.
    - A common argument we hear against this is that it used to work fine and now it doesn't without binding the Subnet to the VLAN. This often occurs because the NetScaler learns which VLAN to send traffic out, but this can take time as it builds its ARP tables. After a reboot or firmware upgrade, as it starts building the ARP tables again, it may initially learn and therefore be using a different path than you desire, such as your default route. It's best to instruct it which path to take by Binding the SNIP to the VLAN. Once a SNIP is bound to a VLAN, the entire subnet for that SNIP will be bound to the VLAN.

  - Ensure every SNIP is bound to a VLAN (except in cases where you have more than 1 SNIP in a subnet, then you only must bind one), and that the VLAN, in turn, is bound to only one Interface or channel. It is also often best to have a SNIP in every Subnet, but that is not required as the most specific Route will be used for any destination Subnet that does not have a SNIP.

- To identify the VLAN and Interface used by a Subnet:

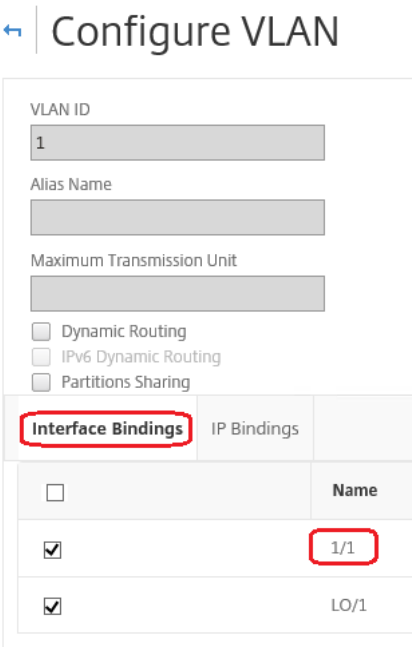
  1. Go to **System>Network > VLANs**.
  2. Edit each VLAN configured, in turn, until you find the correct IP address as explained in the next step.
  3. Click the IP Bindings tab to see which IP, and thus which Subnet is bound and thus is using this VLAN.
  4. Once you identify the VLAN that has an IP bound to it, where that IP is within the subnet of the Default Route, then click the Interface Bindings. Each Interface or Channel bound to this VLAN will be used.

Example

Assume the Default route is `0.0.0.0 0.0.0.0 10.1.1.1`.

Assume you have two SNIPs of 10.0.0.5 and 10.1.1.69. Since 10.1.1.69 is in the subnet of the default route, that is the one you want to look for. In the below screenshots, we are reviewing VLAN 1 and we see the IP 10.1.1.69 is bound to this VLAN, so we know we are looking at the correct VLAN.

Now Click Interface Bindings. In the VLAN Interface bindings we see that Interface `1/1` is used for this subnet, and therefore is used for the default route.



> **NOTE**:
>
> If you do not have any IPs bound to your VLANs, then by default it will be sent out VLAN 1, so in that case look at which Interfaces are bound to VLAN 1. This also means the NetScaler will not use your Configured VLANs for traffic it Initiates unless you bind an IP to the new VLAN.

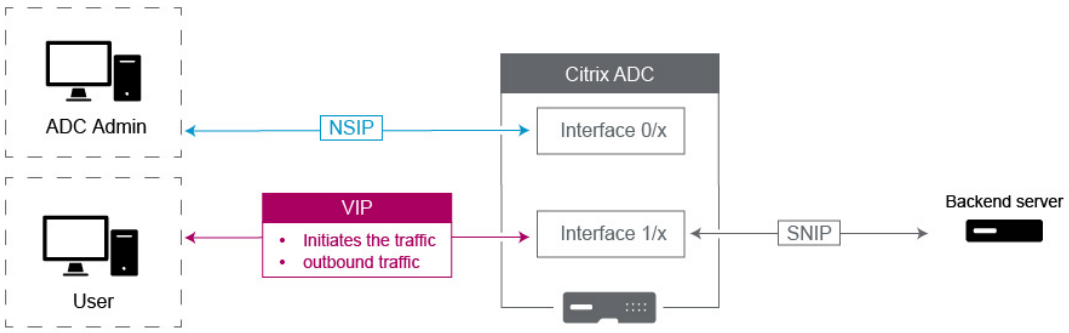
Gratuitous ARP

If GARP does not work, use VMAC - By default, the NetScaler uses GARP to advertise its IP to MAC address bindings to other network devices. This typically works without issue, however, as you create more services in the NetScaler, you may begin to experience issues when failing over on an HA pair. The most common issue is that services remain down in the NetScaler you failed over to due to some network devices not having updated their ARP tables with the new MAC address. You can easily verify this by checking their ARP tables to see if the MAC addresses match those on the now-Primary NetScaler. When this occurs, it is highly likely that some of your network devices are limiting the number of GARP advertisements they honor. In this case it is necessary to configure VMAC on all your active interfaces and/or channels. If you expect to have a large configuration on your NetScaler, it may be best to configure VMAC for all interfaces and channels during the initial deployment.

> **NOTE**:
>
> Do not forget to configure VMAC for the Interface or Channel used by your Default Route.

Citrix ADC owned IP Addresses

This section talks about the best practices for configuring Citrix ADC owned IP addresses:



- **Citrix ADC IP (NSIP)**: Generally this IP used for Management because it is the only IP unique to an individual NetScaler in an HA or Cluster environment. Also important to note is that LDAP, RADIUS, and User scripted Monitor traffic (such as the LDAP monitor and StoreFront monitor) will Source from the NSIP and thus route over the VLAN and Interface the NSIP is bound to (Default Native VLAN 1). If you need the LDAP and RADIUS traffic to source from the SNIP, then create an LB virtual server for your back end servers.
- **Subnet IP (SNIP)**: This IP address is used to initiate communication to back-end servers and is always going to initiate traffic. That said, it can be the destination for traffic in these cases:
  - It can be used as the Gateway address on other devices when doing Layer 3 routing on the NetScaler.
  - It can, when enabled, accept Management services, such as access to the GUI, SSH, and SNMP.
- **Virtual IP (VIP)**: The VIP is unique in that it will never be used to initiate outbound traffic. It is intended to receive Traffic only. Once it receives traffic, it replies and sends traffic outbound back to the client. In other words, the VIP address does not initiate the outbound traffic.

Note this also means it is notused as the source for communicating with back-end servers used in, for example, an LB virtual server.

Configure to source Citrix ADC FreeBSD data traffic from a SNIP address

September 14, 2021

Some Citrix ADC data features run on the underlying FreeBSD OS instead of on the Citrix ADC OS. Because of this reason, these features send traffic sourced from the Citrix ADC IP (NSIP) address instead

of sourced from a SNIP address. Sourcing the data traffic from the NSIP address is not desirable if your setup has configurations to separate all management and data traffic.

The following Citrix ADC data features run on the underlying FreeBSD OS and send traffic sourced from the Citrix ADC IP (NSIP) address:

- Load balancing scriptable monitors
- GSLB autosync

To resolve this issue, you can use the global Layer-2 parameter: `useNetprofileBSDtraffic`. When you enable this parameter, the Citrix ADC features send traffic sourced from one of the SNIP addresses in a net profile associated with the feature.

Before you begin

Before configuring the Citrix ADC appliance to source Citrix ADC features related traffic from a SNIP address, note the following points:

- Currently, the global Layer-2 parameter `useNetprofileBSDtraffic` is supported only for load balancing scriptable monitors.

  For configuring the Citrix ADC appliance to source GSLB autosync traffic from a SNIP address, you can use extended ACL rules and RNAT rules as a workaround.

- The `useNetprofileBSDtraffic` support for load balancing scriptable monitors is applicable only for net profiles bound to the related services. The `useNetprofileBSDtraffic` support is not applicable for net profiles bound to the related service groups.

  In other words, the Citrix ADC appliance does not use any SNIP address from the net profiles bound to the service groups for sourcing load balancing scriptable monitors traffic.

- The `useNetprofileBSDtraffic` support is not applicable for SSL services.

  In other words, the Citrix ADC appliance does not use any SNIP address from the net profiles bound to the SSL services for sourcing load balancing scriptable monitors traffic.

Configure the Citrix ADC appliance to source scriptable monitors traffic from a SNIP address

Configuring the Citrix ADC appliance to source scriptable monitors traffic from a SNIP address consists of the following tasks:

- Enable the global Layer-2 parameter `useNetprofileBSDtraffic`.
- Create a net profile and bind at least one SNIP address to it.
- Bind the net profile to the load balancing services that are using scriptable monitors.

To enable the Layer-2 parameter useNetprofileBSDtraffic by using the CLI:

At the command prompt, type:

- **set l2param** -**useNetprofileBSDtraffic** (**ENABLED** / **DISABLED**)
- **show l2param**

To Create a net profile and bind SNIP addresses to it by using the CLI:

At the command prompt, type:

- **add netProfile** <name> -**srcIP** <string>
- **show netProfile**

To bind a net profile to a load balancing service by using the CLI:

At the command prompt, type:

- **set service** <name> -**netProfile** <string>
- **show service** <name>

Sample configuration

The following sample configuration enables a Citrix ADC appliance to source scriptable monitors traffic from a SNIP address. A net profile NETPROFILE-1 is configured with SNIP address 198.51.100.20 bound to it. A user/scriptable monitor USER-MONITOR-1 is created and is bound to a load balancing service SERVICE-1. NETPROFILE-1 is bound to SERVICE-1. The Citrix ADC appliance sources all scriptable monitors packets of USER-MONITOR-1 from SNIP address 198.51.100.20.

```
1  set l2param -useNetprofileBSDtraffic ENABLED
2
3  set netprofile NETPROFILE-1 -srcip 198.51.100.20
4
5  add lb monitor USER-MONITOR-1 USER -scriptName nsftp.pl -scriptArgs "
       file=Index.png;user=nsroot;password=nsroot" -dispatcherIP 127.0.0.1
       -dispatcherPort 3013 -destIP 203.0.113.90 -destPort 21
6
7  bind service SERVICE-1 -monitorName USER-MONITOR-1
8
9  set service SERVICE-1 -netProfile NETPROFILE-1
10
11 <!--NeedCopy-->
```


Configure the Citrix ADC appliance to source GSLB autosync traffic from a SNIP address

Configuring the Citrix ADC appliance to source GSLB autosync traffic from a SNIP address consists of the following workaround tasks:

- **Create an extended ACL rule**. An extended ACL rule identifies the GSLB autosync packets. This identification is based on the source IP and destination IP addresses.
- **Apply ACLs**. Applying ACLs activates the newly created ACL rule.
- **Create an ACL based RNAT rule**. An RNAT rule changes the source IP address of these packets from the NSIP address to a SNIP address.

> **Note:**
>
> In a high availability or cluster setup, you must add ACL and RNAT rules for all the NSIP addresses of the setup.

To create an extended ACL by using the CLI:

At the command prompt, type:

- **add acl** <aclname> **ALLOW** -**srcIP** = <NSIP address> -**destIP** = <destination IP address of the packets>
- **show acl** <aclName>

To apply extended ACLs by using the CLI:

At the command prompt, type:

- **apply acls**

To create an ACL based RNAT rule by using the CLI:

At the command prompt, type:

- **add rnat** <name> <aclname>
- **bind rnat** <name> -**natIP** <SNIP address - source IP address for the packets>
- **show rnat** <name>

Sample configuration

The following sample configuration enables a Citrix ADC appliance to source GSLB autosync traffic from a SNIP address. ACL-2 identifies GSLB autosync packets, which are sourced from NSIP address 192.0.1.20 and destined to GSLB site IP address 203.0.113.20. RNAT-2 changes the source IP address to SNIP address 198.51.100.20 for these identified packets.

```
add acl ACL-2 ALLOW -srcIP = 192.0.1.20  -destIP = 203.0.113.20

apply acls
```



```
4
5 add rnat RNAT-2 ACL-2
6
7 bind rnat RNAT-2 -natIP 198.51.100.20
8 <!--NeedCopy-->
```

Priority Load Balancing

September 14, 2021

The priority load balancing feature enables you to assign a priority number for each of the services or service groups that are bound to a priority load balancing virtual server. A service or a service group with the lowest number has the highest priority. The application traffic is distributed only to this service or a service group as long as this service or the service group is UP. The service or the service group that is assigned the next priority number becomes operational only when all the services or members in the service group with the highest priority are DOWN. However, when any of the services or a member in the service group with the highest priority becomes available again, the traffic is redirected to that service or the service group.

For example, consider the service groups SVG1, SVG2, and SVG3 that is bound to a priority load balancing virtual server. The maximum number of priority groups is set to three. You assign the priority for each group as follows:

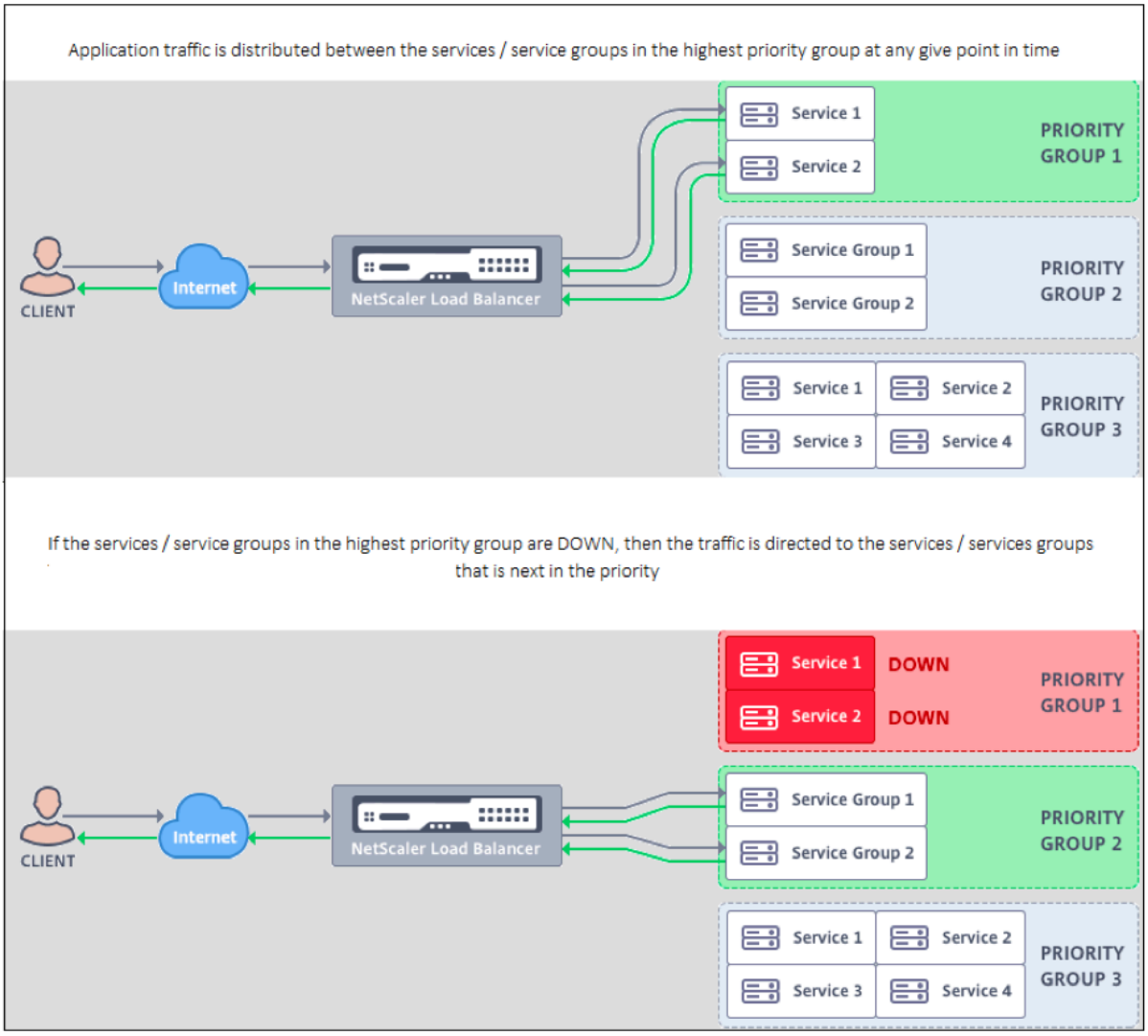
- SVG1 – priority 1
- SVG2 – priority 2
- SVG3 – priority 3

In this scenario, the application traffic is directed to service group SVG1 because this group is assigned the lowest priority number. If all the members in SVG1 are DOWN, the traffic is distributed to service group SVG2 as this group is assigned the next lower priority number. If all the members in SVG2 are also DOWN, the traffic is distributed to SVG3. However, when any of the members in SVG1 is UP, the traffic is redirected to SVG1 because SVG1 is assigned the lowest number and has the highest priority.

You can assign a priority to a service or a service group to upgrade the specific service or service group that has the highest priority, whenever required with minimal or no impact to the production traffic.

Also, if the upgrade is not successful, you can safely switch to the service or the service group that is next in the priority, with minimal or no impact to the production traffic.

The following figure illustrates the priority load balancing feature.



Configure priority load balancing

> **Note**
>
> The Citrix ADC priority load balancing configuration is supported only through the GUI. You cannot configure priority load balancing by using the CLI.

1. Navigate to **Traffic Management** > **Priority Load Balancing**> **Virtual *Servers** and specify the protocol for the virtual server, the IP address, and the port number of the virtual server.
2. In the **Maximum Priority Groups** box, enter the number of priority services or the service groups that can be bound to this virtual server. Default value is 2, and the maximum priority that can be set is 10. This parameter is noneditable after it is configured.

   Note:

   After you specify the maximum number of priority groups and click **OK**, a content switching vir-

tual server and “n” number of backup load balancing virtual servers are created. The alphabet “n” represents the maximum number of priority groups.

For example, if you have entered the virtual server name as vs1, and have set the maximum priority group as 5, then a content switching virtual server with the name `_Pri.LB##vs1##MaxPri=5` and the following 5 load balancing virtual servers are created.

- `_Pri.LB##vs1##MaxPri=5_LB1`
- `_Pri.LB##vs1##MaxPri=5_LB2`
- `_Pri.LB##vs1##MaxPri=5_LB3`
- `_Pri.LB##vs1##MaxPri=5_LB4`
- `_Pri.LB##vs1##MaxPri=5_LB5`

3. After you specify the maximum number of priority groups and click **OK**, you are prompted to choose the services or service groups that must be bound to this content switching virtual server.

   - To bind services to the virtual server, click **Insert** in the Services section. Next, either select an existing service or create a service and set the priority for this service. Also, set the priority number at which this service must be bound.
   - To bind service groups to the virtual server, click **Insert** in the Service Groups section. Next, either select an existing service group or create a service group and set the priority for this service group. Also, set the priority number at which this service group must be bound.

   Repeat step 3, depending on the maximum number of priority groups that you have entered.

   Note:

   - The service or the service group with the highest priority is bound to the load balancing virtual server that represents the highest priority.

   For example, if you have assigned priority 1 and 2 to service groups `SG_App1 and SG_App2` respectively, then `SG_App1` is bound to `virtual server _Pri.LB##vs1##MaxPri=5_LB1 and SG_App2` is bound to `virtual server _Pri.LB##vs1##MaxPri=5_LB2` created in step 2.

   - To change the priority of the service group or service, click the edit icon in the Priority Load Balancing Virtual Server page and change the priority as required.
   - You cannot set the load balancing methods and persistence for each virtual server explicitly, because the configuration of all load balancing virtual servers is identical.

4. From the Advanced Setting sections, complete the other configuration that meets your requirement.

> **Important:**
>
> The entities created during the priority load balancing configuration must not be modified from

> other tabs in the GUI and also from the CLI. It is recommended that you modify the priority load balancing entities from the Priority Load Balancing tab only.

Citrix ADC Extensions

September 14, 2021

Citrix ADC extensions can be used to customize a Citrix ADC appliance by writing extension code. Presently, policy extensions and protocol extensions are supported. Policy extensions can be used to extend the policy language. Protocol extensions can be used to add support for custom protocols on a Citrix ADC appliance.

Citrix ADC extensions are also supported on Citrix ADC CPX.

This document includes the following information:

- Citrix ADC Extensions - Language Overview
- Citrix ADC Extensions - Library Reference
- Citrix ADC Extensions API Reference
- Protocol Extensions
- Policy Extensions

Citrix ADC extensions - language overview

September 14, 2021

The extension language is based on the Lua 5.2 programming language. Lua provides a compact execution engine with good performance that is designed for embedding in C programs, like Citrix ADC software.

The extension language is dynamically typed, which means each object carries its own type information. Any variable can hold any type at any time during execution, so variable types are not declared.

The language is also free form, where white space between tokens is ignored. Statements may be separated by semicolons, but that is not required and usually not done. Blocks of statements are typically terminated by end. There are no brackets around blocks like the { and } in C or Java.

Identifiers are sequences of letters (a through z and A through Z), digits (0 through 9), and underscores (_), not starting in a digit. Identifiers are case-sensitive, so var, VAR, and Var are all different identifiers.

Comments are started by –. Everything after – is ignored to the end of the line. Example:

```
-- This is a comment.
```


Simple types

September 14, 2021

The language allows values of the following simple types:

- Numbers
- Strings
- Boolean
- Nil
- Other Types

Numbers

All numbers (even integers) are represented by IEEE 754 floating point values. Integers up to 2^54 have exact representations. Numeric values can be represented by:

- Signed and unsigned decimal integers (examples: 10, -5)
- Real numbers with decimal points (10.5, 3.14159)
- Real numbers with exponents (1.0e+10)
- Hexadecimals (0xffff0000)

Citrix ADC policy expressions have three numeric types:

- 32-bit integers (num_at)
- 64-bit integers (unsigned_long_at)
- 64-bit floating point (double_at)

All of these are converted into the number type when passed into an extension function, and numbers are converted to the expected policy numeric type when returned.

Strings

Strings are byte sequences of any length. They correspond to the policy **text_at** type. Strings can contain null (0x00) bytes. Arbitrary binary data can be held in strings, including any character code representation (e.g. UTF-8 and full Unicode). However, string functions **likestring.upper()** assume 8-bit ASCII.

Strings are automatically allocated when used. There is no need (or even way) to explicitly allocate buffers for strings. Strings are also automatically deallocated by garbage collection when no longer in use. There is no need (or even way) to explicitly free strings. This automatic allocation and deallocation avoids some common problems in languages like C, such as memory leaks and dangling pointers.

String literals are character strings enclosed in double or single quotes. There is no difference between the two types of quotes: “a string literal” is the same as ‘a string literal’. The usual backslash escaping is available: \s (bell), \b (backspace), \f (form feed), \n (newline/line feed), \t (horizontal tab), \\ (backslash), \“(double quote), and \’ (single quote). Decimal byte values can be entered by a backslash and one to three digits (\d, \dd, \ddd). Hexadecimal byte values can be entered by a backslash, an x, and two hex digits (\xhh)

A special syntax call the long bracket notation can be used for long, multi-line string literals. This notation encloses the string in double square brackets with zero or more equal signs between the brackets – the idea is to come up with a combination of brackets and equals that is not in the string. No escape sequences are honored in the string. Some examples:

[[This is a multi-line string using long bracket notation.]]

[=[This is a multi-line string using long notation with [[and]] and and an unescaped in it.]=]

Long bracket notation can be used to make a multi-line comment. Example:

–[[
This is a multi-line comment.
–]]

Boolean

The usual true and false boolean values are provided. Note that boolean values are different than number values, in contrast to C where zero is assumed to be false and any non-zero value is true.

Nil

nil is a special value that means “no value”. It is its own type and is not equivalent to any other value, in contrast to C where NULL is defined to be zero.

Other types

There are two other types, userdata and threads. These are advanced topics and are not covered here.

Variables

September 14, 2021

Variables hold values that may change during extension execution. Because of dynamic typing, any variable may hold values of any type. There are no type declarations for variables. Instead, a variable’s

type is determined at run time. In fact, the type of a variable's value may change during execution, although this is not a recommended practice. A variable initially has the value nil.

Variable names are identifiers, so are strings of letters, digits, and underscores not beginning in a digit. Examples: headers, combined_headers.

Global variables

In Lua, variables that are not otherwise declared are global within the program. However, global variables are not allowed in policy extension functions, because there are multiple Packet Engines in which a function can be executed, and each Packet Engine has its own memory.

If you use a global variable in your extension, you will get a runtime error: attempt to update or create a global reported in **/var/log/ns.log**.

Typos in variable names are a potential problem, because the variable with the typo will be interpreted as another, global variable, and will not cause a syntax error as in language like C or Java. As noted above, you will get a runtime error instead.

Local variables

A variable may be declared to be local to a block of statements, such as a function. This is done by local variable-name. The variable will be scoped to the block, that is, it will only exist within the block. The local declaration may optionally assign a value to the variable.

Examples:

local headers = {}
local combined_headers = {}

Expressions

September 14, 2021

Expressions compute values from variable and literal values.

- Arithmetic Operations
- Relational Operations
- Logical Operations
- Concatenation
- Length
- Precedence

Arithmetic operations

Arithmetic operations are performed on number values. If a string value is used in an arithmetic operation, it is converted to a number – if that fails, an error is returned.

a + b	add a and b
a - b	subtract b from a
a * b	multiply a and b
a / b	divide a by b
a % b	modulo = a - math.floor(a/b)*b
a ^ b	raise a to the b power; b can be any number
-a	negate a

Relational operations

Relational operations compare two values and return true if the relation is satisfied and false if it is not. Relational operations can be performed between values of any type. If the values are not the same type, false is returned. Numbers are compared in the usual way. Strings are compared using the collating sequence for the current locale.

a == b	a is equal to b
a ~= b	a is not equal to b
a < b	a is less than b
a > b	a is greater than b
a <= b	a is less than or equal to b
a >= b	a is greater than or equal to b

Logical operations

Logical operations are traditionally performed on Boolean values, but in this language they can be performed on any two values. nil and false is considered false and any other value is considered true. Logical operations use short-cut evaluation, where if the first value determines the result of the oper-

ation, the second value is not evaluated.

a and b	if a is false or nil then return a else return b
a or b	if a is not false and not nil then return a else return b
not a	if a is not false or nil return false else return true

The and and or operations can be used for conditional evaluation within an expression:

a or b	can be used to provide a default value b if a is uninitialized (nil). This is useful for optional parameters in functions.
a and b or c	can be used to chose non-nil b or c based on the condition a. If a is true, then a and b returns b, and b or c returns b. If a is false, then a and b returns false and false or c returns c. This is equivalent to a ? b : c in the C programming language.

Concatenation

String concatenation is s1 .. s2. This creates a new string large enough to hold the contents of s1 and s2 and copies the contents to the new string. An error results if s1 or s2 are not strings. Note that repeated concatenation may have considerable copying overhead. If you build a string of n bytes by concatenating one byte at a time, this will copy n*(n+1)/2 bytes. For better performance, you can put pieces of a string to be concatenated into a table (discussed later) and then use the table.concat()function. An example of this is shown in the COMBINE_HEADERS() example.

Length

The length of a string s is returned by #s. The # operator is also used with array tables, as discussed later.

Precedence

Operator precedence determines the order in which operations are performed in an expression, with higher precedence operations done before those with lower precedence. Precedence order can as usual be overridden by parentheses. For example, in `a + b \* c`, * has higher precedence than +, so the expression is evaluated as `a + (b \* c)`.

highest	^
-	not # - (unary)
-	* / %
-	..
-	= ~= < > <= >=
-	and
lowest	or

Operations with the same precedence are performed left to right (left associative), except ^ and .. that are performed right to left (right associative). So a^b^c is evaluated as a^(b^c).

Assignment

September 14, 2021

The assignment statement evaluates an expression and assigns the resulting value to a variable.

```
variable = expression
```

As noted earlier, values of any type can be assigned to any variable, so the following is allowed:

```
local v1 = "a string literal"
v1 = 10
```

An assignment statement can actually set multiple variables, using the form

variable1, variable2, … = expression1, expression2, …

If there are more variables than expressions, the extra variables are assigned nil. If there are more expressions than variables, the extra expression values are discarded. The expressions are all evaluated before the assignments, so this can be used to succinctly exchange the values of two variables:

v1, v2 = v2, v1

is equivalent to

tmp = v1
v2 = v1
v1 = tmp

Tables

September 14, 2021

Tables are collections of entries with keys and values. They are the only aggregate data structure provided. All other data structures (arrays, lists, sets, and so on) are built from tables. Table keys and values can be any type, including other tables. Keys and values within the same table can mix types.

- Table Constructors
- Table Usage
- Tables as Arrays
- Tables as Records

Table constructors

Table constructors allow you to specify a table with keys and associated values. The syntax is:

{[key1] = value1, [key2] = value2, …}

where the keys and values are expressions. If the keys are strings that are not reserved words, the brackets and quotes around the keys can be omitted. Example:

{key1 = “value1”, key2 = “value2”, key3 = “value3”}

An empty table is specified simply by {}.

A table constructor may be used in an assignment to set a variable to refer to a table. Examples:

local t1 = {} – set t1 to an empty table
local t2 = {key1 = “value1”, key2 = “value2”, key3 = “value3”}

Note that tables themselves are anonymous. More than one variable may refer to the same table. Continuing the above example:

local t3 = t2 – both t2 and t3 refer to the same table

Table usage

As you would expect, you can use keys to find values in a table. The syntax is table[key], where table is a table reference (typically a variable assigned a table), and key is an expression providing the key.

If this is used in an expression and the key exists in the table, this returns the value associated with the key. If the key is not in the table, this returns nil. If this is used as the variable in an assignment, and the key does not exist in the table, it creates a new entry for the key and value. If the key already exists in the table, it replaces the key's value with the new value. Examples:

local t = {} – sets t to an empty table
t["k1"] = "v1" – creates an entry for key "k1" and value "v1"
v1 = t["k1"] – sets v1 to the value for key "k1" = "v1"
t["k1"] = "new_v1" – sets the value for key "k1" to "new_v1"

Table as arrays

The traditional array can be implemented using a table with integer keys as indices. An array can have any indices, including negative ones, but the convention is to start arrays at index 1 (not 0 as is the case with languages like C and Java). There is a special purpose table constructor for such arrays:

{value1, value2, value3, … }

Array references are then array[index].

The length operator # returns the number of elements in an array with consecutive indices starting at 1. Example:

local a = {"value1", "value2", "value3"}
local length = #a – sets length to the length of array a = 3

Arrays can be sparse, where only the defined elements are allocated. But # cannot be used on a sparse array with non-consecutive indices. Example:

local sparse_array = {} – set up an empty array
sparse_array[1] = "value1" – add an element at index 1
sparse_array[99] = "value99" – add an element at index 99

Multidimensional arrays can be set up as tables of tables. For example, a 3x3 matrix could be set up by:

local m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9} }
local v22 = m[2][2] – sets v22 to 5

Tables as records

Records with fields can be implemented as tables with field name keys. The reference form table.field can be used for table["field"]. Examples:

local person = {name = "John Smith", phone = "777-777-7777"}
local name = person.name – sets name to "John Smith"

An array of tables can be used for a sequence of records. Example:

local people = {
{name = “John Smith”, phone = “777-777-7777”},
{name = “Jane Doe”, phone = “888-888-8888”}
…
}

name = people[2].name – sets name to “Jane Doe”

Control structures

September 14, 2021

The extension function language provides the usual statements to control program execution.

- If Then Else
- While Do and Repeat Until
- Numeric For
- Break
- Goto

If Then Else

If statements select blocks of statements to execute based on one or more conditions. There are three forms:

If then Form

```
if expression then
     statements to execute if expression is not false or nil
end
<!--NeedCopy-->
```

If then else Form

```
if expression then
     statements to execute if expression is not false or nil
else
     statements to execute if expression is false or nil
end
```



```
<!--NeedCopy-->
```

If then elseif else Form

```
if expression1 then
     statements to execute if expression1 is not false or nil
     elseif expression2 then
          statements to execute if expression2 is not false or nil
. . .
else
     statements to execute if all expressions are false or nil
end
<!--NeedCopy-->
```

Example:

```
if headers[name] then

     local next_value_index = #(headers[name]) + 1
     headers[name][next_value_index] = value

else

     headers[name] = {
 name .. ":" .. value }


end
<!--NeedCopy-->
```

Note:

- The expression is not enclosed in parentheses as is the case in C and Java.
- There is no equivalent to the C/Java switch statement. You have to use a series of if elseif statements to do the equivalent.

While Do and Repeat Until

The **while** and **repeat** statements provide loops controlled by an expression.

```
while expression do
     statements to execute while expression is not false or nil
end
```



```
4
5  repeat
6
7       statements to execute until expression is not false or nil
8
9  until expression
10 <!--NeedCopy-->
```

Example for while:

```
1 local a = {
2  1, 2, 3, 4 }
3
4 local sum, i = 0, 1 -- multiple assignment initializing sum and i
5 while i <= #a do -- check if at the end of the array
6      sum = sum + a[i] -- add array element with index i to sum
7      i = i + 1 -- move to the next element
8 end
9 <!--NeedCopy-->
```

Example for repeat:

```
1 sum, i = 0, 1 -- multiple assignment initializing sum and i
2 repeat
3      sum = sum + a[i] -- add array element with index i to sum
4      i = i + 1 -- move to the next element
5 until i > #a -- check if past the end of the array
6 <!--NeedCopy-->
```

Of course it is possible to write a loop that does not terminate, for example, if you omit the i = i + 1 statement in either of these examples. When such a function is executed, Citrix ADC will detect that the function did not complete in a reasonable time and will kill it with a runtime error:

```
Cpu limit reached. Terminating extension execution in [[string "function
extension function..."]]: line line-number.
```

will be reported in /var/log/ns.log.

Numeric For

There are two types of for loops. The first is the numeric for, which is similar to the usual use of the for statement in C and Java. The numeric for statement initializes a variable, tests if the variable has passed a final value, and if not executes a block of statements, increments the variable, and repeats. The syntax for the numerical for loop is:


```
for variable = initial, final, increment do

     statements in the loop body

end
<!--NeedCopy-->
```

where initial, final, and increment are all expressions that yield (or can be converted to) numbers. variable is considered to be local to the for loop statement block; it cannot be used outside of the loop. increment can be omitted; the default is 1. The expressions are evaluated once at the beginning of the loop. The terminating condition is variable > final if the increment is positive and variable < final if the increment is negative. The loop terminates immediately if the increment is 0.

Example (equivalent to the while and repeat loops in the preceding section):

```
sum = 0
for i = 1, #a do -- increment defaults to 1
     sum = sum + a[i]
end
<!--NeedCopy-->
```

The second type of for loop is the generic for, which can be used for more flexible types of loops. It involves the use of functions, so will be discussed later after functions have been introduced.

Break

The break statement is used inside a while, repeat, or for loop. It will terminate the loop and resume execution at the first statement after the loop. Example (also equivalent to the preceding while, repeat, and for loops):

```
sum, i = 0, 1
while true do
     if i > #a then
          break
     end
     sum = sum + a[i]
     i = i + 1
end
<!--NeedCopy-->
```


Goto

The goto statement can be used to jump forward or backward to a label. The label is an identifier, and its syntax is ::label::. The goto statement is goto label. Example (once again equivalent to the preceding loops):

```
sum, i = 0, 1
::start_loop::
     if i > #a then
          goto end_loop -- forward jump
     end
     sum = sum + a[i]
     i = i + 1
     goto start_loop -- backwards jump
::end_loop::
. . .
<!--NeedCopy-->
```

There has been a long running controversy over using gotos in programming. In general, you should try to use the other control structures to make your functions more readable and reliable. But occasional judicious use of gotos may lead to better programs. In particular, gotos may be useful in handling errors.

Functions

September 21, 2021

Functions are a basic building block of programming – they are a convenient and powerful way to group statements that perform a task. They are the interface between the Citrix ADC appliance and extension code. For policies, you define policy extension functions. For protocols, you implement callback functions for the protocol behaviors. Functions consist of function definitions that specify what values are passed into and out of the function and what statements are run for the function, and function calls, which run functions with specific input data and get results from the function.

Protocol behavior callback functions

The TCP client behavior consists of a callback function (on_data) that processes TCP client data stream events. To implement Message Based Load Balancing (MBLB) for a TCP based protocol, you can add code for this callback function to process the TCP data stream from the client and parse the byte stream into protocol messages.

The callback functions in a behavior are called with a context, which is the processing module state. The context is the instance of the processing module. For example, the TCP client behavior callbacks are called with different contexts for different client TCP connections.

In addition to the context, the behavior callbacks can have other arguments. Usually the rest of the arguments are passed as payload, which is the collection of all the arguments. So, the programmable processing module instances can be seen as a combination of instance state plus event callback functions, that is, the context plus behavior. And the traffic flows through the pipeline as event payload.

Prototype of TCP client callback function:

```
1              Function                client on_data (ctxt, payload)
2
3                                              //.code
4
5              end
```

Where,

- ctxt - TCP client processing context
- payload – event payload
  - payload.data - TCP data received, available as a stream of bytes

Policy extension functions

Since the NetScaler policy expression language is typed, the definition of an extension function must specify the types of its inputs and its return value. The **Lua function** definition has been extended to include these types:

‘

function self-type: function-name(parameter1: parameter1-type, and so on): return-type
statements
end

Where,

The types are NSTEXT, NSNUM, NSBOOL, or NSDOUBLE.

‘

Self-type is the type of the implicit self-parameter that is passed into the function. When the extension function is used in a Citrix ADC policy expression, this is the value generated by the expression to the left of the function. Another way to view this is that the function extends that type in the Citrix ADC policy language.

The parameter-types are the types of each parameter specified in the extension function call in the policy expression. An extension function can have zero or more parameters.

Return-type is the type of the value returned by the extension function call. It is the input to the part of the policy expression, if any, to the right of the function, or else is the value of the expression result.

Example:

```
function NSTEXT:COMBINE_HEADERS(): NSTEXT
```

Use of the extension function in a policy expression:

```
HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n").COMBINE_HEADERS()
```

Here the self-parameter is the result of `HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n")`, which is a text value. The result of the COMBINE_HEADERS() call is text, and since there is nothing to the right of this call, the result of the entire expression is text.

Local function definition

Besides extension functions, no global functions can be defined in an extension file. But local functions can be defined within extension functions using the normal Lua function statement. This declares the name of the function and the names of its parameters (also known as arguments), and like all declarations in Lua, does not specify any types. The syntax for this is:

‘

Local function function-name(parameter1-name, parameter2-name, and so on)
statements
end
‘

The function and parameter names are all identifiers. (The function name is actually a variable and the function statement is shorthand for local function-name = function(parameter1, and so on), but you don’t have to understand this subtlety to use functions.)

Note that and so on is used here for continuation of the pattern of parameter names instead of the usual …. This is because … itself actually means a variable parameter list, which will not be discussed here.

Function body and return

The block of statements between the function and end statements is the function body. In the function body, the function parameters act like local variables, with values supplied by the function calls, as described previously.

The return statement supplies values to be returned to the caller of the function. It must appear at the end of a block (in a function, if then, for loop, and so on; It can be in its own block do return … end). It may specify no, one, or more than one return values:

‘

Return – returns nil
return expression – one return value
return expression1, expression2, … – multiple return values
‘

Examples:

‘

local function fsum(a)
local sum = 0
for i = 1, #a do
sum = sum + a[i]
end
return sum
end

Local function fsum_and_average(a)
local sum = 0
for i = 1, #a do
sum = sum + a[i]
end
return sum, sum/#a
end
‘

Function calls

A function call runs the body of a function, supplying values for its parameters, and receiving results. The syntax for a function call is function-name(expression1, expression2, and so on), where the function parameters are set to the corresponding expressions. The number of expressions and parameters need not be the same. If there are fewer expressions than parameters, the remaining parameters are set to nil. So you can make one or more parameters at the end of the call optional, and your function can check if they are specified by checking if they are not nil. A common way to do this is with the or operation:

‘

Function f(p1, p2) – p2 is optional
p2 = p2 or 0 – if p2 is nil, set to a default of 0
. . .
end
‘

If there are more expressions than parameters, the remaining expression values are ignored.

As noted previously, functions can return multiple values. These returns can be used in a multiple assignment statement. Example:

‘

```
Local my_array = {1, 2, 3, 4}
local my_sum, my_ave = sum_and_average(my_array)
```

‘

Iterator functions and generic for loops

Now that we have introduced functions, we can talk about generic for loops. The syntax for the generic for loop (with one variable) is:

‘

```
For variable in iterator(parameter1, parameter2, and so on) do
statements in the for loop body
end
```

‘

Where iterator() is a function with zero or more parameters that provide a value for variable on each iteration of the loop body. The iterator function keeps track of where it is in the iteration using a technique called closure, which you don’t have to worry about here. It signals the end of the iteration by returning nil. Iterator functions can return more than one value, for use in a multiple assignments.

Writing an iterator function is beyond the scope of this paper, but there are few useful built-in iterators that illustrate the concept. One is the pairs() iterator, which iterates through the entries in a table and returns two values, the key and the value of the next entry.

Example:

‘

```
Local t = {k1 = “v1”, k2 = “v2”, k3 = “v3”}
local a = {} – array to accumulate key-value pairs
local n = 0 – number of key-value pairs
for key, value in pairs(t) do
n = n + 1
a[n] = key.. “ = “.. Value – add key-value pair to the array
end
local s = table.concat(a, “;”) – concatenate all key-value pairs into one string
```

‘

Another useful iterator is the string.gmatch() function, which will be used in the following COMBINE_HEADERS() example.

Citrix ADC extensions - library reference

September 14, 2021

The list of libraries supported in policy extensions.

- Basic library
- String library
- Regular Expression Patterns - Character Classes
- Regular Expression Patterns - Pattern Items
- Table library
- Math library
- Bitwise library
- Operating System library
- Citrix ADC library

Basic library

assert(v[,message])	Issues an error, with an optional message, when v is false.
error(message)	Terminates a function and reports the error message.
ipairs(a)	Iterator for an array a. Returns an index and value for each iteration.
pairs(t)	Iterator for a table t. Returns a key and value for each iteration.
tonumber(e[,base])	Converts e to a number, with an optional base.
tostring(v)	Converts v to a string
type(v)	Returns the type of v: number, string, boolean, table, etc.

getmetatable (object)	Returns nil if the object does not have a metatable. Otherwise, if the object's metatable has a "__metatable" field, returns the associated value. Otherwise, returns the metatable of the given object.
setmetatable (table, metatable)	Sets the metatable for the given table. (You cannot change the metatable of other types from Lua, only from C.) If metatable is nil, removes the metatable of the given table. If the original metatable has a "__metatable" field, raises an error.
select (index, ···)	Returns all arguments after argument number index. If index is string "#", then it returns the total number of extra arguments it received.
pcall (f [, arg1, ···])	Calls function f with the given arguments in protected mode. It returns status code as first result which tells whether call succeeded or not. If call succeeded, then along with status code it also returns all results from the call, otherwise returns error message.
xpcall (f, msgh [, arg1, ···])	This function is similar to pcall, except that it also takes an argument for error handling.
_VERSION	Returns the current interpreter version.

String library

string.byte(s[,i[,j]])	Returns the byte values for s[i] to s[j]. Default i = 1 and j = i
string.char(…)	Returns a string constructed of the integer parameters.
string.find(s,pattern[,init[,plain])	Looks for the first match of a regular expression pattern in s. Return the first and last indices of match or nil. init is index to start, default 1. plain = true means pattern is not a regex.

string.format(form,…)	Returns a formatted version of the parameters.
string.gmatch(s,pattern)	Iterator for searching s with the regex pattern. Returns matching values.
string.gsub(s,pattern,repl[,n])	Returns a copy of s in which all (or n) occurences of the pattern have been replaced by repl.
string.len(s)	Returns the string length.
string.lower(s)	Returns a copy of the string converted to lowercase.
string.match(s,pattern[,init])	Looks for the first match of the regex pattern in s and returns the captures or the whole pattern. init is the index to start, default 1.
string.rep(s,n[,sep])	Returns a string that is n copies of s, with separator sep, default no separator
string.reverse(s)	Returns a string that is s reversed.
string.sub(s,i[,j])	Returns the substring of s from s[i] to s[j], default j is the end of the string.
string.upper(s)	Returns a copy of the string converted to uppercase.
string.dump (function)	Returns a string containing a binary representation of the given function.

Regular expression patterns - character classes

x	the character x, except for magic characters ^$()%.[]*+-?)
.	any character
%a	any letter
%c	any control character
%d	any digit
%g	any printable character except space

%l	any lowercase letter
%p	any punctuation character
%s	any white space character
%u	any uppercase letter
%w	any alphanumber letter
%x	an escaped magic character x (for example %%)
[set]	a set of characters: sequence of individual characters, ranges x-y, and % classes
[^set]	characters not in the set.

Regular expression patterns - pattern items

X	a character class
X*	0 or more longest repetitions of characters in X
X+	1 or more repetitions of characgers in X
X-	0 or more shortest repetitions of characters in X
X?	0 or 1 character in X
%n	n=1 to 9; matches nth captured string
%bxy	matches substring between two balanced characters x and y. Example %b() matches substring between two balanced parentheses.
%f[set]	matches an empty string at at any position such that the next character belongs to set and the previous character does not belong to set.

A pattern is a sequence of pattern items. ^pattern matches the beginning of a string and pattern$ matches the end of the string.

Matched substrings can be captured using (pattern). Parentheses with no pattern () capture the current string position (a number).

Table library

table.concat(list[,sep[,i[,j]]])	Returns a string list[i] .. sep .. list[i+1] .. sep . . . list[j]. Default sep is the empty string. Default i is 1, j is #list.
table.insert(list,[pos,]value)	Inserts value into list at index pos. Default for pos is #list (end of the list).
table.pack(…)	Returns an array containing the parameters starting at index 1, and a key n with the total number of parameters.
table.remove(list[,pos])	Removes from list the element at position pos, shifting elements to fill the position. Returns the removed element. Default for posis #list (end of the list.)
table.sort(list[,comp])	Sort the elements of list in place. comp is the comparison function to use. Default for comp is <.
table.unpack(list[,i[,j]])	Returns list[i] through list[j]. Default for i is 1 and j is #list<c/ode>.

Math library

Various trigonometric and logarithmic functions not shown.

math.abs(x)	Returns the absolute value of x.
math.ceil(x)	Returns the smallest integer >= x.
math.floor(x)	Returns the largest integer <= x.
math.fmod(x,y)	Returns the remainder of x/y rounding the quotient towards zero.
math.huge	A value >= any other number.
math.max(x,…)	Returns the maximum argument.
math.min(x,…)	Returns the minimum argument.
math.modf(x)	Returns the integral and fractional parts of x.

math.random()	Returns a pseudo-random number between 0 and 1.
math.random(m)	Returns a pseudo-random integer between 1 and m.
math.random(m, n)	Returns a pseudo-random integer betwen m and n.
math.randomseed(x)	Sets the pseudo-random number generator set to x.
math.sqrt(x)	Returns the square root of x (x^0.5)
math.acos(x)	Returns the arc cosine of x (in radians).
math.asin(x)	Returns the arc sine of x (in radians).
math.atan(x)	Returns the arc tangent of x (in radians).
math.atan2(y, x)	Returns the arc tangent of y/x (in radians).
math.cos(x)	Returns the cosine of x.
math.cosh(x)	Returns the hyperbolic cosine of x.
math.sin(x)	Returns the sine of x.
math.sinh(x)	Returns the hyperbolic sine of x.
math.tan(x)	Returns the tangent of x.
math.tanh(x)	Returns the hyperbolic tangent of x.
math.deg(x)	Returns the angle x (given in radians) in degrees.
math.exp(x)	Returns the value e^x.
math.frexp (x)	Returns m and e such that x = m2e, e is an integer and the absolute value of m is in the range [0.5, 1).
math.ldexp (m, e)	Returns m2e (e should be an integer).
math.log (x [, base])	Returns the logarithm of x in the given base. The default for base is e.
math.pow (x, y)	Returns x^y.
math.rad (x)	Returns the angle x (given in degrees) in radians.

math.pi	The value of π.

Bitwise library

Unless otherwise stated:

- All functions accept numeric arguments in the range (-2^51,+2^51).
- Each argument is normalized to the remainder of its division by 2^32 and truncated to an integer (in some unspecified way), so that its final value falls in the range [0,2^32 - 1].
- All results are in the range [0,2^32 - 1].

bit32.arshift(x,disp)	Returns x arithmetically shifted disp bits to the right (+disp) or left (-disp).
bit32.band(…)	Returns the bitwise and of the arguments.
bit32.bnot(x)	Returns the bitwise negation of x.
bit32.bor(…)	Returns the bitwise or of the arguments.
bit32.btest(…)	Returns true if the bitwise and of the arguments is not zero.
bit32.bxor(…)	Returns the bitwise exclusive or of the arguments.
bit32.extract(n,field[,width])	Returns the bits in n from field to field + width - 1 (bits number from most to least significant). Default for width is 1.
bit32.replace(n,v,field[,width])	Returns a copy of n with bits from field to field + width -1 replaced by v. Default width is 1.
bit32.lrotate(x,disp)	Returns x rotated disp bits to the left (+disp) or right (-disp).
bit32.lshift(x,disp)	Returns x shifted disp bits to the left (+disp) or right (-disp).
bit32.rrotate(x,disp)	Returns x rotated disp bits to the right (+disp) or left (-disp).
bit32.rshift(x,disp)	Returns x shifted disp bits to the right (+disp) or left (-disp).

Operating system library

os.clock ()	Returns an approximation of the amount in seconds of CPU time.
os.date ([format [, time]])	Returns a string or a table containing date and time, formatted according to the given string format.
os.time ([table])	Returns the current time when called without arguments, or a time representing the date and time specified by the given table.
os.difftime (t2, t1)	Returns the number of seconds from time t1 to time t2.

Citrix ADC library

ns.logger:level(message)	To log messages where level is emergency, alert, critical, error, warning, notice, info, or debug. The parameters are the same as the C printf() function: a format string, and a variable number of arguments to supply values for the % specifiers in the format string.

Citrix ADC extensions API reference

September 14, 2021

Behaviors are a formalization of common programmable patterns that are available on a Citrix ADC appliance. For example, a TCP virtual server supports a TCP client behavior and a TCP server behavior. A behavior is a pre-defined set of callback functions. You can implement behaviors by providing callback functions. For example, a TCP client behavior can consist of the on_data function, which processes the TCP data stream.

TCP client behavior

on_data - function callback for TCP client data events. The callback takes two arguments:

- **ctxt** - TCP client processing context
- **payload** – event payload
  - **payload.data** - TCP data received, available as a stream of bytes

TCP server behavior

on_data - function callback for TCP server data events, the callback takes two arguments:

- **ctxt** - TCP server processing context
- **payload** – event payload
  - **payload.data** - tcp data received, available as a stream of bytes

TCP client context

The context that is passed to the TCP client event callbacks:

- **ctxt.output** - The next processing context in the pipeline. Extension call back handlers can send ns.tcp.stream type data to ctxt.output using the events DATA, which means partial message or EOM which means end of protocol message. The EOM event may or may not have TCP data with it. An EOM event with TCP data can be sent without a preceding DATA event to send a whole protocol message data and mark the end of message. The load balancing decision is made, downstream by the load balancing virtual server, on the first data received. A new load balancing decision is made after the receipt of the EOM message. So, to stream protocol message data, send multiple DATA events with the last event as EOM. All the contiguous DATA events and the following EOM events are sent to the same server connection selected by the load balancing decision on the first DATA event in the sequence.
- **ctxt.input** - The previous processing context in the pipeline where the TCP stream data is coming from.
- **ctxt:hold(data)** - Function to store the data for future processing. On calling hold with data, the data is stored in the context. Later, when more data is received on the same context, newly received data is appended to the previously stored data and the combined data stream is then passed to the on_data callback function. After calling a hold, the data reference is no longer usable and gives error on any usage.
- **ctxt.vserver** - The virtual server context.
- **ctxt.client** – Client connection processing context. This processing context can be used to send data to the client, and to fetch some connection related information like IP address, source and destination ports.

- **ctxt:close()** – Close the client connection by sending FIN to the client. After calling this API, the client processing context is no longer usable and gives error on any usage.

TCP server context

The context that is passed to the TCP server event callbacks:

- **ctxt.output** – The next processing context in the pipeline. Extension call back handlers can send ns.tcp.stream type data to ctxt.output using the events DATA, which means partial message or EOM which means end of protocol message.
- **ctxt.input** - The previous processing context in the pipeline where the TCP stream data is coming from.
- **ctxt:hold(data)** - Function to store the data for future processing. On calling hold with data, the data is stored in the context. Later, when more data is received on the same context, newly received data is appended to the previously stored data and the combined data stream is then passed to the on_data callback function. After calling a hold, the data reference is no longer usable and gives error on any usage.
- **ctxt.vserver** - The virtual server context.
- **ctxt.server** - Server connection processing context. This processing context can be used to send data to the server, and to fetch some connection related information like IP address, source and destination ports.
- **ctxt:reuse_server_connection ()** - This API is used to allow the server connection to be reused for other client connections in the server context only. This API can be used only if an EOM event is used (in ns.send() API) to send the data in the client context. Otherwise, the ADC appliance throws an error.

  To allow a server connection to be reused by other clients, this API must be called at the end of each response message. After calling this API, if more data is received on this server connection, this is treated as an error and the server connection is closed. If this API is not used, then the server connection can be used only for that client for which it was opened. Also, if the same server is selected for another load balancing decision for that client, then the same server connection is used to send the client data. After using this API, server connection stops being tied to the client connection for which it was opened, and can be reused for a new load balancing decision for any other client connection. After calling this API, the server context is no longer usable and throws an error on any usage.

  Note: This API is available in Citrix ADC 12.1 build 49.xx and later.
- **ctxt:close()** – Close the server connection by sending FIN to the server. After calling this API, the client processing context is no longer usable and displays an error on any usage.

  Note: This API is available in Citrix ADC 12.1 build 50.xx and later.

Vserver context

The user virtual server context available through the contexts passed to callbacks:

- **vserver:counter_increment(counter_name)** - Increments the value of a virtual server counter passed as argument. Currently the following built-in counters are supported.
  - **- invalid_messages** – Number of invalid requests/responses on this virtual server.
  - **- invalid_messages_dropped** – Number of invalid requests/responses dropped by this virtual server.
- **vserver.params** - The configured parameters for the user virtual server. Parameters provide configurability of extensions. The extension code can access parameters that are specified in the CLI to add a user virtual server.

Client connection context

Client connection processing context to get connection related information.

- **client.ssl** – SSL context
- **client.tcp** – TCP context
- **client.is_ssl** – True if client connection is SSL-based

Server connection context

Server connection processing context to get connection related information.

- **server.ssl** – SSL context
- **server.tcp** – TCP context
- **server.is_ssl** – True if server connection is SSL-based

TCP context

TCP context operates on TCP protocol.

- **tcp.srcport** – Source port as a number
- **tcp.dstport** - Destination port as a number

IP context

IP context works on IP or IPv6 protocol data.

- **ip.src** - Source IP address context.
- **ip.dst** - Destination IP address context.

Note: This API is available in Citrix ADC 12.1 build 51.xx and later.

IP address context

IP address context works on IP or IPv6 address data.

- **<address>.to_s** - The address string in the appropriate ASCII notation.
- **<address>.to_n** - The address' numeric value as a string of bytes in network order (4 bytes for IPv4 and 16 bytes for IPv6).
- **<address>.version** - Returns 4 for IPv4 and 6 for IPv6.
- **<address>:subnet(<prefix value>)** - Returns the subnet address string after applying the prefix number.
  - For IPv4 address, value must be between 0 and 32
  - For IPv6 address, value must be between 0 and 128.
- **<address>:apply_mask(<mask string>)** - Returns the address string after applying mask string. API validates the version of argument and does appropriate error checking.
- **address>:eq(<address string>)** - Returns true or false based on whether the argument is equivalent to the address object. API validates the version of the arguments.

Note: This API is available in Citrix ADC 12.1 build 51.xx and later.

SSL context

The SSL context provides information related to frontend SSL connection.

- **ssl.cert** – SSL certificate context. For the client connection, it provides client certificate context and for the server connection, it provides server certificate context.
- **ssl.version** - A number that represents the SSL protocol version of the current transaction, as follows:
  - - 0: The transaction is not SSL-based
  - - 0x002: The transaction is SSLv2
  - - 0x300: The transaction is SSLv3
  - - 0x301: The transaction is TLSv1
  - - 0x302: The transaction is TLSv1.1
  - - 0x303: The transaction is TLSv1.2
- **ssl.cipher_name** - SSL Cipher name as string if invoked from an SSL connection, otherwise gives NULL string.
- **ssl.cipher_bits** – Number of bits in cryptographic key.

SSL certificate context

- **Cert.version** – Version number of the certificate. If the connection is not SSL based, returns 0.

- **Cert.valid_not_before** – Date in string format before which certificate is not valid.
- **Cert.valid_not_after** – Date in string format after which certificate is no longer valid.
- **Cert.days_to_expire** – Number of days before which certificate is valid. Returns -1 for expired certificate.
- **Cert.to_pem** – Certificate in binary format.
- **cert.issuer** - Distinguished Name (DN) of the Issuer in the certificate as a name-value list. An equals sign (“=”) is the delimiter for the name and the value, and the slash (“/”) is the delimiter that separates the name-value pairs.

  Following is an example of the returned DN:
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@n
- **cert.auth_keyid** – Context of Authority Key Identifier extension of the X.509 V3 certificate.
  - **auth_keyid.exists** - TRUE if the certificate contains an Authority Key Identifier extension.
  - **auth_keyid.issuer_name** - Issuer Distinguished Name in the certificate as a name-value list.
    An equals sign (“=”) is the delimiter for the name and the value, and the slash (“/”) is the delimiter that separates the name-value pairs.

  Following is an example:
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@n
  - **auth_keyid.keyid** - KeyIdentifier field of the Authority Key Identifier as a blob
  - **auth_keyid.cert_serialnumber** - SerialNumber field of the Authority Key Identifier as a blob.
- **cert.pk_algorithm** - Name of the public key algorithm used by the certificate.
- **cert.pk_size** - Size of the public key used in the certificate.
- **cert.serialnumber** - Serial number of the client certificate. If this is a non-SSL transaction or there is an error in the certificate, this gives an empty string.
- **cert.signature_algorithm** - Name of the cryptographic algorithm used by the CA to sign this certificate.
- **cert.subject_keyid** - Subject KeyID of the client certificate. If there is no Subject KeyID, this gives a zero-length text object.
- **cert.subject** - Distinguished Name of the Subject as a name-value. An equals sign (“=”) separates names and values and a slash (“/”) delimits name-value pairs.

Following is an example:
/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycom

Citrix ADC libraries

- **ns.tcp.stream** - String like library for handling TCP data as a stream of bytes. The maximum size of TCP stream data on which these APIs can work is 128 KB. The ns.tcp.stream library functions can also be called in the usual extension object oriented style of calling. For example, data:len() is the same as ns.tcp.stream.len(data)
  - **ns.tcp.stream.len(data)** - Returns length of data in bytes, similar to Lua's string.len
  - **ns.tcp.stream.find(data, pattern [, init])** - Function similar to Lua's string.find. In addition, it also does partial matching at the end of data. Upon partial match, the start index is returned and the end index becomes nil.
  - **ns.tcp.stream.split(data, length)** - Splits the data into two chunks, the first chunk is of the specified length. After a successful split the original data is no longer usable as a TCP data stream. Any attempt to use it that way causes an error.
  - **ns.tcp.stream.byte(data[, i [, j]])** - Function similar to Lua's string.byte. Returns the internal numerical codes of the characters data[i], data[i+1], …, data[j].
  - n**s.tcp.stream.sub(data, i [, j])** - Function similar to Lua's string.sub. Returns the substring of s that starts at i and continues until j.
  - **ns.tcp.stream.match(data, pattern, [, init])** - Function similar to Lua's string.match. Looks for the first *match* of pattern in string s.
- **ns.send(processing_ctxt, event_name, event_data)** - Generic function to send events to a processing context. Event data is a Lua table that can have any content. The contents depend on the event. After the ns.send() API is called, the data reference is no longer usable. Any attempt to use it causes an error.
- **ns.pipe(src_ctxt, dest_ctxt)** - Using a call to pipe() API, extension code can connect source context to a destination context. After a call to pipe, all the events that are sent from source context to the next module in the pipeline go directly to the destination context. This API is typically used by the module that makes the pipe() call, to remove itself from the pipeline.
- **ns.inet** – Library for internet addresses.
  - **ns.inet.apply_mask(address_str, mask_str)** - returns the address string after applying mask string.
  - **ns.inet.aton(address_str)** - Returns address' numeric value as a string of bytes in network order (4 bytes for IPv4 and 16 for IPv6).
  - **ns.inet.ntoa(byte_str)** - Converts numeric byte value as a string of bytes to address string.
  - **ns.inet.ntohs(number)** - Convert the given network byte order to host byte order. If the input is greater than 2^16 - 1, then throws an error.
  - **ns.inet.htons(number)** - Converts the given host byte order to network byte order. If the input is greater than 2^16 - 1, then throws an error.
  - **ns.inet.ntohl(number)** - Converts the given network byte order to host byte order. If the

input is greater than 2^32 - 1, then throws an error.
- **ns.inet.htonl(number)** - Converts the given host byte order to network byte order. If the input is greater than 2^32 - 1, then throws an error.
- **ns.inet.subnet(address_str, subnet_value)** – Returns the subnet address string after applying given subnet.

Protocol extensions

September 14, 2021

The Citrix ADC appliances have native support for protocols such as HTTP. In addition to this, you can use protocol extensions to add support for custom protocols. Presently only TCP-based custom protocols are supported, for example, Message Queuing Telemetry Transport (MQTT) protocol. For secure transactions, TCP over SSL is also supported.

The protocol extensions on the Citrix ADC appliance are part of the high-level scripting infrastructure available on the Citrix ADC appliance. The scripting language is based on the Lua 5.2 programming language. To add a custom protocol to a Citrix ADC appliance, the user has to write extension code to implement the applicable behaviors. For example, the ns.tcp.client and ns.tcp.server behaviors are applicable to TCP based protocols. To implement a behavior, implement only the callbacks that you want to customize. If callback is not implemented, its default takes effect. For more information about the scripting language, see Citrix ADC Extensions - Language Overview. For details on behaviors, see Citrix ADC Extensions API Reference.

The Citrix ADC protocol extensions can be used for the following:

- Add new protocol support on the Citrix ADC appliance programmatically, using extensions.
- Parse protocol traffic and do protocol-specific message based load balancing (MBLB).
- Configure user defined load balancing persistence.

Protocol extensions - architecture

September 14, 2021

To achieve traffic-level extensibility, the traffic processing on a Citrix ADC appliance is exposed as a pipeline of separate processing modules. Traffic flows through them as it processes it from ingress to egress. These modules in the pipeline follow a shared nothing model. Message passing is used to send the traffic data from one module in the pipeline to the next module.

Certain points in the traffic processing pipeline are made extensible, so that you can add code to customize the Citrix ADC behavior.

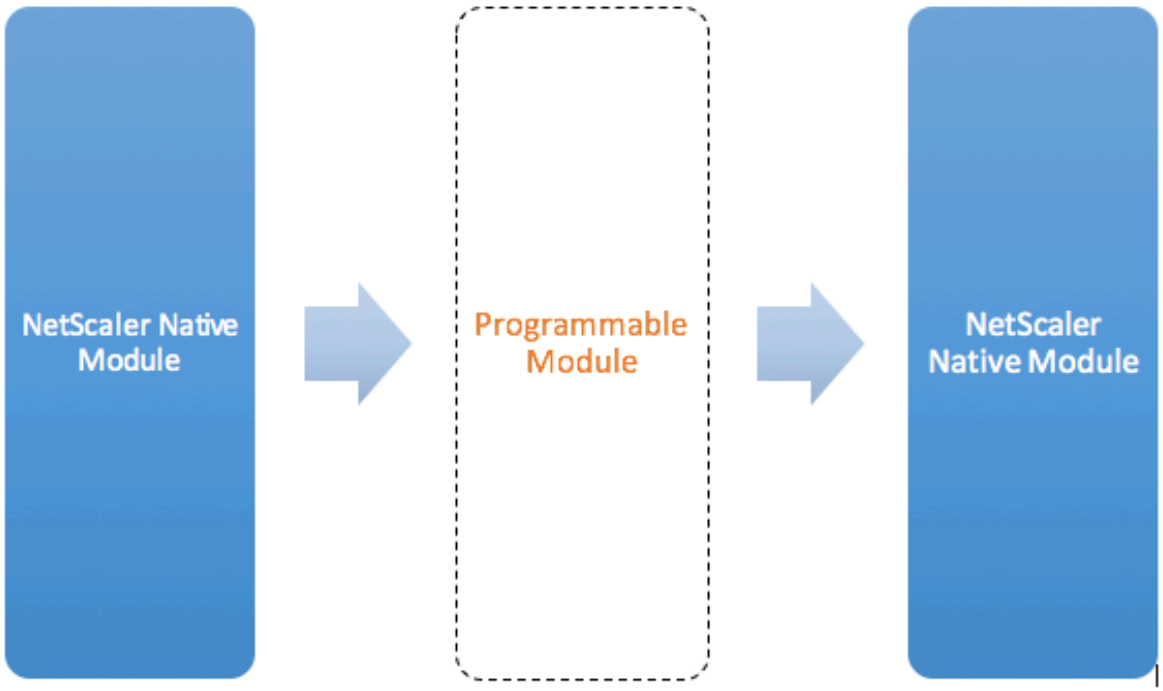


Figure: A Programmable Module In the Traffic Pipeline

By default, the traffic bypasses a programmable module to which you do not add any code.

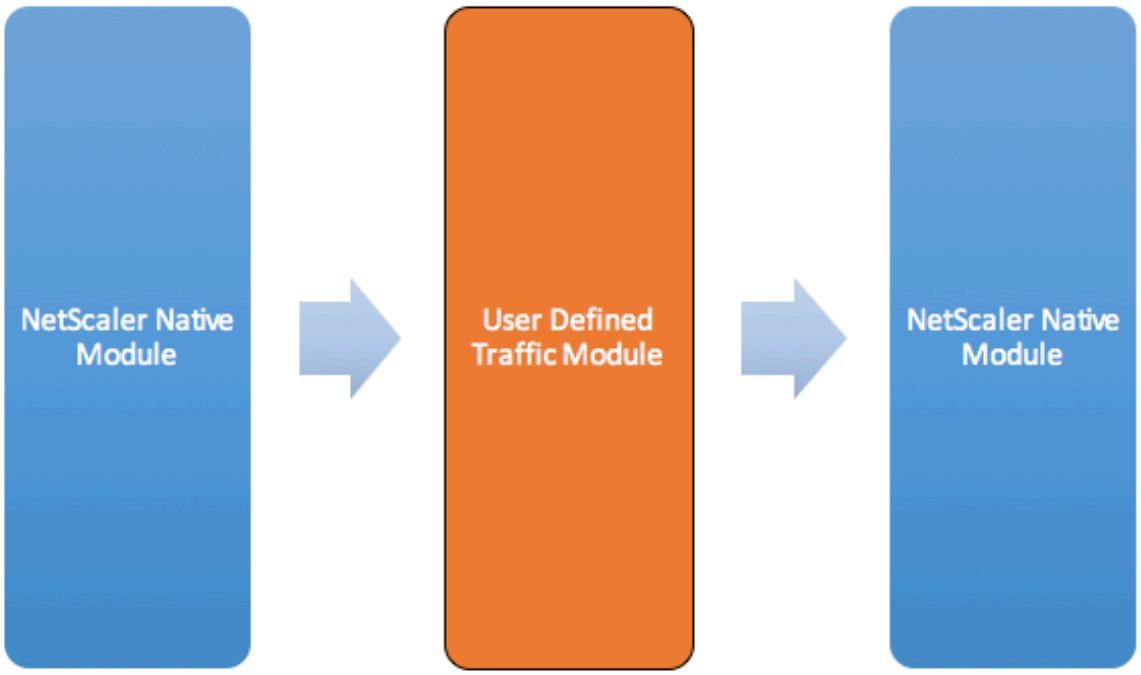


Figure: User Defined Traffic Module

Behaviors

The programmable interfaces for customizing the traffic handling are called behaviors. Behaviors are basically a formalization of common programmable patterns that are available on a Citrix ADC appliance. The behaviors consist of a pre-defined set of event callback functions. You can implement a behavior by providing callback functions conforming to the behavior.

For example, the TCP client behavior consists of a callback function (on_data) that processes TCP client data stream events. To implement Message Based Load Balancing (MBLB) for a TCP based protocol, you can add code for this callback function to process the TCP data stream from the client and parse the byte stream into protocol messages.

Context:

The callback functions in a behavior are called with a context, which is the processing module state. The context is the instance of the processing module. For example, the TCP client behavior callbacks are called with different contexts for different client TCP connections.

Payload:

In addition to the context, the behavior callbacks can have other arguments. Usually the rest of the arguments are passed as payload, which is the collection of all the arguments.

So, the programmable processing module instances can be seen as a combination of instance state plus event callback functions, that is, the context plus behavior. And the traffic flows through the pipeline as event payload.

For Citrix ADC API extensions, see Citrix ADC extension API reference.

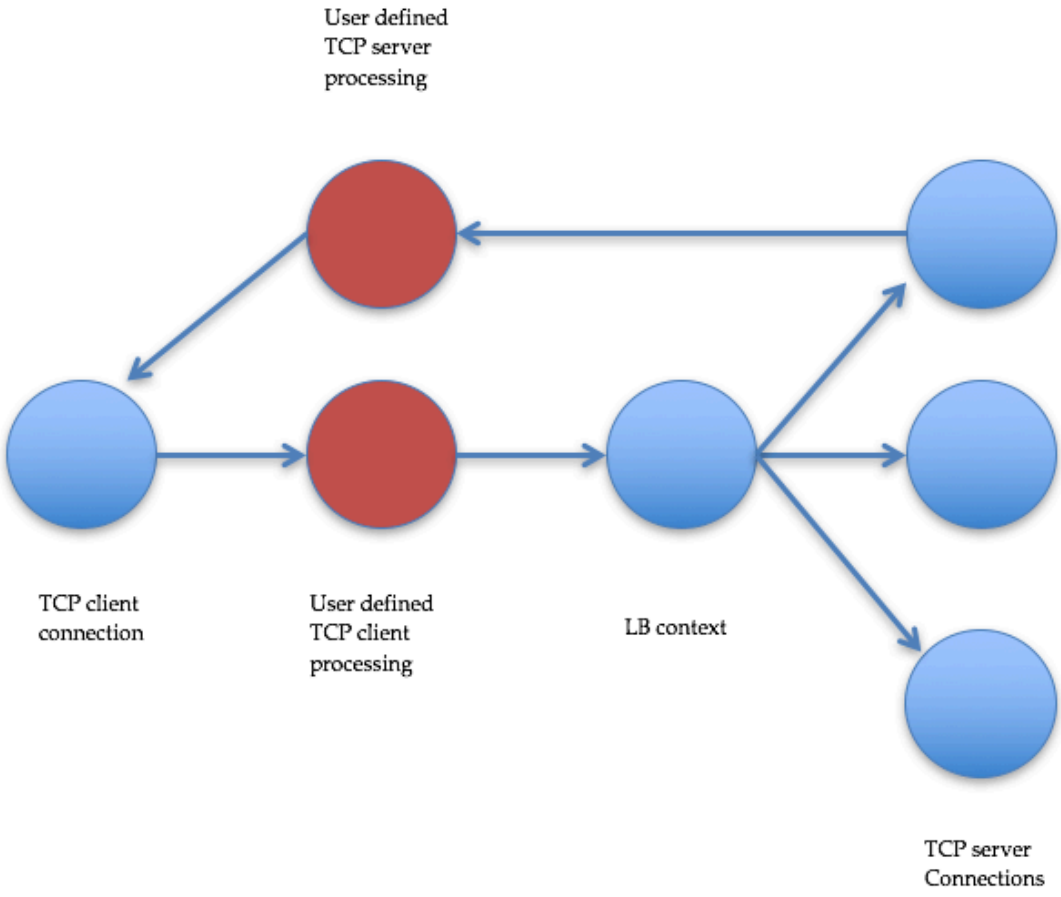
The following code snippet shows a user defined function to handle TCP client data stream events. The context and payload are passed to the function by Citrix ADC code. This code simply forwards the TCP data received in every call to the next processing module context in the pipeline. In this case, the next module is the load balancing (LB) context, which is a Citrix ADC native module.

```
function client.on_data(ctxt, payload)
    ns.send(ctxt.output, "DATA", {
 data = payload.data }
 )
end
<!--NeedCopy-->
```


Protocol extensions - traffic pipeline for user defined TCP client and server behaviors

September 14, 2021

The following figure illustrates the sample protocol extension - traffic pipeline for user defined TCP client and server behaviors



Traffic Pipeline For User Defined TCP Client And Server Behaviors

Add a custom protocol by using protocol extensions

The command line interface (CLI) commands for custom protocol use the keyword “user” to signify the user defined nature of the underlying configuration entities. With the help of extension code, you can add a new user protocol to the system and add user virtual servers for user-defined protocols. The user virtual servers are in turn configurable by setting parameters. Configured values for virtual server parameters are available in the extension code.

The following example illustrates the user flow for adding support for a new protocol. The example adds MQTT protocol support to the system. MQTT is a machine-to-machine "Internet of Things" connectivity protocol. It is a lightweight publish/subscribe messaging transport. Useful for connections with remote locations, this protocol uses client and broker tools to publish messages to subscribers.

1. Import the MQTT protocol extension implementation file to Citrix ADC system. The code listing for mqtt.lua is given below. The example below imports the MQTT extension file hosted on a webserver.

   ```
   import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
   ```

2. Add a new user TCP based protocol to the system using the extension.

   ```
   add user protocol MQTT -transport TCP -extension mqtt_code
   ```

3. Add a user load balancing vserver and bind backend services to it.

   ```
   1 add service mqtt_svr1 10.217.24.48 USER_TCP 1501
   2 add service mqtt_svr2 10.217.24.48 USER_TCP 1502
   3 add lb vserver mqtt_lb USER_TCP - lbmethod USER_TOKEN
   4 bind lb vserver mqtt_lb mqtt_svr1
   5 bind lb vserver mqtt_lb mqtt_svr2
   6 <!--NeedCopy-->
   ```

4. Add a user vserver for the newly added protocol. Set the defaultlb to the LB vserver configured above.

   ```
   add user vserver mqtt_vs MQTT 10.217.24.28 8765 -defaultLb mqtt_lb
   ```

5. Optionally, enable MQTT session persistence based on ClientID, set the persistence type to USERSESSION.

   ```
   set lb vserver mqtt_lb -persistenceType USERSESSION
   ```

Protocol extensions - use cases

September 14, 2021

Protocol extensions can be used for the following use cases.

- Message based load balancing (MBLB)
- Streaming
- Token based load balancing
- Load balancing persistence
- TCP connection based load balancing
- Content based load balancing

- SSL
- Modify traffic
- Originate traffic to client or server
- Process data on the connection establishment

Message based load balancing

Protocol extensions support Message Based Load Balancing (MBLB), which can parse any protocol on a Citrix ADC appliance and load balance the protocol messages arriving on one client connection, that is, distribute the messages over multiple server connections. MBLB is achieved by user code that parses the client TCP data stream.

The TCP data stream is passed to the on_data callbacks for client and server behaviors. The TCP data stream is available to the extension functions through a Lua string like interface. You can use an API similar to the Lua string API to parse the TCP data stream.

Useful APIs include:

```
data:len()
```

```
data:find()
```

```
data:byte()
```

```
data:sub()
```

```
data:split()
```

Once the TCP data stream has been parsed into a protocol message, the user code achieves load balancing by just sending the protocol message to the next context available from the context passed to the on_data callback for the client.

The ns.send() API is used to send messages to other processing modules. In addition to the destination context, the send API takes the event name and optional payload as arguments. There is one-to-one correspondence between the event name and the callback function names for the behaviors. The callbacks for events are called on_<event_name>. The callback names use only lowercase.

For example, the TCP client and server on_data callbacks are user-defined handlers for events named “DATA.” For sending the whole protocol message in one send call, the EOM event is used. EOM, which stands for end of message, signifies the end of protocol message to the LB context down stream, so a new load balancing decision is made for data that follows this message.

The extension code might sometimes not receive the whole protocol message in the on_data event. In such a case, the data can be held by using the ctxt:hold() API. The hold API is available for both TCP-client and server-callback contexts. When “hold with data” is called, the data is stored in the context. When more data is received in the same context, the newly received data is appended to the previously stored data and the on_data callback function is called again with the combined data.

Note: The load balancing method used depends on the configuration of the load balancing virtual server corresponding to the load balancing context.

The following code snippet shows the use of the send API to send the parsed protocol message.

Example:

```
    function client.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to lb
        ns.send(ctxt.output, "EOM", {
 data = message }
 )
    end -- client.on_data

    function server.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to client
        ns.send(ctxt.output, "EOM", {
 data = message }
 )

    end -- server.on_data
<!--NeedCopy-->
```

Streaming

In some scenarios, holding the TCP data stream until the whole protocol message is collected might not be necessary. In fact, it is not advised unless it is required. Holding the data increases memory usage on Citrix ADC appliance and can make the appliance susceptible to DDoS attacks by exhausting the memory on Citrix ADC appliance with incomplete protocol messages on many connections.

Users can achieve streaming of TCP data in the extension callback handlers by using the send API. Instead of holding the data until the whole message is collected, data can be sent in chunks. Sending data to ctxt.output by using the DATA event sends a partial protocol message. It can be followed by more DATA events. An EOM event must be sent to mark the end of the protocol message. The load balancing context downstream makes the load balancing decision on the first data received. A new load balancing decision is made after the receipt of the EOM message.

To stream protocol message data, send multiple DATA events followed by an EOM event. The contiguous DATA events and the following EOM event are sent to the same server connection selected by load

balancing decision for the first DATA event in the sequence.

For a send to client context, EOM and DATA events are effectively the same, because there is no special handling by the client context downstream for EOM events.

Token based load balancing

For natively supported protocols, a Citrix ADC appliance supports a token based load balancing method that uses PI expressions to create the token. For extensions, the protocol is not known in advance, so PI expressions cannot be used. For token based load balancing, you have to set the default load balancing virtual server to use the USER_TOKEN load balancing method, and provide the token value from the extension code by calling the send API with a user_token field. If the token value is sent from the send API and the USER_TOKEN load balancing method is configured on the default load balancing virtual server, the load balancing decision is made by calculating a hash based on the token value. The maximum length of token value is 64 bytes.

```
add lb vserver v\\_mqttlb USER\\_TCP -lbMethod USER\\_TOKEN
```

The code snippet in the following example uses a send API to send an LB token value.

Example:

```
        -- send the message to lb




        -- user_token is set to do LB based on clientID




        ns.send(ctxt.output, "EOM", {
 data = message,

                                user_token = token_info }
 )
<!--NeedCopy-->
```

Load balancing persistence

Load balancing persistence is closely related to token based load balancing. Users have to be able to programmatically calculate the persistence session value and use it for load balancing persistence. The send API is used to send persistence parameters. To use load balancing persistence, you have

to set the USERSESSION persistence type on the default load balancing virtual server and provide a persistence parameter from the extension code by calling the send API with a user_session field. The maximum length of the persistence parameter value is 64 bytes.

If you need multiple types of persistence for a custom protocol, you have to define user persistence types and configure them. The names of the parameters used to configure the virtual servers are decided by the protocol implementer. A parameter's configured value is also available to the extension code.

The following CLI and code snippet shows the use of a send API to support load balancing persistence. The code listing in the section Code Listing for mqtt.lua also illustrates the use of the user_session field.

For persistency, you have to specify the USERSESSION persistency type on the load balancing virtual server and pass the user_session value from the ns.send API.

```
add lb vserver v\\_mqttlb USER\\_TCP -persistencetype USERSESSION
```

Send the MQTT message to the load balancer, with the user_session field set to clientID in the payload.

Example:

```
1 -- send the data so far to lb
2
3 -- user_session is set to clientID as well (it will be used to persist
     session)
4
5 ns.send(ctxt.output,  "DATA" , {
6  data = data, user_session = clientID }
7  )
8 <!--NeedCopy-->
```

TCP connection based load balancing

For some protocols, MBLB might not be needed. Instead, you might need TCP connection based load balancing. For example, the MQTT protocol must parse the initial part of the TCP stream to determine the token for load balancing. And, all the MQTT messages on the same TCP connection must be sent to the same server connection.

TCP connection based load balancing can be achieved by using the send API with only DATA events and not sending any EOM. That way the downstream load balancing context bases the load balancing decision on the data received first, and sends all the subsequent data to the same server connection selected by the load balancing decision.

Also, some use cases might require the ability to bypass extension handling after the load balancing decision has been made. Bypassing the extension calls results in better performance, because the

traffic is processed purely by native code. Bypass can be done by using the ns.pipe() API. A call to the pipe() API extension code can connect input context to an output context. After the call to pipe(), all the events coming from input context directly go to the output context. Effectively, the module from which the pipe() call is made is removed from the pipeline.

The following code snippet shows streaming and the use of the pipe() API to bypass a module. The code listing in the section Code Listing for mqtt.lua also illustrates how to do streaming and the use of pipe() API to bypass the module for rest of the traffic on the connection.

Example:

```
        -- send the data so far to lb
        ns.send(ctxt.output, "DATA", {
 data = data,
                                      user_token = clientID }
 )
        -- pipe the subsequent traffic to the lb - to bypass the client
            on_data handler
        ns.pipe(ctxt.input, ctxt.output)
<!--NeedCopy-->
```

Content based load balancing

For native protocols, content switching like feature for protocol extensions is supported. With this feature, instead of sending the data to the default load balance, you can send the data to the selected load balancer.

Content switching feature for protocol extensions is achieved by using the ctxt:lb_connect(<lbname>) API. This API is available to the TCP client context. Using this API, the extension code can obtain a load balancing context corresponding to an already configured load balancing virtual server. You can then use the send API with the load balancing context thus obtained.

The lb context can be NULL sometimes:

- Virtual server does not exists
- Virtual server is not of user protocol type
- Virtual server's state is not UP
- Virtual server is user virtual server, not load balancing virtual server

If you remove the target load balancing virtual server when it is in use, then all connections associated with that load balancing virtual server is reset.

The following code snippet shows the use of lb_connect() API. The code maps the client ID to load balancing virtual server names (lbname) using the Lua table lb_map and then gets the LB context for lbname using lb_connect(). And finally sends to the LB context using send API.


```
    local lb_map = {

       ["client1*"] = "lb_1",
       ["client2*"] = "lb_2",
       ["client3*"] = "lb_3",
       ["client4*"] = "lb_4"
     }


    -- map the clientID to the corresponding LB vserver and connect to
       it
    for client_pattern, lbname in pairs(lb_map) do
       local match_idx = string.find(clientID, client_pattern)
       if (match_idx == 1) then
      lb_ctxt = ctxt:lb_connect(lbname)
      if (lb_ctxt == nil) then
         error("Failed to connect to LB vserver: " .. lbname)
      end
      break
       end
    end
    if (lb_ctxt == nil) then
    -- If lb context is NULL, the user can raise an error or send data
       to default LB
       error("Failed to map LB vserver for client: " .. clientID)
    end
-- send the data so far to lb
ns.send(lb_ctxt, "DATA", {
 data = data }

<!--NeedCopy-->
```

SSL

SSL for protocols using extensions is supported in ways similar to how SSL for native protocols is supported. Using the same parsing code for creating custom protocols, you can create a protocol instance over TCP or over SSL which can then be used to configure the virtual servers. Similarly, you can add user services over TCP or SSL.

For more information, see Configuring SSL Offloading for MQTT and Configuring SSL Offloading for MQTT With End-To-End Encryption.

Server connection multiplexing

Sometimes, the client sends one request at a time and sends the next request only after the response for the first request is received from the server. In such a case, server connection can be reused for other client connections, and for the next message on the same connection, after the response has been sent to the client. To allow reuse of server connection by other client connections, you must use the ctxt: reuse_server_connection() API on the server side context.

Note: This API is available in Citrix ADC 12.1 build 49.xx and later.

Modify traffic

To modify data in the request or response, you must use the native rewrite feature that uses an advanced policy PI expression. Because you cannot use PI expressions in extensions, you can use the following APIs to modify a TCP stream data.

```
data:replace(offset, length, new_string)
data:insert(offset, new_string)
data:delete(offset, length)
data:gsub(pattern, replace [,n]))
```

The following code snippet shows the use of replace() API.

```
-- Get the offset of the pattern, we want to replace
   local old_pattern =  “pattern to repalace”
local old_pattern_length = old_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
    goto send_data
   end
  -- If the data we want to modify is not completely present, then
  -- wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
data:replace(pat_off, old_pattern_length,  “new pattern” )
::send_data::
ns.send(ctxt.output,  “EOM” , {
 data = data }
 )
::done::
```


The following code snippet shows the use of insert() API.

```
data:insert(5,  “pattern to insert” )
```

The following code snippet shows the use of insert() API, when we want to insert after or before some pattern:

```
-- Get the offset of the pattern, after or before which we want to
    insert
   local pattern =  “pattern after/before which we need to insert”
local pattern_length = pattern:len()
   local pat_off, pat_end = data:find(pattern)
-- pattern is not present
   if (not pat_off) then
    goto send_data
   end
  -- If the pattern after which we want to insert is not
  -- completely present, then wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
-- Insert after the pattern
data:insert(pat_end + 1,  “pattern to insert” )
   -- Insert before the pattern
data:insert(pat_off,  “pattern to insert” )
::send_data::
    ns.send(ctxt.output,  “EOM” , {
 data = data }
 )
::done::
```

The following code snippet shows the use of delete() API.

```
-- Get the offset of the pattern, we want to delete
   local delete_pattern =  “pattern to delete”
local delete_pattern_length = delete_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
          goto send_data
   end
  -- If the data we want to delete is not completely present,
  -- then wait for more data
```



```
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
    goto done
  end
data:delete(pat_off, delete_pattern_length)
::send_data::
ns.send(ctxt.output,  "EOM" , {
 data = data }
 )
::done::
```

The following code snippet shows the use of gsub() API.

```
    -- Replace all the instances of the pattern with the new string
data:gsub( "old pattern" ,  "new string" )
-- Replace only 2 instances of  "old pattern"
data:gsub( "old pattern" ,  "new string" , 2)
-- Insert new_string before all instances of  "http"
data:gsub( "input data" ,  "(http)" ,  "new_string%1" )
-- Insert new_string after all instances of  "http"
data:gsub( "input data" ,  "(http)" ,  "%1new_string" )
-- Insert new_string before only 2 instances of  "http"
data:gsub( "input data" ,  "(http)" ,  "new_string%1" , 2)
```

Note: This API is available in Citrix ADC 12.1 build 50.xx and later.

Originate traffic to client or server

You can use the ns.send() API to send data that originates from the extension code to a client and a back-end server. To send or receive response directly with a client, from client context, you must use ctxt.client as the target. To send or receive response directly with a back-end server from server context, you must use ctxt.server as the target. The data in the payload can be a TCP stream data or a Lua string.

To stop traffic processing on a connection, you can use ctxt:close() API from either the client or the server context. This API closes the client-side connection or any server connections linked to it.

When you call the ctxt:close() API, extension code sends TCP FIN packet to the client and server connections and if more data is received from the client or server on this connection, then the appliance resets the connection.

The following code snippet shows the use of ctxt.client and ctxt:close() APIs.

```
    -- If the input packet is not MQTT CONNECT type, then
```



```
-- send some error response to the client.
function client.on_data(ctxt, payload)
    local data = payload.data
    local offset = 1
    local msg_type = 0
    local error_response =  “Missing MQTT Connect packet.”
    byte = data:byte(offset)
msg_type = bit32.rshift(byte, 4)
if (msg_type ~= 1) then
-- Send the error response
   ns.send(ctxt.client,  “DATA” , {
 data = error_response }
 )
-- Since error response has been sent, so now close the connection
    ctxt:close()
end
```

The following code snippet shows the example when user can inject the data in the normal traffic flow.

```
-- After sending request, send some log message to the server.
function client.on_data(ctxt, payload)
local data = payload.data
local log_message =  “client id :  “..data:sub(3, 7)..”  user name :  “
    data:sub(9, 15)
-- Send the request we get from the client to backend server
ns.send(ctxt.output,  “DATA” , {
 data = data }
 )
After sending the request, also send the log message
ns.send(ctxt.output,  “DATA” , {
 data = log_message”  }
 )
end
```

The following code snippet shows the use of ctxt.to_server API.

```
-- If the HTTP response status message is  “Not Found” ,
-- then send another request to the server.
function server.on_data(ctxt, payload)
    local data = payload.data
    local request  “GET /default.html HTTP/1.1\r\n\r\n” ss
    local start, end = data:find( “Not Found” )
    if (start) then
    -- Send the another request to server
        ns.send(ctxt.server,  “DATA” , {
```



```
10   data = request }
11   )
12  end
```

Note: This API is available in Citrix ADC 12.1 build 50.xx and later.

Data processing on the connection establishment

There might be a use case where you want to send some data at the connection establishment (when the final ACK is received). For example, in proxy protocol, you might want to send client's source and destination IP addresses and ports to the back-end server at the connection establishment. In this case, you can use client.init() callback handler to send the data on connection establishment.

The following code snippet shows the use of client.init() callback:

```
1  -- Send a request to the next processing context
2  -- on the connection establishment.
3  function client.init(ctxt)
4     local request  "PROXY TCP4"  + ctxt.client.ip.src.to_s +  "  " +
          ctxt.client.ip.dst.to_s +  "  " + ctxt.client.tcp.srcport +  "  "
           +     ctxt.client.tcp.dstport
5  -- Send the another request to server
6     ns.send(ctxt.output,  "DATA" , {
7   data = request }
8   )
9    end
```

Note: This API is available in Citrix ADC 13.0 build xx.xx and later.

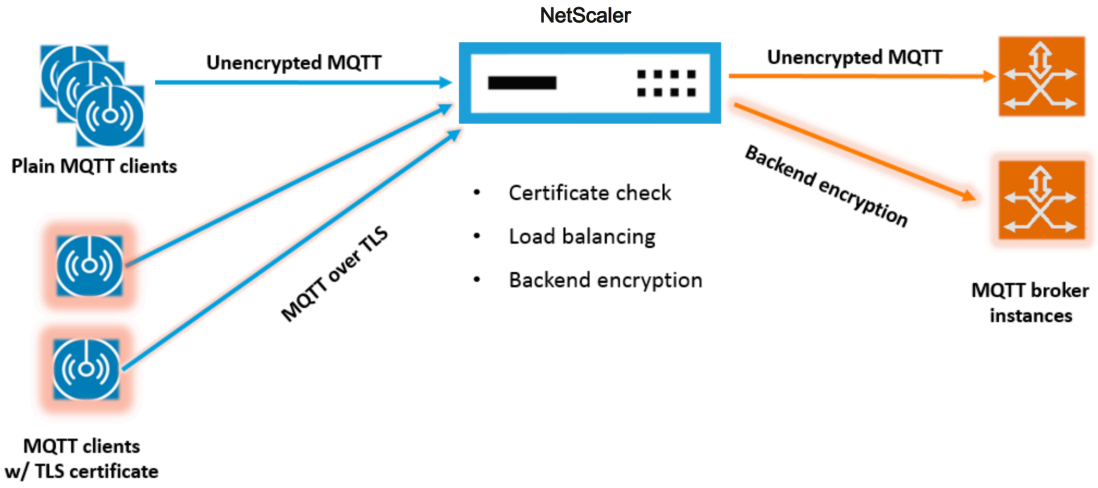
Tutorial – Add MQTT protocol to the Citrix ADC appliance by using protocol extensions

September 14, 2021

The command line interface (CLI) commands for custom protocol use the keyword "user" to signify the user defined nature of the underlying configuration entities. With the help of extension code, you can add a new user protocol to the system and add user virtual servers for user-defined protocols. The user virtual servers are in turn configurable by setting parameters. Configured values for virtual server parameters are available in the extension code.

MQTT protocol is used for illustration purposes.

The following diagram illustrates a Citrix ADC appliance and MQTT client and broker tools.



Code listing for mqtt.lua

September 14, 2021

The code listing below, mqtt.lua, gives the code to implement the MQTT protocol on Citrix ADC using protocol extensions. The code only has the TCP client data callback function defined - client.on_data(). For server data, it does not add a callback function and the server to client takes the fast native path. For client data, the code parses the CONNECT MQTT protocol message and extracts the ClientID. It then uses the ClientID for user_token value, which is used to load balance all the client traffic for the connection based on the ClientID by setting LB method for the LB vserver as USER_TOKEN. It uses the ClientID also for user_session value, which can be used for LB persistence by setting persistence type for the LB vserver as USERSESSION. The code uses the ns.send() to do LB and send the initial data. It uses the ns.pipe() API to send the rest of the client traffic directly to server connection, bypassing calls to extension callback handler.

```
--[[

  MQTT event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

    - parse the client ID from the connect message - the first message
       should be connect

```



```
11      - send the data to LB with ClientID as user token and session
12
13      - pipe the subsequent data to LB directly. This way the subsequent
            MQTT traffic will
14
15        bypass the tcp client on_data handler
16
17      - if a parse error is seen, throw an error so the connection is
            reset
18
19  --]]
20
21  function client.on_data(ctxt, payload)
22
23      local data = payload.data
24
25      local data_len = data:len()
26
27      local offset = 1
28
29      local byte = nil
30
31      local utf8_str_len = 0
32
33      local msg_type = 0
34
35      local multiplier = 1
36
37      local max_multiplier = 128 * 128 * 128
38
39      local rem_length = 0
40
41      local clientID = nil
42
43      -- check if MQTT fixed header is present (fixed header length is
            atleast 2 bytes)
44
45      if (data_len < 2) then
46
47          goto need_more_data
48
49      end
50
51      byte = data:byte(offset)
52
```



```
53      offset = offset + 1
54
55         -- check for connect packet - type value 1
56
57      msg_type = bit32.rshift(byte, 4)
58
59      if (msg_type ~= 1) then
60
61          error("Missing MQTT Connect packet.")
62
63      end
64
65      -- parse the remaining length
66
67      repeat
68
69          if (multiplier > max_multiplier) then
70
71             error("MQTT CONNECT packet parse error - invalid Remaining
                   Length.")
72
73         end
74
75         if (data_len < offset) then
76
77            goto need_more_data
78
79         end
80
81         byte = data:byte(offset)
82
83         offset = offset + 1
84
85         rem_length = rem_length + (bit32.band(byte, 0x7F) * multiplier)
86
87         multiplier = multiplier * 128
88
89      until (bit32.band(byte, 0x80) == 0)
90
91      -- protocol name
92
93      -- check if protocol name length is present
94
95      if (data_len < offset + 1) then
96
```



```
97          goto need_more_data
98
99       end
100
101      -- protocol name length MSB
102
103      byte = data:byte(offset)
104
105      offset = offset + 1
106
107      utf8_str_len = byte * 256
108
109      -- length LSB
110
111      byte = data:byte(offset)
112
113      offset = offset + 1
114
115      utf8_str_len = utf8_str_len + byte
116
117         -- skip the variable header for connect message
118
119      -- the four required fields (protocol name, protocol level, connect
             flags, keep alive)
120
121      offset = offset + utf8_str_len + 4
122
123      -- parse the client ID
124
125      --
126
127      -- check if client ID len is present
128
129      if (data_len < offset + 1) then
130
131         goto need_more_data
132
133      end
134
135         -- client ID length MSB
136
137      byte = data:byte(offset)
138
139      offset = offset + 1
140
```



```
141     utf8_str_len = byte * 256
142
143     -- length LSB
144
145     byte = data:byte(offset)
146
147     offset = offset + 1
148
149     utf8_str_len = utf8_str_len + byte
150
151     if (data_len < (offset + utf8_str_len - 1)) then
152
153         goto need_more_data
154
155     end
156
157     clientID = data:sub(offset, offset + utf8_str_len - 1)
158
159     -- send the data so far to lb, user_token is set to do LB based on
           clientID
160
161     -- user_session is set to clientID as well (it will be used to
           persist session)
162
163     ns.send(ctxt.output, "DATA", {
164 data = data,
165
166                                 user_token = clientID,
167
168                                 user_session = clientID }
169 )
170
171        -- pipe the subsequent traffic to the lb - to bypass the
              extension handler
172
173     ns.pipe(ctxt.input, ctxt.output)
174
175     goto parse_done
176
177     ::need_more_data::
178
179     ctxt:hold(data)
180
181     ::parse_done::
182
```



```
183     return
184
185 end
186 <!--NeedCopy-->
```

Configure MQTT by using protocol extensions

September 14, 2021

The following steps add a MQTT protocol to the Citrix ADC appliance.

Import the extension file to the Citrix ADC appliance, from either a web server (using HTTP) or your local workstation. For details about importing the extension file, see Import extensions.

```
import ns extension local:mqtt_generic_fs.lua mqtt_code
```

Add a new user TCP based protocol to the system by using the extension.

```
add user protocol MQTT -transport TCP -extension mqtt_code
```

Add a service of type USER_TCP to indicate that this is a user-defined protocol.

```
add service s1 10.102.90.112 USER_TCP 80
```

Add a user load balancing vserver and bind backend services to it.

```
add lb vs mysv USER_TCP
```

```
bind lb vs mysv s1
```

Add a user virtual server for the newly added protocol and make the load balancing virtual server configured in the previous step the default load balancer.

```
add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv
```

Optionally, enable MQTT session persistence based on ClientID, set the persistence type to USERSESSION.

```
set lb vserver mqtt_lb -persistenceType USERSESSION
```

Configuring SSL offloading for MQTT

September 14, 2021

You can implement SSL offloading for user protocols by adding an SSL instance for the protocol. The example below shows how to do SSL offloading for a user protocol. The traffic to backend services is unencrypted with this configuration.

Note: This example does not provide details related to adding or updating a certificate-key pair and binding it to a virtual server. For those details, see SSL certificates.

The following commands add the MQTT_SSL protocol by including mqtt.lua with transport value "SSL."

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

The following commands add a user load balancing virtual server and bind backend services to it.

```
add service mqtt_svr1 10.217.24.48 USER_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_TCP 1502
add lb vserver mqtt_lb USER_TCP  - lbMethod ROUNDROBIN
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

The following command adds a user virtual server for the newly added protocol MQTT_SSL. Using MQTT_SSL means the Citrix ADC appliance will do SSL offloading, because MQTT_SSL was configured with SSL transport. The command also sets the defaultlb to the load balancing virtual server configured in the previous step.

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb mqtt_lb
```

For SSL offloading, you also need to enable the SSL feature and bind a certkey to the user virtual server. For more information, see the following topics:

Add or update a certificate-key pair

Bind the certificate-key pair to the SSL virtual server

Example:

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

Configuring SSL offloading with end-to-end encryption for MQTT

September 14, 2021

The following example shows how to do SSL offloading for MQTT with end-to-end encryption.

Note: This example does not provide details related to adding or updating a certificate-key pair and binding it to a virtual server. For those details, see SSL certificates.

The following commands import the extension file and add the MQTT_SSL protocol with SSL transport.

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

The following commands add a user load balancing virtual server and bind backend services to it. Both the load balancing virtual server and the services are configured for the service type USER_SSL_TCP.

```
add service mqtt_svr1 10.217.24.48 USER_SSL_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_SSL_TCP 1502
add lb vserver mqtt_lb USER_SSL_TCP  - lbmethod RR
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

The following command adds a user virtual server for the newly added protocol MQTT_SSL. Using MQTT_SSL means the Citrix ADC appliance will do SSL offloading, because MQTT_SSL was configured with SSL transport. The command also makes the load balancing virtual server, configured in previous step, the default load balancer.

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb mqtt_lb
```

For end-to-end encryption, you also need to enable the SSL feature and bind a certkey to the user and default load balancing virtual servers. For more information, see the following topics:

Add or update a certificate-key pair

Bind the certificate-key pair to the SSL virtual server

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_lb  -certkeyName mqtt_svr_cert_key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```


Tutorial - load balancing syslog messages by using protocol extensions

September 14, 2021

Syslog protocol available on the Citrix ADC appliance works only for the messages generated on the Citrix ADC appliance. It does not load balance the messages coming from external nodes. To load balance such messages, you need to use the protocol extensions feature and write the syslog message parsing logic by using the Lua 5.2 programming language.

Code for parsing syslog message

The code only has the TCP client data callback function defined - client.on_data(). For server data, it does not add a callback function and the server to client takes the fast native path. The code identifies message boundary based on the trailing character. If the TCP packet contains more than one syslog messages, then we split the packet based on the trailing character and load balance each message.

```
--[[

  Syslog event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

--]]

function client.on_data(ctxt, payload)

                  local message = nil

                  local data_len

                  local data = payload.data

                  local trailing_character = "\n"

                  ::split_message::

                                    -- Get the offset of trailing
                                       character

                                    local new_line_character_offset =
                                       data:find(trailing_character)
```



```
26
27                                  -- If trailing character is not
                                       found, then wait for more data.
28
29                                  if (not new_line_character_offset)
                                       then
30
31                                                  goto
                                                       need_more_data
32
33                                  end
34
35                                  -- Get the length of the current
                                       message
36
37                                  data_len = data:len()
38
39                                  -- Check whether we have more than
                                       one message
40
41                                  -- by comparing trailing character
                                       offset and
42
43                                  -- current data length
44
45                                  if (data_len >
                                       new_line_character_offset) then
46
47                                                  -- If we have
                                                       more than one
                                                       message, then
                                                       split
48
49                                                  -- the data into
                                                       two parts such
                                                        that first
                                                       part
50
51                                                  -- will contain
                                                       message upto
                                                       trailing
                                                       character
52
53                                                  -- offset and
                                                       second part
```



```
                                                                will contain
54
55                                                          -- remaining
                                                                message.
56
57                                                          message, data =
                                                                data:split(
                                                                new_line_character_offse
                                                                )
58
59                                      else
60
61                                                          message = data
62
63                                                          data = nil
64
65                                      end
66
67  -- Send the data to the backend server.
68
69                                      ns.send(ctxt.output, "EOM", {
70   data = message }
71   )
72
73                                      goto done
74
75                                      ::need_more_data::
76
77                                                          -- Wait for more
                                                                data
78
79                                                          ctxt:hold(data)
80
81                                                          data = nil
82
83                                                          goto done
84
85                                      ::done::
86
87                                                          -- If we have
                                                                more data to
                                                                parse,
88
89                                                          -- then do
                                                                parsing again.
```



```
90
91                                                  if (data) then
92
93                                                                  goto
                                                                      split_
94
95                                                  end
96
97  end
98  <!--NeedCopy-->
```

Configuring syslog protocol by using protocol extensions

September 14, 2021

The following steps add an user SYSLOG protocol to the Citrix ADC appliance.

Import the extension file to the Citrix ADC appliance, from either a web server (using HTTP) or your local workstation. For details about importing the extension file, see Importing Extensions.

```
import ns extension local:syslog_parser.lua syslog_parser_code
```

Add a new user TCP-based protocol to the system by using the extension.

```
add user protocol USER_SYSLOG -transport TCP -extension syslog_parser_code
```

Add a service of type USER_TCP to indicate that this is a user-defined protocol.

```
add service s1 10.102.90.112 USER_TCP 80
```

Add a user load balancing vserver and bind backend services to it.

```
1  add lb vs mysv USER_TCP
2
3  bind lb vs mysv s1
4  <!--NeedCopy-->
```

Add a user virtual server for the newly added protocol and make the load balancing virtual server configured in the previous step the default load balancer.

```
add user vs v_syslog USER_SYSLOG 10.217.24.28 80 -defaultlb mysv
```


Protocol extensions command reference

September 14, 2021

The following table lists all the new commands added for custom protocols, and the existing commands that have been modified for custom protocols.

```
show lb persistentSessions [<vserv-name>]
```

- **CLI command:**

  ```
  add user protocol <name> -transport ( TCP | SSL )-extension <string> -comment <string>]>
  ```

- **Description:**

  Adds a new user protocol to the Citrix ADC appliance by using extensions. Currently only user protocols with transport value TCP or SSL are supported.

  > **Example:**
  >
  > add user protocol MQTT -transport TCP -extension mqtt_code

- **CLI command:**

  ```
  rm user protocol <name>
  ```

- **Description:**

  Removes a user protocol previously added to the Citrix ADC appliance.

  > **Example:**
  >
  > rm user protocol mqtt

- **CLI command:**

  ```
  set user protocol <name> -comment <string>
  ```

- **Description:**

  Changes settings for a user protocol previously added to the Citrix ADC appliance.

  > **Example:**
  >
  > set user protocol mqtt -comment “MQTT protocol implementation”

- **CLI command:**

  ```
  unset user protocol <name> -comment
  ```

- **Description:**

  Removes settings for a user protocol previously added to the Citrix ADC appliance.

> **Example:**
>
> unset user protocol mqtt -comment “MQTT protocol implementation”

- **CLI command:**

  ```
  update ns extension <extension name>
  ```

- **Description:**

  Updates the implementation for a previously added user protocol by using extensions.

  You can update the protocol implementation only if the protocol is not being used by any user virtual server.

  > **Example:**
  >
  > update ns extension my-extension

- **CLI command:**

  ```
  add lb vserver <name> [USER_TCP | USER_SSL_TCP] [-lbmethod USER_TOKEN]
  [-persistencetype USERSESSION] [-timeout <value>]
  ```

- **Description:**

  Adds a load balancing virtual server to the Citrix ADC appliance. This is an existing CLI command.

  For load balancing user virtual servers, the service type to use is USER_TCP or USER_SSL_TCP. The IP address and port are not allowed with user load balancing virtual servers.

  For user load balancing virtual servers, only the ROUNDROBIN load balancing method is allowed, and the token value is provided by the extension code. Similarly, only USERSESSION persistence is allowed, and the persistence setting is provided by the extension code.

  > **Example:**
  >
  > add lb vserver mysv USER_TCP –lbmethod ROUNDROBIN

- **CLI command:**

  ```
  add user vserver <name> <userProtocol> <IPAddress> <port> -defaultLB <
  string> [-params <string>] [-comment <string>]
  ```

- **Description:**

  Adds a virtual server for a user protocol by using extensions. The configured default user load balancing virtual server is available to the TCP client data extension handler as ctxt.output. For a virtual server, extension parameters can be set by using the -params option with a name and a value pair. The corresponding parameter value is available to the extension handlers as ctxt.vserver.params.<paramName>.

> **Example:**
>
> add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv

- **CLI command:**

```
rm user vserver <name>
```

- **Description:**

  Removes a user virtual server previously added to to the Citrix ADC appliance.

> **Example:**
>
> rm user vserver v_mqtt

- **CLI command:**

```
set user vserver <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-defaultLB
<string>] [-params <string>] [-comment <string>]
```

- **Description:**

  Changes settings for a user virtual server previously added to to the Citrix ADC appliance. When an extension parameter is assigned a new value by the -params option, the old value is overwritten.

> **Example:**
>
> set user vs v_mqtt MQTT 10.217.24.28 -defaultlb mysv -comment “MQTT protocol implementation”

- **CLI command:**

```
unset user vserver <name> [-params] [-comment]
```

- **Description:**

  Removes the settings for a user virtual server previously added to to the Citrix ADC appliance. If you use the –params option to unset an extension parameter, the corresponding parameter value available to extension handlers is changed to nil.

> **Example:**
>
> unset user vs v_mqtt MQTT 10.217.24.28 -defaultlb mysv -comment “MQTT protocol implementation”

- **CLI command:**

```
show user protocol [<name>]
```

- **Description:**

  Displays information about a user protocol, such as extension and callbacks.

> **Example:**
>
> show user protocol mqtt

- **CLI command:**

  `show user vserver [<name>]`

- **Description:**

  Displays information about a user virtual server.

  > **Example:**
  >
  > show user vserver vs_mqtt

- **CLI command:**

  `stat user vserver [<name>]`

- **Description:**

  Displays statistics about a user virtual server.

  > **Example:**
  >
  > stat user vserver vs_mqtt

- **CLI command:**

  `show lb persistentSessions [<vserv-name>]`

- **Description:**

  Displays information about persistent sessions. This is an existing CLI. For user protocols, the persistence type is shown as USERSESSION.

- **CLI command:**

  `rm lb vserver <name>`

- **Description:**

  Removes a user LB vserver previously added to the Citrix ADC appliance.

  > **Example:**
  >
  > rm lb vserver mysv

- **CLI command:**

  `add service <name> <IPAddr> (USER_TCP | USER_SSL_TCP)<Port>`

- **Description:**

  Adds a backend service to be used for a user protocol. This is an existing CLI command with new service types USER_TCP and USER_SSL_TCP.

> **Example:**
>
> add service mqtt_svr1 10.217.24.48 USER_TCP 1501
>
> **Note**: The existing “set service and unset service” commands can be used to remove or change the settings of a previously added service for a user protocol.

- **CLI command:**

  `bind lb vserver <name> <serviceName>`

- **Description:**

  Binds a service to a user LB vserver. The service type should be USER_TCP/USER_SSL_TCP for binding to an LB vserver with type of USER_TCP/USER_SSL_TCP.

  > **Example:**
  >
  > bind lb vserver mysv mqtt_svr1

- **CLI command:**

  `unbind lb vserver <name> <serviceName>`

- **Description:**

  Unbinds a previously bound service to a user LB vserver.

  > **Example:**
  >
  > unbind lb vserver mysv mqtt_svr1

- **CLI command:**

  `rm service <name>`

- **Description:**

  Removes a service that is previously added for a user protocol.

  > **Example:**
  >
  > rm service mqtt_svr1

Troubleshooting protocol extensions

September 14, 2021

If your extension function is not behaving as expected, you can use extension tracing functionality to verify the behavior of your extension function. You can also add logging to your extension function by

using the custom logging functionality, where you can define the log level to be captured on the Citrix ADC appliance.

Custom logging

You can also add your own logging to your extension function. To do so, use the built-in ns.logger:level() function, where level is emergency , alert , critical , error , warning , notice , info , or debug . The parameters are the same as the C printf() function: a format string, and a variable number of arguments to supply values for the % specified in the format string. For example, you might add the following to the COMBINE_HEADERS function to log the result of a call:

```
local result\_str = table.concat(combined\_headers, "\\r\\n") .. "\\r\\
   n\\r\\n"

ns.logger:info("Result: %s", result\_str)

return result\_str
<!--NeedCopy-->
```

The above function would log the following message to/var/log/ns.log for the sample input shown in the abbreviated log messages examples in the Extension Tracing section above.

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M
 H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)libcurl
/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: \*/\*^M H2: h2val1, h2val2,
h2val3^M ^M"
```

Policy extensions

September 14, 2021

The policy extension feature enables you to write extension functions for built-in policy types. The extensions can be used in policy expressions, just like built-in functions. They are executed when the corresponding policy expressions are evaluated. This feature is useful for:

- Adding customized functions to existing Policies.
- Implementing logical constructs for complex customer requirements.

The policy extension feature addresses these limitations by enabling users to write extension functions for built in Policy types. The extensions can then be used in the policy expressions, just like built-in functions. They are executed when the corresponding policy expressions are evaluated.

The following table lists the policy types that can be used when writing an extension, and their associated mappings.

Policy Type	Mapped Policy Type	Output
TEXT_T	NSTEXT	String
BOOL_AT	NSBOOL	Boolean
NUM_AT	NSNUM	Number (double-precision floating point)
DOUBLE_AT	NSDOUBLE	Number (double-precision floating point)

Prerequisites for using policy extensions

The imported functions must conform to the existing policy standards. Therefore:

- The function name must start with a letter and may contain numbers or underscores.
- The function name is treated as case insensitive by Citrix ADC policies.
- The function must return a single value even if the extension language returns multiple values.
- Functions with a variable number of arguments are not supported.

How do policy extensions work?

The existing policies on a Citrix ADC appliance use an interpreter to evaluate the functions, which are imported in a policy extension file. When a user imports a new function in a policy extension file:

1. The extension file is validated for syntax and other conditions.
2. If the validation fails, the error is reported to the user.
3. If the validation succeeds, the extension file is imported to the Citrix ADC appliance and its contents can be used in policy expressions, just like any built-in policy function
    a) If the policy expression evaluation returns an error during runtime, it is reported as an undef event and the associated error counter is incremented.
    Note: If a policy undef event occurs and the policy rule contains one or more policy extension functions, the `show ns extension <name>` command displays the undef hits when applied to those policy extensions. If the extension function is aborted, the abort counter value is incremented.
    b) If the policy expression evaluation is successful, expression evaluation resumes until the entire expression is evaluated, or until it is aborted because of an error.

If the extension function takes too long to run, it is aborted, and the error counter pertaining to that extension function is incremented. The extension function is sandboxed, which prevents:

- Excessive CPU usage on the Citrix ADC appliance.
- Excessive memory usage on the Citrix ADC appliance.
- Usage of harmful built-in libraries or third-party libraries or binaries.
- Long-running scripts that could potentially cause the Citrix ADC appliance to reboot.

Configuring policy extensions

September 14, 2021

When your policy extension file is ready, import it to the Citrix ADC appliance. The import process copies the extension file into a directory on the Citrix ADC appliance and checks for syntax errors.

After the import, you have to make the extension file available for use in the policy expressions.

Note: The import command is used to download the file content from an external source \<src\>, or an internal source, onto the Citrix ADC file system. To load this file content into one or more packet engines for the first time, use the add command. If there is an update to the file content, the updated content can be downloaded to the Citrix ADC file system by issuing the import command with the overwrite argument. The command updates the contents in the file system. To load the updated content to one or more packet engines, use the update command.

Configure policy extensions by using the CLI

1. Import the policy extension file to the Citrix ADC Appliance, from either a web server (using HTTP) or your local workstation.

   a) HTTP Import

   If you have a web server available, you can store the extension file in the webserver directory and import it to the Citrix ADC appliance.

   ```
   import ns extension <src> <name> [-comment<string>] [-
      overwrite]
   <!--NeedCopy-->
   ```

   Example:

   ```
   import ns extension http://myhost/path/to/extension
      myextension -comment  “Custom crc calculation
   <!--NeedCopy-->
   ```

   b) Local Import

You can use the SSH client to copy the extension file from your workstation to the /var/tmp directory of the Citrix ADC appliance

```
scp extension-file-name <ns-userid@ns-ip-addr>:/var/tmp
<!--NeedCopy-->
```

where,

- extension-file-name is the name of the extension file on your client machine.
- ns-userid is the Citrix ADC appliance user with permission to write to /var/tmp.
- ns-ip-addr is the Citrix ADC IP address.

After copying the file to the Citrix ADC appliance, run the import command on the Citrix ADC appliance.

```
import ns extension local:\<extension-file-name extension-name
   \>
<!--NeedCopy-->
```

Note: The CLI must be used to import a local extension file, by running the **import** command.

2. Add the policy extension to the packet engine for evaluation.

```
add ns extension <name> [-comment <string>]
<!--NeedCopy-->
```

Example:

```
add ns extension myextension
<!--NeedCopy-->
```

After an extension file is imported, you can update it, if you included the -overwrite parameter in the import command, or remove it. You can also display the details of an imported extension file.

Update an extension file on the Citrix ADC appliance from the source

At the command prompt, type:

```
update ns extension <name>
<!--NeedCopy-->
```

Note: You can update the extension file only after importing the specified extension file to the Citrix ADC appliance with the -overwrite parameter.

Example:


```
update ns extension myextension
<!--NeedCopy-->
```

Remove an extension file from the Citrix ADC appliance

At the command prompt type:

```
rm ns extension <name>
<!--NeedCopy-->
```

Example:

```
rm ns extension myextension
<!--NeedCopy-->
```

Display the details of the specified extension function on the Citrix ADC appliance

At the command prompt, type:

```
show ns extension <name>
<!--NeedCopy-->
```

Example:

```
show ns extension myextension
<!--NeedCopy-->
```

Configure policy extensions by using the GUI

1. Import the policy extension file to the Citrix ADC Appliance, from either a web server (using HTTP) or your local workstation.
   a) Navigate to **AppExpert** > **Policy Extensions**, click **Policy Extension**, from the **Import From** drop-down list, select the URL for the location of the extension file that you want to import.
   b) Navigate to **AppExpert** > **Policy Extensions**, **Policy Extension** and import the extension file by selecting File in the **Import** From drop-down list.
2. Add the policy extension to the packet engine for evaluation.

   Navigate to **AppExpert** > **Policy Extensions** and, on the**Policy Extensions** tab, add the extension file.

Update an extension file on the Citrix ADC appliance from the source

Navigate to **AppExpert** > **Policy Extensions** and, on the **Policy Extensions** tab, update the extension file.

Remove an extension file from the Citrix ADC appliance

Navigate to **AppExpert** > **Policy Extensions** and, **Policy Extensions** tab, remove the extension file.

Display the details of the specified extension function on the Citrix ADC appliance

Navigate to **AppExpert** > **Policy Extensions** and, on the **Policy Extensions Functions** tab, click the click drop-down list arrow of the extension function that you want to see the details.

Policy extensions - use cases

September 14, 2021

Certain customer applications have requirements that cannot be addressed with existing policies and expressions. The policy extension feature enables customers to add customized functions to their applications to meet their requirement.

The following use cases illustrate the addition of new functions using the policy extension feature on the Citrix ADC appliance.

- Case 1: Custom hash
- Case 2: Collapse double slashes in URLs
- Case 3: Combine headers

Case 1: Custom hash

The CUSTOM_HASH function provides a mechanism to insert any type of hash value in the responses sent to the client. In this use case, the hash function is used to compute the hash of the query string for a rewrite HTTP request and insert an HTTP header named CUSTOM_HASH with the computed value. The CUSTOM_HASH function implements the DJB2 hash algorithm.

Sample Usage of CUSTOM_HASH:

```
1 > add rewrite action test_custom_hash insert_http_header "CUSTOM_HASH"
     "HTTP.REQ.URL.QUERY.CUSTOM_HASH"
2 <!--NeedCopy-->
```


Sample Definition of CUSTOM_HASH():

```
    -- Extension function to compute custom hash on the text

    -- Uses the djb2 string hash algorithm
    function NSTEXT:CUSTOM_HASH() : NSTEXT

        local hash = 5381

        local len = string.len(self)

        for i = 1, len do

             hash = bit32.bxor((hash * 33), string.byte(self, i))

        end

        return tostring(hash)

    end
<!--NeedCopy-->
```

Line-by-line description of the above sample:

```
function NSTEXT:CUSTOM_HASH() : NSTEXT

Defines the CUSTOM_HASH() function, with text input and a text return
   value.

local hash = 5381
local len = string.len(self)

Declares two local variables:

 - hash. Accumulates the compute hash value and is seeded with the
    number 5381

 - len. Sets to the length of the self input text string, using the
    built-in string.len() function.

for i = 1, len do
      hash = bit32.bxor((hash * 33), string.byte(self, i))
end

Iterates through each byte of the input string and adds the byte to the
```



```
        hash. It uses the built-in string.byte() function to get the byte
       and the built-in bit32.bxor() function to compute the XOR of the
       existing hash value (multiplied by 33) and the byte.
19
20 return tostring(hash)
21
22 Calls the built-in tostring() function to convert the numeric hash
       value to a string and returns the string as the value of the
       function.
23 <!--NeedCopy-->
```

Case 2: Collapse double slashes in URLs

Collapsing double slashes in URLs improves the website rendering time, because browsers parse the single slash URLs more efficiently. The single slash URLs also to maintain compatibility with applications that do not accept double slashes. The policy extension feature allows customers to add a function that replaces the double slashes with single slashes in the URLs. The following example illustrates the addition of a policy extension function that that collapses double slashes in URLs.

Sample Definition of COLLAPSE_DOUBLE_SLASHES():

```
1      -- Collapse double slashes in URL to a single slash and return the
          result
2      function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT
3
4          local result = string.gsub(self, "//", "/")
5
6          return result
7
8      end
9  <!--NeedCopy-->
```

Line-by-line description of the above sample:

```
1  function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT
2
3  Declares the COLLAPSE_DOUBLE_SLASHES() function with text input and
       return.
4
5  local result = string.gsub(self, "//", "/")
6
7  Declares a local variable named result and uses the built-in string.
       gsub() function to replace all double slashes with single slashes in
        the self input text.
```



```
8
9  The second parameter of string.gsub() is actually a regular expression
       pattern, although here a simple string is used for the pattern.
10
11 return result
12
13 Returns the resulting string.
14 <!--NeedCopy-->
```

Case 3: Combine headers

Certain customer applications cannot handle multiple headers in a request. Also, parsing of duplicate headers with same header values, or multiple headers with same name but different values in a request, consumes time and network resources. The policy extension feature allows customers to add a function to combine these headers into single headers with a value combining the original values. For example, combining the values of the headers H1 and H2.

Original request:

```
1  GET /combine_headers HTTP/1.1
2  User-Agent: amigo unit test
3  Host: myhost
4  H2: h2val1
5  H1: abcd
6  Accept: */*
7  H2: h2val2
8  Content-Length: 0
9  H2: h2val3
10 H1: 1234
11 <!--NeedCopy-->
```

Modified request:

```
1  GET /combine_headers HTTP/1.1
2  User-Agent: amigo unit test
3  Host: myhost
4  H2: h2val1, h2val2, h2val3
5  H1: abcd, 1234
6  Accept: */*
7  Content-Length: 0
8  <!--NeedCopy-->
```

In general, this type of request modification is done using the Rewrite feature, using policy expressions to delineate the part of the request to be modified (the target) and the modification to be performed

(the string builder expression). However, policy expressions do not have the ability to iterate over an arbitrary number of headers.

The solution to this problem requires an extension to the policy facility. To do this, we will define an extension function, called COMBINE_HEADERS. With this function, we can set up the following rewrite action:

```
> add rewrite action combine_headers_act replace 'HTTP.REQ.FULL_HEADER
.AFTER_STR("HTTP/1.1rn")''HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1rn").
COMBINE_HEADERS'
```

Here, the rewrite target is HTTP.REQ.FULL_HEADER.AFTER_STR(“HTTP/1.1rn”). The AFTER_STR(“HTTP/1.1rn”) is required because FULL_HEADER includes the first line of the HTTP request (e.g. GET /combine_headers HTTP/1.1).

The string builder expression is HTTP.REQ.FULL_HEADER.AFTER_STR(“HTTP/1.1rn”).COMBINE_HEADERS, where the headers (minus the first line) are fed into the COMBINE_HEADERS extension function, which combines and returns the values for headers.

Sample Definition of COMBINE_HEADERS():

```
1     -- Extension function to combine multiple headers of the same name
         into one header.
2
3
4
5     function NSTEXT:COMBINE_HEADERS(): NSTEXT
6
7         local headers = {
8   }
9   -- headers
10
11        local combined_headers = {
12  }
13  -- headers with final combined values
14        -- Iterate over each header (format "name:valuer\r\n")
15
16        -- and build a list of values for each unique header name.
17
18        for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n"
             ) do
19
20            if headers[name] then
21
22                local next_value_index = #(headers[name]) + 1
23
```



```
                headers[name][next_value_index] = value

            else

                headers[name] = {
 name .. ":" .. value }


            end

        end



        -- iterate over the headers and concat the values with
           separator ","

        for name, values in pairs(headers) do

            local next_header_index = #combined_headers + 1

            combined_headers[next_header_index] = table.concat(values,
               ",")

        end



        -- Construct the result headers using table.concat()

        local result_str = table.concat(combined_headers, "\r\n") .. "\
           r\n\r\n"

        return result_str

    end
<!--NeedCopy-->
```

Line-by-line description of the above sample:

```
function NSTEXT:COMBINE_HEADERS(): NSTEXT

Defines the COMBINE_HEADERS extension function, with the text input
   into the function from the policy expression and a text return type
   to the policy expression.
```



```
4
5  local headers = {
6    }
7    -- headers
8  local combined_headers = {
9    }
10   -- headers with final combined values
11
12 Declares local variables headers and combined_headers and initialize
       these variables to empty tables. headers will be a table of arrays
       of strings, where each array holds one or more values for a header.
       combined_headers will be an array of strings, where each array
       element is a header with its combined values.
13
14 for name, value in string.gmatch(self, "([^:]+):([^\r\n\]*)\r\n") do
15 . . .
16 end
17 <!--NeedCopy-->
```

This generic for loop parses each header in the input. The iterator is the built-in string.gmatch() function. This function takes two parameters: a string to search, and a pattern to use to match pieces of the string. The string to search is supplied by the implicit self parameter, which is the text for the headers input into the function.

The pattern is expressed using a regular expression (regex for short). This regex matches the header name and value for each header, which the HTTP standard defines as `*name*:*value*\r\n`. The parentheses in the regex specify the matching parts to be extracted, so the regex schematic is (*match-name*):(*match-value*)\r\n. The *match-name* pattern needs to match all characters except the colon. This is written [^:]+([^:] is any character except : and + is one or more repetitions). Similarly, the *match-value* pattern has to match any characters except the \r\n, so it is written [^\r\n] ([^\r\n] matches any character except \r and \n and * is zero or more repetions). This makes the complete regex `([^:]+):([^\r\n]*)\r\n`.

The for statement uses a multiple assignment to set name and value to the two matches returned by the string.gmatch() iterator. These are implicitly declared as local variables within the body of the for loop.

```
1  if headers[name] then
2       local next_value_index = #(headers[name]) + 1
3       headers[name][next_value_index] = value
4  else
5       headers[name] = {
6   name .. ":" .. value }
7
```



```
8 end
9 <!--NeedCopy-->
```

These statements within the for loop put the header names and values into the headers table. The first time a header name is parsed (say H2: h2val1 in the example input), there is no headers entry for the name and headers[name] is nil.

Since nil is treated as false, the else clause is executed. This sets the headers entry for name to an array with one string value *name*:*value*.

Note: The array constructor in the else loop is equivalent to {[1] = name .. ":" .. value}, which sets the first element of the array.) For the first H2 header, it sets headers["H2"] = {"H2:h2val1"}.

On subsequent instances of a header, (say, H2: h2val2 in the example input). headers[name] is not nil, so the then clause is executed. This determines the next available index in the array value for headers[name], and puts the header value into that index. For the second H2 header, it sets headers["H2"] = {"H2:h2val1", "h2val2"}.

```
1 for name, values in pairs(headers) do
2     local next_header_index = #combined_headers + 1
3     combined_headers[next_header_index] = table.concat(values, ",")
4 end
5 <!--NeedCopy-->
```

After the original headers have been parsed and the headers table filled in, this loop builds the combined_headers array. It uses the pairs() function as the for loop iterator.

Each call to pairs() returns the name and value of the next entry in the headers table.

The next line determines the next available index in the combined_headers array, and the next line sets that array element to the combined header. It uses the built-in table.concat() function, which takes as its arguments an array of strings and a string to use as a separator, and returns a string that is the concatenation of the array strings, separated by the separator.

For example, for values = {"H2:h2val1", "h2val2"}, this produces "H2:h2val1, h2val2"

```
1 local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
2 <!--NeedCopy-->
```

After the combined_headers array are built, it concatenates the elements into one string, and adds a double \r\n that terminates the HTTP headers.

```
1 return result_str
2 <!--NeedCopy-->
```

Returns a string as the result of the COMBINE_HEADERS extension function.

Troubleshooting policy extensions

September 14, 2021

If your extension function is not behaving as expected, you can use extension tracing functionality to verify the behavior of your extension function. You can also add logging to your extension function by using the custom logging functionality, where you can define the log level to be captured on the Citrix ADC appliance.

This topic provides information on:

- Extension tracing
- Custom logging

Extension tracing

To show what your extension function is doing, extension tracing functionality logs the execution of the function to the Citrix ADC system log (/var/log/ns.log). The trace logging uses the DEBUG log level, which normally is not enabled. Therefore, you have to enable ALL log levels. Then you can enable tracing by setting the -trace option of the set ns extension command. The available settings are:

- off turn off tracing (equivalent to unset ns extension -trace).
- calls trace function calls with arguments and function returns with the first return value.
- lines trace the above plus line numbers for executed lines.
- all trace the above plus local variables changed by executed lines.

Example:

```
set audit syslogParams -loglevel ALL

set ns extension combine_headers -trace all
<!--NeedCopy-->
```

Each trace message has the format

```
log-header : default NSEXTENSION Message message-number 0 : "TRACE function
-name CALL call-number: event"
```

Where,

- log-header supplies timestamps, the Citrix ADC IP address, and the Packet Engine ID.
- message-number is a sequential number identifying the log message.
- function-name is the extension function name.

- call-number is a sequential number for each extension function call. It can be used to group all the trace messages for an extension function call.
- event is one of the following:
  - CALL function-name ; parameter-values indicates that the function has been called with the specified parameters.
  - RETURN FROM function-name ; return = value indicates that a function has returned the specified (first) value. (Additional return values are not reported.)
  - LINE line-number ; variable-values indicates that a line has been executed and lists any variables with changed values.

Where,

- value or values is
  - a number, with or without a decimal point,
  - a string, enclosed in double quotes and with escaped characters as described earlier,
  - a boolean true or false,
  - nil,
  - a table constructor, of the format {[key1]=value1,[key2]=value2, …}.
- parameter-values is parameter1 = value1 ; parameter2 = value2 , …
- variable-values is variable1 = value1 ; variable2 = value2 , …

An example of abbreviated log messages:

```
>shell tail -f /var/log/ns.log | grep TRACE | more

... NSEXTENSION Message 3035 0 : "TRACE combine_headers CALL 30 : CALL
    COMBINE_HEADERS; self = "User-Agent: curl/7.24.0 (amd64-portbld-
    freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nHost:
    10.217.24.7\r\nAccept: */*\r\nH2: h2val1\r\nH1: abcd\r\nH2: h2val2\r
    \nH2: h2val3\r\n\r\n""

... NSEXTENSION Message 3036 0 : "TRACE combine_headers CALL 30 : LINE
    4; headers = {
  }
 "

... NSEXTENSION Message 3037 0 : "TRACE combine_headers CALL 30 : LINE
    5; combined_headers = {
  }
 "

... NSEXTENSION Message 3038 0 : "TRACE combine_headers CALL 30 : CALL
    gmatch"

```



```
15 ... NSEXTENSION Message 3039 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM gmatch; return = function 0x2bee5a80"
16
17 ... NSEXTENSION Message 3040 0 : "TRACE combine_headers CALL 30 : CALL
      for iterator"
18
19 ... NSEXTENSION Message 3041 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM for iterator; return = " curl/7.24.0 (amd64-portbld-
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""
20
21 ... NSEXTENSION Message 3042 0 : "TRACE combine_headers CALL 30 : LINE
      9; name = "User-Agent"; value = " curl/7.24.0 (amd64-portbld-
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""
22
23 ... NSEXTENSION Message 3043 0 : "TRACE combine_headers CALL 30 : LINE
      10"
24
25 ... NSEXTENSION Message 3044 0 : "TRACE combine_headers CALL 30 : LINE
      14; headers = {
26  ["User-Agent"]={
27  [1]="User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
        OpenSSL/0.9.8y zlib/1.2.3" }
28   }
29  "
30
31 . . .
32
33 ... NSEXTENSION Message 3117 0 : "TRACE combine_headers CALL 30 : CALL
      for iterator"
34
35 ... NSEXTENSION Message 3118 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM for iterator; return = nil"
36
37 ... NSEXTENSION Message 3119 0 : "TRACE combine_headers CALL 30 : LINE
      19"
38
39 ... NSEXTENSION Message 3120 0 : "TRACE combine_headers CALL 30 : CALL
      concat"
40
41 ... NSEXTENSION Message 3121 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM concat; return = "User-Agent: curl/7.24.0 (amd64-portbld
      -freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
      nAccept: */*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3""
      ... NSEXTENSION Message 3122 0 : "TRACE combine_headers CALL 30 :
      LINE 25; result_str = "User-Agent: curl/7.24.0 (amd64-portbld-
```



```
      freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
      nAccept: */*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3\r\n\
      r\n""
42
43 ... NSEXTENSION Message 3123 0 : "TRACE combine_headers CALL 30 :
      RETURN FROM COMBINE_HEADERS; return = "User-Agent: curl/7.24.0 (
      amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r
      \nH1: abcd\r\nAccept: */*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2
      , h2val3\r\n\r\n""
44 <!--NeedCopy-->
```

Custom logging

You can also add your own logging to your extension function. To do so, use the built-in ns.logger:*level*() function, where *level* is emergency, alert, critical, error, warning, notice, info, or debug. The parameters are the same as the C printf() function: a format string, and a variable number of arguments to supply values for the % specified in the format string. For example, you might add the following to the COMBINE_HEADERS function to log the result of a call:

```
1 local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
2
3 ns.logger:info("Result: %s", result_str)
4
5 return result_str
6 <!--NeedCopy-->
```

The above function would log the following message to/var/log/ns.log for the sample input shown in the abbreviated log messages examples in the Extension Tracing section above.

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M
 H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)libcurl
/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: */*^M H2: h2val1, h2val2,
h2val3^M ^M"
```

Optimization

September 14, 2021

The Citrix ADC optimization features reduce transaction times between the clients and the servers, and they reduce bandwidth consumption. They also enhance server performance by offloading some tasks and making others more efficient.

Feature	Description
Client Keep-Alive	Handles multiple requests on a single client connection. The client does not have to negotiate a new connection for each request to the server.
HTTP Compression	Compresses HTTP responses sent from the servers to compression-aware browsers. The smaller responses reduce download time and save bandwidth.
Integrated Caching	Stores responses to client requests. Subsequent requests for the same content are served from the Citrix ADC cache instead of being forwarded to the origin server.
Front End Optimization	Reduces the load and render time of web pages by simplifying and optimizing the content served to the client browser. **Note**: Supported from NetScaler 10.5 onwards.
Content Accelerator	Stores server responses on a Citrix ByteMobile T2100 appliance. **Note**: Supported from NetScaler 10.1 onwards.
SPDY (Speedy)	Acts as a SPDY gateway between clients and your servers, providing SPDY support without the need to configure/upgrade SPDY on the servers. **Note**: Supported from NetScaler 10.1 onwards.

Client keep-alive

September 14, 2021

The client keep-alive feature enables multiple clients requests to be sent on a single connection. This feature benefits transaction management. When the Client Keep-Alive mode is enabled on an appliance and the server response to the client request contains the Connection: close the HTTP header and performs the following tasks:

- Renames the existing Connection header name by shuffling the characters in the header name.

- Adds a new Connection: header with Keep-Alive as the value for the header.

The Client Keep-Alive mode enables the Citrix ADC appliance to process multiple requests and responses using the same socket connection. The feature keeps the connection between the client and the appliance (client-side connection) open even after the server closes the connection with the appliance. This allows multiple clients requests using a single connection and saves the round trips associated in opening and closing a connection. Client keep-alive is most beneficial in SSL sessions.

Client keep-alive is useful for the following scenarios:

- If the server does not support the client keep-alive.
- If the server supports but an application on the server does not support the client keep-alive.

> **Note:**
> Client keep-alive is applicable for HTTP and SSL traffic. Client-keep alive can be configured globally to handle all traffic. Also, you can activate it on specific services.

In the client keep-alive environment, the configured services intercept the client traffic and the client request is directed to the origin server. The server sends the response and closes the connection between the server and the appliance. If a "Connection: Close" header is present in the server response, the appliance corrupts this header in the client-side response, and the client-side connection is kept open. As a result, the client does not have to open a new connection for the next request. Instead, the connection to the server is reopened.

> **Note:**
> If a server sends back two "Connection: Close" headers, only one is edited. This results in significant delays on the client rendering of the object because a client does not assume that the object has been delivered completely until the connection is closed.

Configure client keep-alive

Client keep-alive, by default, is disabled on the Citrix ADC, both globally and at service level. Therefore, you must enable the feature at the required scope.

> **Note:**
> If you enable the client keep-alive globally, it is enabled for all services, regardless of whether you enable it at the service level. Also, you must configure some HTTP parameters to specify the following:
>
> - the maximum number of HTTP connections retained in the connection reuse pool.
> - enable connection multiplexing, and enable persistence `Etag`.
>
> **Note:**
> When Persistent `ETag` is enabled, the `ETag` header includes information about the server that

> served the content. This ensures that cache validation conditional requests or browser requests, for that content, always reaches the same server.

Configure client keep-alive by using Citrix ADC command interface

At the command prompt, do the following:

1. Enable client keep-alive on the Citrix ADC.
   - At global level - `enable ns mode cka`
   - At service level - `set service <name> -CKA YES`

   > **Note:**
   >
   > Client keep-alive can be enabled only for HTTP and SSL services.

2. Configure HTTP parameters on the HTTP profile that is bound to one or more services.

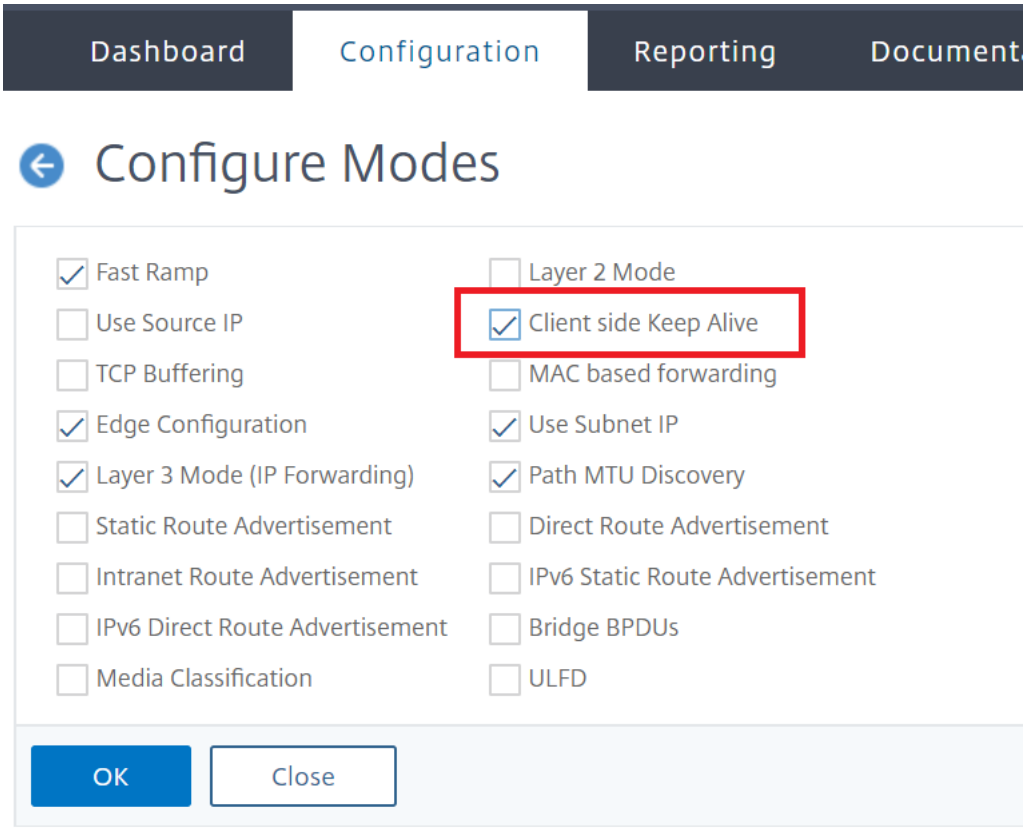
```
set ns httpProfile <name> -maxReusePool <value> -conMultiplex
    ENABLED -persistentETag ENABLED
<!--NeedCopy-->
```

> **Note:**
>
> Configure these parameters on the `nshttp_default _profile HTTP` profile, to make them available globally.

Configure client keep-alive by using Citrix ADC GUI

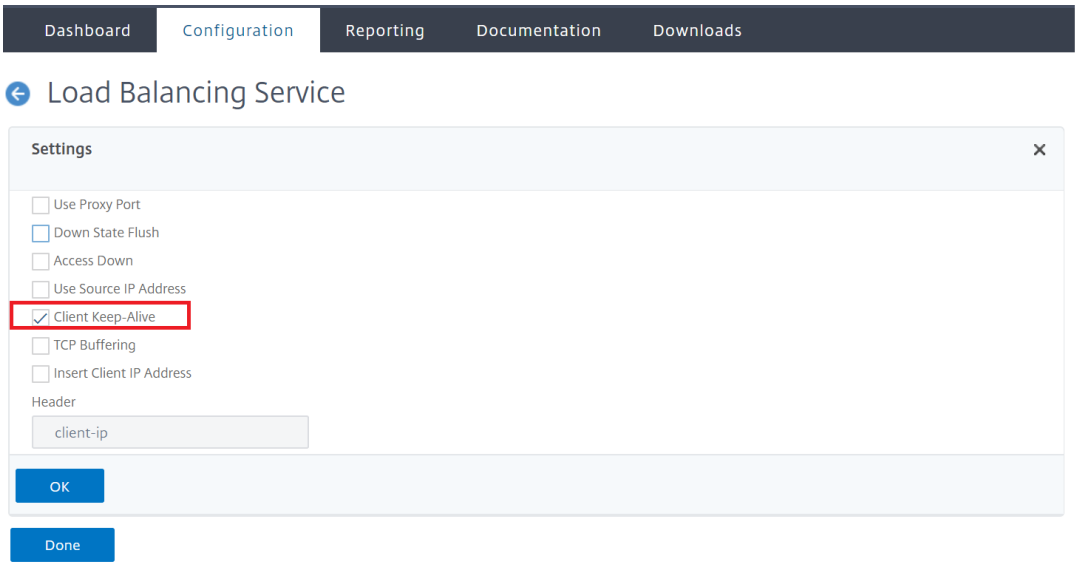
1. Enable client keep-alive on the Citrix ADC.
   - At global level

     Navigate to **System** > **Settings**, click **Configure Modes** and select **Client side Keep Alive**.



- At service level

  Navigate to **Traffic Management** > **Load Balancing** > **Services**, and select the required service. In the **Settings** section, select **Client Keep-Alive** check box.



2. Configure the required HTTP parameters on the HTTP profile that is bound to one or more ser-

vices.

3. Navigate to **System** > **Profiles**, and on **HTTP Profiles** tab, select the required profile and update the required HTTP parameters.

HTTP compression

September 14, 2021

For websites with compressible content, the HTTP compression feature implements lossless compression to alleviate latency, long download times, and other network-performance problems by compressing the HTTP responses sent from servers to compression-aware browsers. You can improve server performance by offloading the computationally intensive compression task from your servers to the Citrix ADC appliance.

The following table describes the capabilities of the HTTP compression feature:

Functionality	Description
Compression Ratio	Compression ratio depends on the types of files in the responses, but is always significant, noticeably reducing the amount of data transmitted over the network.
Browser Awareness	Citrix ADC serves compressed data to compression aware browsers only, reducing the transaction time between the client and the server. Most modern web browsers support HTTP compression.
Compression blocking	You can define content filters to selectively block compression by applying built-in actions.
Compression Caching	With the integrated caching feature enabled, subsequent requests for the same content are served from the local cache, reducing the number of round trips to the server and improving transaction times.
HTTPS Support	Compression is useful on SSL connections, because it reduces the amount of content that has to be encrypted, either on the server or by the Citrix ADC appliance, and decrypted by the client.

Functionality	Description
Intelligent Response Filtering	The Citrix ADC compression engine intelligently filters server responses based on defined compression parameters. For example, the compression engine detects zero-content-length responses and compressed responses and does not compress them. The detection of compressed responses enables origin sites to use server-based compression with the Citrix ADC compression feature.
Compression Switching	The Citrix ADC appliance transparently directs requests from compression aware clients to compression capable servers, so that responses to those clients are compressed, and responses to other clients are not delayed by compression processing.

How HTTP compression works

A Citrix ADC can compress both static and dynamically generated data. It applies the GZIP or the DEFLATE compression algorithm to remove extraneous and repetitive information from the server responses and represent the original information in a more compact and efficient format. This compressed data is sent to the client's browser and uncompressed as determined by the browser's supported algorithm or algorithms (GZIP or DEFLATE).

Citrix ADC compression treats static and dynamic content differently.

- Static files are compressed only once, and a compressed copy is stored in local memory. Subsequent client requests for cached files are serviced from that memory.
- Dynamic pages are dynamically created each time a client requests them.

When a client sends a request to the server:

1. The client request arrives at the Citrix ADC. The ADC examines the headers and stores information about what kind of compression, if any, the browser supports.
2. The ADC forwards the request to the server and receives the response.
3. The Citrix ADC compression engine examines the server response for compressibility by matching it against policies.
4. If the response matches a policy associated with a compression action, and the client browser

supports a compression algorithm specified by the action, the Citrix ADC applies the algorithm and sends the compressed response to the client browser.

5. The client applies the supported compression algorithm to decompress the response.

Configure HTTP compression

By default, compression is disabled on the Citrix ADC. You must enable the feature before configuring it. If the feature is enabled, the ADC compresses server requests specified by compression policies.

To enable HTTP compression by using the CLI

Compression can be enabled for HTTP and SSL services only. You can enable it globally, so that it applies to all HTTP and SSL services, or you can enable it just for specific services.

At the command prompt, enter one of the following commands to enable compression globally or for a specific service:

- `enable ns feature cmp`
  OR
- `set service \<name\> -CMP YES`

To configure compression by using the GUI

Do one of the following:

To enable compression globally, navigate to System > Settings, click **Configure Basic Features**, and select HTTP Compression.

To enable compression for a specific service, navigate to **Traffic Management > Load Balancing > Services**, select the service, and click Edit. In the Settings group, click the pencil icon and enable Compression.

Configuring a compression action

A compression action specifies the action to take when a request or response matches the rule (expression) in the policy with which the action is associated. For example, you can configure a compression policy that identifies requests that will be sent to a particular server, and associate the policy with an action that compresses the server's response.

There are four built-in compression actions:

- COMPRESS: Uses the GZIP algorithm to compress data from browsers that support either GZIP or both GZIP and DEFLATE. Uses the DEFLATE algorithm to compress data from browsers that support only the DEFLATE algorithm. If the browser does not support either algorithm, the browser's response is not compressed.
- NOCOMPRESS: Does not compress data.

- GZIP: Uses the GZIP algorithm to compress data for browsers that support GZIP compression. If the browser does not support the GZIP algorithm, the browser's response is not compressed.
- DEFLATE: Uses the DEFLATE algorithm to compress data for browsers that support the DEFLATE algorithm. If the browser does not support the DEFLATE algorithm, the browser's response is not compressed. After creating an action, you associate the action with one or more compression policies.

At the command prompt, enter the following command to create a compression action:

```
add cmp action <name> <cmpType> [-addVaryHeader <addVaryHeader> -varyHeaderValue
<string>]
```

To configure a compression policy by using the CLI

A compression policy contains a rule, which is a logical expression that enables the Citrix ADC appliance to identify the traffic that should be compressed.

When the Citrix ADC receives an HTTP response from a server, it evaluates the built-in compression policies and any custom compression policies to determine whether to compress the response and, if so, the type of compression to apply. Priorities assigned to the policies determine the order in which the policies are matched against the requests.

At the command prompt, enter the following command to create a compression policy:

```
add cmp policy <name> -rule <expression> -resAction <string>
```

To create a compression action by using the GUI

Navigate to **Optimization** > **HTTP Compression** > **Actions** , click **Add** , and create a compression action to specify the type of compression to be performed on the HTTP response.

Configuring a compression policy

A compression policy contains a rule, which is a logical expression that enables the Citrix ADC appliance to identify the traffic that should be compressed.

When the Citrix ADC receives an HTTP response from a server, it evaluates the built-in compression policies and any custom compression policies to determine whether to compress the response and, if so, the type of compression to apply. Priorities assigned to the policies determine the order in which the policies are matched against the requests.

The following table lists the built-in HTTP compression policies. These policies are activated globally when you enable compression.

Built-in Classic or Default Syntax Policy	Description
ns_nocmp_mozilla_47, ns_adv_nocmp_mozilla_	Prevents compression of CSS files when a request is sent from a Mozilla 4.7 browser.
ns_cmp_mscss, ns_adv_cmp_mscss	Compresses CSS files when the request is sent from a Microsoft Internet Explorer browser.
ns_cmp_msapp, ns_adv_cmp_msapp	Compresses files that are generated by the following applications: Microsoft Office Word, Microsoft Office Excel, Microsoft Office PowerPoint.
ns_cmp_content_type, ns_adv_cmp_content_type	Compresses data when the response containsContent-Type header and contains text.
ns_nocmp_xml_ie, ns_adv_nocmp_xml_ie	Prevents compression when a request is sent, from a Microsoft Internet Explorer browser and the response contains a Content-Type header and contains text or xml.

Binding a compression policy

To put a compression policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the Citrix ADC, or to a specific virtual server, so that the policy applies only to requests whose destination is the VIP address of that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

To bind a Compression Policy by using the CLI

At the command prompt, enter one of the following commands to bind a compression policy globally or to a specific virtual server:

- `bind cmp global <policyName> [-priority <positive_integer>] [-state ( ENABLED|DISABLED)]...`
- `bind lb vserver <vserverName> -policyName <policyName> -priority < positive_integer>.`

Repeat this command for each virtual server to which you want to bind the compression policy.

To bind a compression policy by using the GUI

Do one of the following:

At global level Navigate to **Optimization** > **HTTP Compression** > **Policies**, click **Policy Manager** and bind the required policies by specifying the relevant Bind Point and Connection Type (Request/Response).

At virtual server level

For load balancing virtual server, Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select the required virtual server, click **Policies**, and bind the relevant policy.

For content switching virtual server, Navigate to **Traffic Management > Content Switching > Virtual Servers**, select the required virtual server, click **Policies**, and bind the relevant policy.

Set the Global Compression Parameters for Optimal Performance

Many users accept the default values for the global compression parameters, but you might be able to provide more effective compression by customizing these settings.

> **Note**
> After you configure the global compression parameters, you do not have to reboot your appliance. They get applied to the new flows immediately.

The following table describes the compression parameters that you can set on the Citrix ADC.

Compression Parameters	Description
Quantum size	Size, in KB, of the buffer maintained for accumulating server responses. The responses are compressed when the buffer size exceeds this value. For example, if you set the quantum size to 50 KB, the Citrix ADC compresses the buffer's contents when its size becomes larger than 50 KB. Minimum value: 1. Maximum value: 63488. Default: 57344.
Compression level	Level of compression to apply to server responses. Possible values: Best Speed, Best Compression, optimal.
Minimum HTTP response size	Minimum size, in bytes, of an HTTP response that is compressed. Responses smaller than the value specified by this parameter are sent without being compressed.
Bypass compression on CPU usage	Citrix ADC CPU usage, as a percentage, at or above which no compression is done. Default: 100.

Compression Parameters	Description
Policy Type*	Type of policies used for compression. Possible values: Classic, Default Syntax. Default: Classic.
Allow Server-side compression	Allow servers to send compressed data to the Citrix ADC.
Compress push packet	Upon receipt of a packet with a TCP PUSH flag, compress the accumulated packets immediately, without waiting for the quantum buffer to be filled.
External Cache	Issue a private response directive indicating that the response message is intended for a single user and must not be cached by a shared or proxy cache.

To configure HTTP compression by using the GUI

Do one of the following:

- To enable compression globally, navigate to **System** > **Settings**, click **Configure Basic Features**, and select **HTTP Compression**.
- To enable compression for a specific service, navigate to **Traffic Management** > **Load Balancing** > **Services**, select the service, and click **Edit**.
- In the **Settings** group, click the pencil icon and enable **Compression**.

To create a compression action by using the GUI

Navigate to **Optimization** > **HTTP Compression** > **Actions** , click **Add** , and create a compression action to specify the type of compression to be performed on the HTTP response

To create a compression policy by using the GUI

Navigate to **Optimization** > **HTTP Compression** > **Policies** , click **Add**, and create a compression policy by specifying the condition and the corresponding action to be run.

Evaluate compression configuration

You can view the compression statistics in the dashboard utility or in an SNMP monitor. The dashboard utility displays summary and detailed statistics in a tabular and graphic format.

Optionally, you can also view statistics for a compression policy, including the number of requests that the policy counter increments during the policy based compression.

> **Note**
>
> - For more information about the statistics and charts, see the Dashboard help on the Citrix ADC appliance.
> - For more information about SNMP, see SNMP topic.

To view compression statistics by using the CLI

At the command prompt, enter the following commands to display the compression statistics:

1. To display compression statistics summary.

```
stat cmp
```

> **Note**
> The stat cmp policy command displays statistics for default syntax compression policies only.

2. To display compression policy hits and details

```
show cmp policy \<name\>
```

3. To display detailed compression statistics

```
stat cmp -detail
```

To view compression statistics by using the dashboard:

In the Dashboard utility, you can display the following types of compression statistics:

- Select Compression to display a summary of the compression statistics.
- To display detailed compression statistics by protocol type, click the Details
- To display the rate of requests processed by the compression feature, click the Graphical View tab.

To view compression statistics by using SNMP

You can view the following compression statistics by using the SNMP network management application.

- Number of compression requests (OID: 1.3.6.1.4.1.5951.4.1.1.50.1)
- Number of compressed bytes transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.2)
- Number of compressible bytes received (OID: 1.3.6.1.4.1.5951.4.1.1.50.3)
- Number of compressible packets transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.4)
- Number of compressible packets received (OID: 1.3.6.1.4.1.5951.4.1.1.50.5)
- Ratio of compressible data received and compressed data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.6)
- Ratio of total data received to total data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.7)

To view more compression statistics by using the GUI

1. To display HTTP compression statistics:

Navigate to **Optimization** > **HTTP Compression** and click **Statistics**.

1. To display statistics of a compression policy.

Navigate to **Optimization** > **HTTP Compression** > **Policies** > select the policy, and click **Statistics**.

1. To display statistics of a compression policy label
2. Navigate to **Optimization** > **HTTP Compression** > **Policies** > select a policy label, and click **Statistics**.

Offloading HTTP compression

Performing compression on a server can affect the server's performance. A Citrix ADC placed in front of your web servers and configured for HTTP compression offloads compression of both static and dynamic content, saving server CPU cycles and resources.

You can offload compression from the Web servers in either of two ways:

Disable compression on the web servers, enable the Citrix ADC Compression feature at a global level, and configure services for compression.

Leave the compression feature enabled on the web servers and configure the Citrix ADC appliance to remove the "Accept Encoding" header from all HTTP client requests. The servers then send uncompressed responses. The Citrix ADC compresses the server responses before sending them to the clients.

> **Note**
> The second option does not work if the servers automatically compress all responses. The Citrix ADC does not attempt to compress a response that is already compressed.

The Servercmp parameter enables the Citrix ADC appliance to handle offload HTTP compression. By default, this parameter is set ON for the server to send compressed data to the Citrix ADC appliance. To offload HTTP compression, you need to set the servercmp parameter to OFF. At the command prompt, enter the following commands:

```
set service <service name> -CMP YES
```

Repeat this command for each service for which you want to enable compression.

```
show service <service name>
```

Repeat this command for each service, to verify that compression is enabled.

```
Save config
```

```
set cmp parameter -serverCmp OFF
```

> **Note:**
>
> When the `Servercmp` parameter is turned on and if the appliance receives compressed response from the server, the appliance does not further compress the data. Instead, it forwards the com-

> pressed response to the client.

Integrated caching

September 14, 2021

The integrated cache provides in-memory storage on the Citrix ADC appliance and serves Web content to users without requiring a round trip to an origin server. For static content, the integrated cache requires little initial setup. After you enable the integrated cache feature and perform basic setup (for example, determining the amount of Citrix ADC appliance memory the cache is permitted to use), the integrated cache uses built-in policies to store and serve specific types of static content, including simple webpages and image files. You can also configure the integrated cache to store and serve dynamic content that is marked as non-cacheable by Web and application servers (for example, database records and stock quotes).

> **Note:**
>
> The term Integrated Cache can be interchangeably used with AppCache; note that from a functionality point of view, both terms mean the same.

When a request or response matches the rule (logical expression) specified in a built-in policy or a policy that you have created. The Citrix ADC appliance performs the action associated with the policy. By default, all policies store cached objects in and retrieve them from the default content group. You can create your own content groups for different types of content.

To enable the appliance to find cached objects in a content group, you can configure selectors. The selectors match cached objects against expressions, or you can specify parameters for finding objects in the content group. If you use selectors as recommended by Citrix, configure them first, so that you can specify selectors when you configure content groups. Next, set up any content groups that you want to add, so that they are available when you configure the policies. To complete the initial configuration, create policy banks by binding each policy to a global bind point or a virtual server. Or, you can bind a label that can be called from other policy banks.

Integrated caching can be improved using pre-loading cached object method before they are scheduled to expire. To manage the handling of cached data, you can configure caching-related headers inserted into the responses. The integrated cache can also act as a forward proxy for other cache servers.

> **Note:**
>
> Integrated caching requires some familiarity with HTTP requests and responses.

> For information about the structure of HTTP data, see *Live HTTP Headers* at "<http://livehttpheaders.mozdev.org/>."

How integration cache works

The integrated cache monitors HTTP and SQL requests that flow through the Citrix ADC appliance and compares the requests with stored policies. Depending on the outcome, the integrated cache feature either searches the cache for the response or forwards the request to the origin server. For HTTP requests, integrated caching serves as partial content from the cache in response to single byte-range and multi-part byte-range requests.

Cached data is compressed if the client accepts compressed content. You can configure expiration times for a content group, and you can selectively expire entries in a content group.

Data that is served from the integrated cache is a request, and data served from the origin is a cache miss, as described in the following table.

Transaction Type	Specification
Cache Hit	Responses that the Citrix ADC appliance serves from the cache, including: Static objects, for example, image files and static webpages, 200 OK pages, 203 Non-Authoritative Response pages, 300 Multiple Choices pages, 301 Moved Permanently pages, 302 Found pages, 304 Not Modified pages, These responses are known as positive responses. The Citrix ADC appliance also caches the following negative responses: 307 Temporary Redirect pages, 403 Forbidden pages, 404 Not Found pages, 410 Gone pages. To further improve performance, you can configure the Citrix ADC appliance to cache more types of content.
Storable Cache Miss	For a storable cache miss, the Citrix ADC appliance fetches the response from the origin server, and stores the response in the cache before serving it to the client.

Transaction Type	Specification
Non-Storable Cache Miss	A non-storable cache miss is inappropriate for caching. By default, any response that contains the following status codes is a non-storable cache miss: 201, 202, 204, 205, 206 status codes, All 4xx codes, except 403, 404 and 410, 5xx status codes

> **Note:**
> To integrate dynamic caching with your application infrastructure, use the NITRO API to issue cache commands remotely. For example, you can configure triggers that expire cached responses when a database table is updated.

To ensure the synchronization of cached responses with the data on the origin server, you configure expiration methods. When the Citrix ADC appliance receives a request that matches an expired response, it refreshes the response from the origin server.

> **Note:**
> Citrix recommends that you synchronize the times on the Citrix ADC appliance and one or more back-end servers.

How dynamic cache works

Dynamic caching evaluates HTTP requests and responses based on parameter-value pairs, strings, string patterns, or other data. For example, suppose that a user searches for Bug 31231 in a bug reporting application. The browser sends the following request on the user's behalf:

```
1     GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
         Template=view&TableId=1000
2
3     Host: mycompany.net
4
5     User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
          Gecko/2008052906 Firefox/3.0
6
7     Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
         =0.8
8
9     Accept-Language: en-us,en;q=0.5
10 <!--NeedCopy-->
```


In this example, GET requests for this bug reporting application always contain the following parameters:

- IssuePage
- RecordID
- Template
- TableId

GET requests do not update or alter the data, so you can configure these parameters in caching policies and selectors, as follows:

- You configure a caching policy that looks for the string mybugreportingsystem and the GET method in HTTP requests. This policy directs matching requests to a content group for bugs.
- In the content group for bugs, you configure a `hit` selector that matches various parameter-value pairs, including IssuePage, RecordID, and so on.

> **Note**
>
> A browser can send multiple GET requests based on one user action. The following is a series of three separate GET requests that a browser issues when a user searches for a bug based on a bug ID.

```
1     GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
         Template=view&TableId=1000
2
3     GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
         viewbtns&RecordId=31231&TableId=1000
4
5     GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
         viewbody&RecordId=31231&tableid=1000
6 <!--NeedCopy-->
```

To fulfill these requests, multiple responses are sent to the user's browser, and the webpage that the user sees is an assembly of the responses.

If a user updates a bug report, the corresponding responses in the cache must be refreshed with data from the origin server. The bug reporting application issues HTTP POST requests when a user updates a bug report. In this example, you configure the following to ensure that POST requests trigger invalidation in the cache:

- A request-time invalidation policy that looks for the string mybugreportingsystem and the POST HTTP request method, and directs matching requests to the content group for bug reports.
- An invalidation selector for the content group for bug reports that expires cached content based on the RecordID parameter. This parameter appears in all the responses, so the invalidation selector can expire all relevant items in the cache.

The following excerpt shows a POST request that updates the sample bug report.

```
POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\
    r\n

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
    Opera 7.23 [en]\r\n

Host: mybugreportingsystem\r\n

Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23
    unowned%3Afalse%23submitter%3Afalse%23incsub%3Atrue;

Cookie2: $Version=1\r\n

. . .

\r\n

ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=
    Update&CopyProjectId=0&ReloadForm=0&State=&RecordLockId=49873+
    issues+in+HTTP&F43. . .
<!--NeedCopy-->
```

When the Citrix ADC appliance receives this request, it does the following:

- Matches the request with an invalidation policy.
- Finds the content group that is named in the policy.
- Applies the invalidation selector for this content group and expires all responses that match RecordID=31231.

When a user issues a new request for this bug report, the Citrix ADC appliance goes to the origin server for updated copies of all the responses that are associated with the report instance. It stores the responses in the content group, and serves them to the user's browser, which reassembles the report and displays it.

Configure integrated cache

To use the integrated cache, you must install the license and enable the feature. After you enable the integrated cache, the Citrix ADC® appliance automatically caches static objects as specified by built-in policies and generates statistics on cache behavior. (Built-in policies have an underscore in the initial position of the policy name.)

Even if the built-in policies are adequate for your situation, you might want to modify the global attributes. For example, you might want to modify the amount of Citrix ADC appliance memory allo-

cated to the integrated cache.

If you would like to observe cache operation before changing settings, see "Displaying Cached Objects and Cache Statistics."

> **Note:**
>
> The Citrix ADC cache is an in-memory store that is purged when you restart the appliance.

To install integrated cache license

- An integrated cache license is required.
- Obtain a license code from Citrix, go to the command line interface, and log in.

At the command line interface, copy the license file to the `/nsconfig/license` folder.

- Reboot the Citrix ADC appliance by using the following command:

```
reboot
```

To enable integrated caching:
When you enable integrated caching, the Citrix ADC appliance begins caching server responses. If you have not configured any policies or content groups, the built-in policies store cached objects in the Default content group.

At the command prompt, type one of the following commands to enable or disable integrated caching:

```
enable ns feature IC
```

Configure global attributes for caching

Global attributes apply to all cached data. You can specify the amount of Citrix ADC memory allocated to the integrated cache, Via header insertion. A criterion for verifying that a cached object must be served. The maximum length of a POST body permitted in the cache, whether to bypass policy evaluation for HTTP GET requests, and an action to take when a policy cannot be evaluated.

The cache memory capacity is limited only by the memory of the hardware appliance. Also, any packet engine (central distribution hub of all incoming TCP requests) in the nCore Citrix ADC appliance is aware of objects cached by other packet engines in the nCore Citrix ADC appliance.

> **Note:**
>
> When the default global memory limit is set as 0 and the Integrated Caching (IC) feature is enabled, the appliance does not cache any objects. For caching, you must explicitly configure the global memory limit. However, if you enable "set authentication, authorization, and auditing parameter enableStaticPageCaching" option, there will be some default memory configured in the appliance. This memory is insufficient for caching large Objects and so it is necessary to assign a higher memory limit for IC. You can perform this by configuring the "set cache parameter

> –memLimit” command. The new setting is applied only after you save the configuration and reboot the appliance.

You can modify the global memory limit configured for caching objects. However, when you update the global memory limit to a value lower than the existing value (for example, from 10 GB to 4 GB), the appliance continues to use the memory limit.

It means that even though the integrated caching limit is configured to some value, the actual limit used can be higher. This excessive memory is however released when the objects are removed from cache.

The output of the show cache parameter command indicates the configured value (memory Usage limit) and the actual value being used (memory usage limit (active value)).

At the command prompt, type:

```
1     set cache parameter [-memLimit <MBytes>] [-via <string>] [-
          verifyUsing <criterion>] [-maxPostLen <positiveInteger>] [-
          prefetchMaxPending <positiveInteger>] [-enableBypass(YES|NO)] [-
          undefAction (NOCACHE|RESET)]
2 <!--NeedCopy-->
```

Enable integrated caching by Citrix ADC GUI

Navigate to **System** > **Settings**, click **Configure Basic Features**, and select **Integrated Caching**.

Configure global settings for caching by using the Citrix ADC GUI

Navigate to **Optimization** > **Integrated Caching**, click **Change Cache Settings**, and configure the global settings for caching.

Set up built-in content group, pattern set, and policies for Integrated Cache

The Citrix ADC appliance includes a built-in integrated caching configuration that you can use for caching content. The configuration consists of a content group called ctx_cg_poc, a pattern set called ctx_file_extensions, and a set of integrated cache policies. In the content group ctx_cg_poc, only objects that are 500 KB or smaller are cached. The content is cached for 86000 seconds, and the memory limit for the content group is 512 MB. The pattern set is an indexed array of common extensions for file-type matching.

The following table lists the built-in integrated caching policies. By default, the policies are not bound to any bind point. You must bind the policies to a bind point if you want the Citrix ADC appliance to evaluate traffic against the policies. The policies cache objects in the ctx_cg_poc content group.

Integrated caching policy name	Policy rule
_cacheVPNStaticObjects	HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_IN
_cacheTCPVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(".css")
_cacheOCVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(".pdf")
_cacheWFStaticObjects	HTTP.REQ.URL.ENDSWITH(".js")
_mayNoCacheReq	HTTP.RES.HEADER("Content-Type").CONTAINS("application/x-javascript")
_noCacheRest	TRUE

Flush cache configuration

You can flush a cache group, cache groups, or cache object locator. Following are the commands to flush cache objects.

At the command prompt, type:

```
flush cache contentgroup all
```

Example

```
1          0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hello
2          0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hi
3
4          Flush cache contentGroup all
5          done
6
7  `flush cache contentgroup <content group name>`
8  <!--NeedCopy-->
```

Example:

```
1          0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hello
2          0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hi
3
4          Flush cache ob -| 0x00000089bae000000004
5          done
```



```
6
7 `flush cache object (-locator <positive_integer> | (-url <URL> (-host <
     string> [-port <port>] [-groupName <string>] [-httpMethod ( GET |
     POST )])))`
8 <!--NeedCopy-->
```

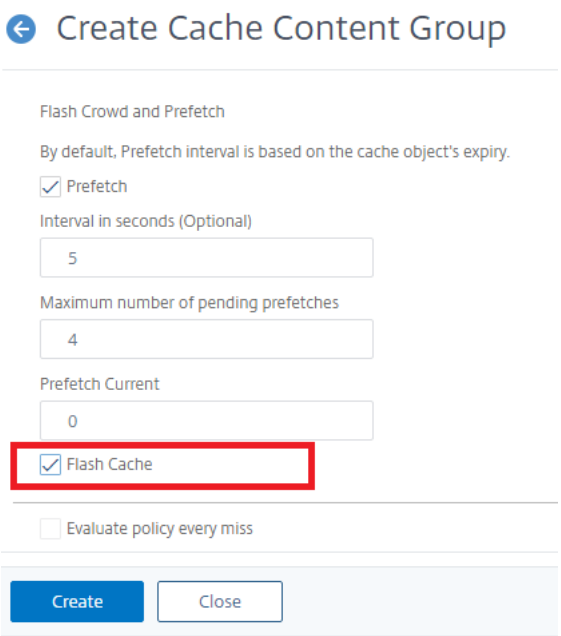
Example:

```
1         0x00000089bae000000006 DEFAULT GET //1.1.1.1:80/html/index.html
2
3         flush cache ob -URL /html/index.html -host 1.1.1.1 -groupName
             DEFAULT
4         done
5 <!--NeedCopy-->
```

Flush cache configuration by using the Citrix ADC GUI

Complete the steps to configure cache flushing using the Citrix ADC GUI

1. Navigate to **Optimization** > **Content Groups**.
2. In the **Content Groups** detailed pane, click **Add**.
3. In the **Create Cache Content Groups** page, set the following parameter under **Others** tab:
   a) Flush Cache. Select the check box to flush the cache object.
4. Click **Create** and **Close**.



Configure integrated caching for various scenarios

The following section describes the configuration of integrated caching on NetScaler appliance for various scenarios.

Starting from NetScaler 9.2 release, the integrated caching has more memory for caching. The integrated caching memory is only limited by the memory available on the hardware appliance. You can allocate up to 50 percent of the available memory to the integrated caching feature.

To set the memory allocation for the cache by using the CLI

At the command prompt, type:

```
set cache parameter -memlimit <value>
```

> **Note:**
>
> The default global memory limit for integrated caching is zero. Therefore, even if you enable the integrated caching feature, the NetScaler appliance does not cache any objects until the global memory limit is explicitly set.

The following section instructs you to configure integrated caching on different scenarios.

> **Note:**
>
> The memory limit of the NetScaler appliance is identified when the appliance starts. Therefore, any changes to the memory limit require you to restart the appliance to make the changes applicable across the packet engines.

Integrated caching is enabled and cache memory limit is set to non-zero

Consider a scenario, where you start the appliance, the integrated caching feature is enabled and the global memory limit is set to a positive number. The memory you had set earlier is allocated to the integrated caching feature during the boot process. You might want to change the memory limit to another value depending on the available memory on the appliance.

Configuring by using the CLI

1. Display the cache parameter

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 MBytes
4                  Memory usage limit (active value): 500 MBytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3: 18
7                  Verify cached object using: HOSTNAME_AND_IP
```



```
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

1. Set a non-zero memory limit

```
set cache parameter -memlimit 600
```

> **Note:**
>
> The preceding command displays the following warning message: **Warning: To use a new Integrated Cache memory limit, save the configuration and restart the NetScaler appliance.**

1. Save the configuration

```
save config
```

1. From the shell prompt, run the following command to verify in the configuration file.

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. Change the memory limit

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. Reboot the appliance

```
root@ns## reboot
```

1. Verify the new value for the memory limit

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 600 MBytes
                Memory usage limit (active value): 600 MBytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```


After all packet engines start successfully, the integrated caching feature negotiates the memory you had configured. If the appliance cannot use the configured memory, then the memory is allocated accordingly. If the available memory is less than the one you allocated, the appliance recommends a lesser number. The integrated caching feature uses same as the active value.

Integrated caching is disabled and cache memory limit is set to non-zero

In this scenario, when you start the appliance, the integrated caching feature is disabled and the global memory limit is set to a positive number. Therefore, no memory is allocated to the integrated caching during the boot process.

Configuring by using the CLI

1. Display the cache parameter

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 MBytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

1. Set a new memory limit

```
set cache parameter -memlimit 500
```

> **Note:**
>
> The preceding command displays the following warning message: **Warning: Feature not enabled [IC]**.

1. Save the configuration

```
save config
```

1. From the shell prompt, run the following command to verify in the configuration file

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. Change the memory limit


```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. Verify the new value for the memory limit

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 MBytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

1. Enable the integrated caching feature

```
enable ns feature IC
```

1. Verify the new value for the memory limit

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 Mbytes
4                  Memory usage limit (active value): 500 Mbytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3: 18
7                  Verify cached object using: HOSTNAME_AND_IP
8                  Max POST body size to accumulate: 0 bytes
9                  Current outstanding prefetches: 0
10                 Max outstanding prefetches: 4294967295
11                 Treat NOCACHE policies as BYPASS policies: YES
12                 Global Undef Action: NOCACHE
13 <!--NeedCopy-->
```

> **Note:**
>
> 500 MB of memory is allocated to the integrated caching feature.

1. Save the configuration to ensure that the memory is automatically allocated to the feature when the appliance is restarted.

Integrated caching is enabled and cache memory is set to zero

In this scenario, when you start the appliance, the integrated caching feature is enabled and the global memory limit is set to zero. Therefore, no memory is allocated to the integrated caching during the boot process.

Configuring by using the CLI

1. Verify the memory limits set in the ns.conf file from shell prompt

```
root@ns## cat ns.conf | grep memLimit
```

1. Change the memory limit

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

1. Verify the value for the memory limit

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 0 Mbytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

> **Note:**
>
> The memory limit is set to 0 MB and no memory is allocated to the integrated caching feature.

1. Set the memory limits to ensure the integrated caching feature caches objects

```
set cache parameter -memLimit 600
```

Once you run the preceding command, the appliance negotiates memory for the integrated caching feature and the available memory is assigned to the feature. It results in appliance caching objects without restarting the appliance.

1. Verify the value for the memory limit

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 Mbytes
```



```
                Memory usage limit (active value): 600 Mbytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3:
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> **Note:**
>
> 600 MB of memory is allocated to the integrated caching feature.

1. Save the configuration. Ensure that the memory is automatically allocated to the feature when the appliance is restarted.
2. Verify the memory limits set in the ns.conf file from shell prompt

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. Change the memory limit

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

Integrated caching is disabled and cache memory is set to zero

In this scenario, when you start the appliance, the integrated caching feature is disabled and the global memory limit is set to zero. Therefore, no memory is allocated to the integrated caching during the boot process.

Configuring by using the CLI

1. Verify the memory limits set in the ns.conf file from shell prompt

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. Change the memory limit

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

1. Verify the value for the memory limit

```
        > show cache parameter
                Integrated cache global configuration:
```



```
3                   Memory usage limit: 0 Mbytes
4                   Maximum value for Memory usage limit: 843 MBytes
5                   Via header: NS-CACHE-9.3: 18
6                   Verify cached object using: HOSTNAME_AND_IP
7                   Max POST body size to accumulate: 0 bytes
8                   Current outstanding prefetches: 0
9                   Max outstanding prefetches: 4294967295
10                  Treat NOCACHE policies as BYPASS policies: YES
11                  Global Undef Action: NOCACHE
12  <!--NeedCopy-->
```

> **Note:**
>
> The memory limit is set to 0 MB and no memory is allocated to the integrated caching feature. Also, when you run any cache configuration command, the following warning message is displayed: **Warning: Feature not enabled [IC]**.

1. Enable the integrated caching feature

```
enable ns feature IC
```

> **Note:**
>
> At this stage, when you enable the integrated caching feature, the appliance does not allocate memory to the feature. As a result, no object is cached to the memory. Also, when you run any cache configuration command, the following warning message is displayed: **No memory is configured for IC. Use set cache parameter command to set the memory limit.**

1. Set the memory limits to ensure the integrated caching feature caches objects

```
set cache parameter -memLimit 500
```

Once you run the preceding command, the appliance negotiates memory for the integrated caching feature and the available memory is assigned to the feature. It results in the appliance caching objects without restarting the appliance.

> **Note:**
>
> The order in which you enable the feature and set the memory limits is important. If you set the memory limits before enabling the feature, then the following warning message is displayed: **Warning: Feature not enabled [IC]**.

1. Verify the value for the memory limit

```
1           > show cache parameter
2                   Integrated cache global configuration:
3                   Memory usage limit: 500 Mbytes
4                   Memory usage limit (active value): 500 Mbytes
5                   Maximum value for Memory usage limit: 843 MBytes
```



```
                Via header: NS-CACHE-9.3:
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> **Note:**
>
> 500 MB of memory is allocated to the integrated caching feature.

1. Save the configuration

```
save config
```

1. Verify the memory limits set in the ns.conf file from shell prompt

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. Change the memory limit

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

Configure selectors and basic content groups

September 14, 2021

You can configure selectors and apply them to content groups. When you add a selector to one or more content groups, you specify whether the selector is to be used for identifying cache requests or identifying cached objects to be invalidated (expired). Selectors are optional. Alternatively, you can configure content groups to use `hit` parameters and invalidation parameters. However, Citrix recommends that you configure selectors.

After configuring selectors, or deciding to use parameters instead, you are ready to set up a basic content group. After creating the basic content group, you need to decide how objects should be expired from the cache, and configure cache expiration. You can further modify the cache as described in Improving Cache Performance and Configuring Cookies, Headers, and Polling, but you might first want to configure caching policies.

> **Note**
>
> Content group parameters and selectors are used only at request time, and you typically asso-

> ciate them with policies that use MAY_CACHE or MAY_NOCACHE actions.

Advantages of selectors

A selector is a filter that locates particular objects in a content group. If you do not configure a selector, the Citrix® ADC appliance looks for an exact match in the content group. This can lead to multiple copies of the same object residing in a content group. For example, a content group that does not have a selector may need to store URLs for host1.domain.com\mypage.htm, host2.domain.com\mypage.htm, and host3.domain.com\mypage.htm. In contrast, a selector can match just the URL (mypage.html, using the expression http.req.url) and the domain (.com, using the expression http.req.hostname.domain), allowing the requests to be satisfied by the same URL.

Selector expressions can perform simple matching of parameters (for example, to find objects that match a few query string parameters and their values). A selector expression can use Boolean logic, arithmetic operations, and combinations of attributes to identify objects (for example, segments of a URL stem, a query string, a string in a POST request body, a string in an HTTP header, a cookie). Selectors can also perform programmatic functions to analyze information in a request. For example, a selector can extract text in a POST body, convert the text into a list, and extract a specific item from the list.

For more information about expressions and what you can specify in an expression, see Policies and Expressions.

Use parameters Instead of selectors

Although Citrix recommends the use of selectors with a content group, you can instead configure `hit` parameters and invalidation parameters. For example, suppose that you configure three `hit` parameters in a content group for bug reports: BugID, Issuer, and Assignee. If a request contains BugID=456, with Issuer=RohitV and Assignee=Robert, the Citrix ADC appliance can serve responses that match these parameter-value pairs.

Invalidation parameters in a content group expire cached entries. For example, suppose that BugID is an invalidation parameter and a user issues a POST request to update a bug report. An invalidation policy directs the request to this content group, and the invalidation parameter for the content group expires all cached responses that match the BugID value. (The next time a user issues a GET request for this report, a caching policy can enable the Citrix ADC appliance to refresh the cached entry for the report from the origin server.)

Note that the same parameter can be used as a `hit` parameter or an invalidation parameter.

Content groups extract request parameters in the following order:

- URL query
- POST body

- Cookie header

After the first occurrence of a parameter, regardless of where it occurred in the request, all its subsequent occurrences are ignored. For example, if a parameter exists both in the URL query and in the POST body, only the one in the URL query is considered.

If you decide to use hit and invalidation parameters for a content group, configure the parameters when you configure the content group.

Note: Citrix recommends that you use selectors rather than parameterized content groups, because selectors are more flexible and can be adapted to more types of data.

Configure a selector

A content group can use a hit selector to retrieve cache hits or use an invalidation selector to expired cached objects and fetch new ones from the origin server.

A selector contains a name and a logical expression, called an *advanced expression*.

For more information about advanced expressions, see Policies and Expressions.

To configure a selector, you assign it a name and enter one or more expressions. As a best practice, a selector expression should include the URL stem and host, unless there is a strong reason to omit them.

To configure a selector by using the CLI

At the command prompt, type:

```
add cache selector \<selectorName\> ( \<rule\> ... )
```

For information about configuring the expression or expressions, see To configure a selector expression by using the command line interface.

```
1 >add cache selector product_selector "http.req.url.query.value(\"
    ProductId\")" "http.req.url.query.value(\"BatchNum\")" "http.req.url
    .query.value(\"depotLocation\")"
2
3 > add cache selector batch_selector "http.req.url.query.value(\"
    ProductId\")" "http.req.url.query.value(\"BatchId\")" "http.req.url.
    query.value(\"depotLocation\")"
4
5 > add cache selector product_id_selector "http.req.url.query.value(\"
    ProductId\")"
6
7 > add cache selector batchnum_selector "http.req.url.query.value(\"
    BatchNum\")" "http.req.url.query.value(\"depotLocation\")"
8
```



```
> add cache selector batchid_selector "http.req.url.query.value(\"
    depotLocation\")" "http.req.url.query.value(\"BatchId\")"

<!--NeedCopy-->
```

To configure a selector by using the GUI

Navigate to **Optimization > Integrated Caching > Cache Selectors**, and add the cache selector.

Content groups

A content group is a container for cached objects that can be served in a response. When you first enable the integrated cache, cacheable objects are stored in a content group named Default. You can create content groups that have unique properties. For example, you can define separate content groups for image data, bug reports, and stock quotes, and you can configure the stock quote content group to be refreshed more often than the other groups.

You can configure expiration of an entire content group or selected entries in a content group.

The data in a content group can be static or dynamic, as follows:

- **Static content groups**. Finds an exact match between the URL stem and host name on the request and the URL stem and host name of the response.
- **Dynamic content groups**. Looks for objects that contain particular parameter-value pairs, arbitrary strings, or string patterns. Dynamic content groups are useful when caching data that is updated frequently (for example, a bug report or a stock quote).

Serve a request from a content group

1. A user enters search criteria for an item, such as a bug report, and clicks the Find button in an HTML form.
2. The browser issues one or more HTTP GET requests. These requests contain parameters (for example, the bug owner, bug ID, and so on).
3. When the Citrix ADC appliance receives the requests, it searches for a matching policy, and if it finds a caching policy that matches these requests, it directs the requests to a content group.
4. The content group looks for appropriate objects in the content group, based on criteria that you configure in a selector.

For example, the content group can retrieve responses that match `NameField=username and BugID=ID`.

1. If it finds matching objects, the Citrix ADC appliance can serve them to the user's browser, where they are assembled into a complete response (for example, a bug report).

Invalidate an object in a content group

1. A user modifies data (for example, the user modifies the bug report and clicks the Submit button).
2. The browser sends this data in the form of one or more HTTP requests. For example, it can send a bug report in the form of several HTTP POST requests that contain information about the bug owner and bug ID.
3. The Citrix ADC appliance matches the requests against invalidation policies. Typically, these policies are configured to detect the HTTP POST method.
4. If the request matches an invalidation policy, the Citrix ADC appliance searches the content group that is associated with this policy, and expires responses that match the configured criteria for invalidation.

For example, an invalidation selector can find responses that match `NameField=username and BugID=ID`.

1. The next time the Citrix ADC appliance receives a GET request for these responses, it fetches refreshed versions from the origin server, caches the refreshed responses, and serves these responses to the user's browser, where they are assembled into a complete bug report.

Set up a basic content group

By default, all cached data is stored in the default content group. You can configure more content groups and specify these content groups in one or more policies.

You can configure content groups for static content, and you must configure content groups for dynamic content. You can modify the configuration of any content group, including the default group.

To set up a basic content group by using the command line interface

At the command prompt, type:

```
add cache contentgroup <name> (-hitSelector <hitSelectorName> -invalSelector
 <invalidationSelectorName> | -hitParams <hitParamName> -invalParams<
invalidationParamName>)-type <type> [-relExpiry <sec> | -relExpiryMilliSec
<msec>] [-heurExpiryParam <positiveInteger>]
```

```
add cache contentgroup Products_Details -hitSelector product_selector -
invalSelector id_selector
```

```
add cache contentgroup bugrep -hitParams IssuePage RecordID Template
TableId -invalParams RecordID -relExpiry 864000
```

To set up a basic content group by using the GUI

Navigate to **Optimization > Integrated Caching > Content Groups**, and create the content group.

Expire or flush cached objects

If a response does not have an Expires header or a Cache-Control header with an expiration time (Max-Age or Smax-Age), you must expire objects in a content group by using one of the following methods:

- Configure content group expiration settings to determine whether and how long to keep the object.
- Configure an invalidation policy and action for the content group. For more information, see Configuring Policies for Caching and Invalidation.
- Expire the content group or objects within it manually.

After a cached response expires, the Citrix ADC appliance refreshes it the next time the client issues a request for the response. By default, when the cache is full, the Citrix ADC appliance replaces the least recently used response first.

The following list describes methods for expiring cached responses using settings for a content group. Typically, these methods are specified as a percent or in seconds:

- **Manual**. Manually invalidate all responses in a content group or all responses in the cache.
- **Response-based**. Specific expiration intervals for positive and negative responses. Response-based expiry is considered only if the Last-Modified header is missing in the response.
- **Heuristic expiry**. For responses that have a Last-Modified header, the heuristic expiry specifies the mount of time taken from when the response was modified (calculated as the current time minus the Last-Modified time, multiplied by the heuristic expiry value). For example, if a Last-Modified header indicates that a response was updated 2 hours ago, and the heuristic expiry setting is 10%, cached objects expire after 0.2 hours. This method assumes that frequently updated responses must be expired more often.
- **Absolute or relative**. Specify an exact (absolute) time when the response expires every day, in HH:MM format, local time, or GMT. Local time may not work in all time zones.

Relative expiration specifies some seconds or milliseconds from the time a cache miss causes a trip to the origin server to the expiration of the response. If you specify relative expiration in milliseconds, enter a multiple of 10. This form of expiration works for all positive responses. Last-Modified, Expires, and Cache-Control headers in the response are ignored.

Absolute and relative expiration overrides any expiration information in the response itself.

- **On download**. The option Expire After Complete Response Received expires a response when it is downloaded. This is useful for frequently updated responses, for example, stock quotes. By default, this option is disabled.

Enabling both Flash Cache and Expire After Complete Response Received accelerates the performance of dynamic applications. When you enable both options, the Citrix ADC appliance fetches only one response for a block of simultaneous requests.

- **Pinned**. By default, when the cache is full the Citrix ADC appliance replaces the least recently

used response first. The Citrix ADC appliance does not apply this behavior to content groups that are marked as pinned.

If you do not configure expiration settings for a content group, the following are more options for expiring objects in the group:

- Configure a policy with an INVAL action that applies to the content group.
- Enter the names of content groups when configuring a policy that uses an INVAL action.

How expiration methods are applied

Expiration works differently for positive and negative responses. Positive and negative responses are described in the table, *Expiration of Positive, and Negative Responses* mentioned below.

The following are rules of thumb for understanding the expiration method that is applied to a content group:

- You can control whether the Citrix ADC appliance evaluates response headers when deciding whether to expire an object.
- Absolute and relative expiration causes the Citrix ADC appliance to ignore the response headers (they override any expiration information in the response).
- Heuristic expiration settings and "Weak Positive" and "Weak Negative" expiration (labeled as **Default** values in the configuration utility) cause the Citrix ADC appliance to examine the response headers. These settings work together as follows:
  - The value in an Expires or Cache-Control header overrides these content group settings.
  - For positive responses that lack an Expires or Cache-Control header but have a Last-Modified header, the Citrix ADC appliance compares heuristic expiration settings with the header value.
  - For positive responses that lack an Expires, Cache-Control, or Last-Modified header, the Citrix ADC appliance uses the "weak positive" value.
  - For negative responses that lack an Expires or Cache-Control header, the Citrix ADC appliance uses the "weak negative" value.

The following table describes how these methods are applied.

Response type	Expiration header type	Content Group Setting	Period the Object Remains in the Cache
Positive	Any header	Expire Content After (relExpiry) with no other settings	Use the value of the **Expire Content After** setting.

Response type	Expiration header type	Content Group Setting	Period the Object Remains in the Cache
Positive	Any header	Expire Content At (absExpiry) with no other settings	Subtract current date from the value of the **Expire Content At** setting.
Positive	Any header	Expire Content After (relExpiry) and Expire content at (absExpiry)	Use the smaller of the two values for the content group settings. See the previous rows in this table.
Positive	Last-Modified (with any other headers)	Heuristic (heurExpiry Param) with any other setting	Subtract the Last-Modified date from the current date, multiply the result by the value of the heuristic expiry setting, and then divide by 100.
Positive	Last-Modified (with any other headers)	Default (positive) (weakPosRel Expiry) and no other setting	Use the value of the Default (positive) expiry setting.
Positive	Expires or Cache-Control: Max-Age header is present	Last-Modified header is absent, Heuristic (heurExpiry Param), Default (positive) (weakPosRel Expiry), or both	Subtract the current date from the Expires or the `Cache-Control:Max-Age` date.
Positive	no caching headers	Default (positive) (weakPosRel Expiry) and any other expiration setting	Use the value of the Default (positive) setting.

Response type	Expiration header type	Content Group Setting	Period the Object Remains in the Cache
Positive	no caching headers	Heuristic (heurExpiry Param) is present, Default (positive) (weakPosRel Expiry) setting is absent.	If the Last-Modified header is absent, the response is not cached or it is cached with an Already Expired status. If the Last-Modified header is present, use the heuristic expiry value.
Negative	Expires or `Cache-Control:Max-Age`	Expire Content After (relExpiry), Expire Content At (absExpiry), or both settings	Subtract the current date from the value of the Expires header, or use the value of the Cache-Control:Max-Age header.
Negative	Expires or Cache-Control headers are absent	Expire Content After (relExpiry), Expire Content At (absExpiry), or both settings	Response is not cached, or is cached with an Already Expired status.
Negative	Expires or `Cache-Control:Max-Age`	Any setting	Subtract the current date from the Expires or `Cache-Control:Max-Age` date.
Negative	Expires and Cache-Control:Max-Age headers are absent	Default (negative) (weakNegRel Expiry)	Use the value of the Default (negative) setting.
Negative	Expires and Cache-Control:Max-Age headers are absent	Any setting other than Default (negative) (weakNegRel Expiry)	Object is not cached or is cached with an Already Expired status.

Expire a content group by manual method

You can manually expire all of the entries in a content group.

To manually expire all responses in a content group by using the command line interface

At the command prompt, type:

```
expire cache contentGroup <name>
```

To manually expire all responses in a content group by using the GUI

Navigate to **Optimization > Integrated Caching > Content Groups**, select the content group, and click Invalidate to expire all the responses in a content group.

To manually expire all responses in the cache by using the GUI

Navigate to **Optimization > Integrated Caching > Content Groups**, and click Invalidate All to expire all the responses in cache.

Configure periodic expiration of a content group

You can configure a content group so that it performs selective or full expiration of its entries. The expiration interval can be fixed or relative.

To configure content group expiration by using the command line interface

At the command prompt, type:

```
set cache contentgroup \<name> (-relExpiry|-relExpiryMilliSec|-absExpiry
|-absExpiryGMT| -heurExpiryParam|-weakPosRelExpiry|-weakNegRelExpiry| -
expireAtLastBye)\<expirationValue>
```

To configure content group expiration by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, select the content group, and specify expiry method.

Expire individual responses

Expiring a response forces the Citrix ADC appliance to fetch a refreshed copy from the origin server. Responses that do not have validators, for example, `ETag` or Last-Modified headers, cannot be revalidated. As a result, flushing these responses has the same effect as expiring them.

To expire a cached response in a content group for static data, you can specify a URL that must match the stored URL. If the cached response is part of a parameterized content group, you must specify the group name and the exact URL stem. The host name and the port number must be the same as in the host HTTP request header of the cached response. If the port is not specified, port 80 is assumed.

To expire individual responses in a content group by using the command line interface

At the command prompt, type:


```
expire cache object -url <URL> -host <hostName> [-port <port>] [-groupName<
contentGroupName>] [-httpMethod GET|POST]
```

To expire individual responses in a content group by using the CLI

At the command prompt, type the following command:

```
expire cache object -locator <positiveInteger>
```

To expire a cached response by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Cached Objects**, select the cached response, and expire.

To expire a response by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Cached Objects**, click **Search** and, set the search criteria to find the required cached response and expire.

Flushing responses in a content group

You can remove, or flush, all responses in a content group, some responses in a group, or all responses in the cache. Flushing a cached response frees up memory for new cached responses.

> **Note:**
>
> To flush responses for more than one object at a time, use the configuration utility method. The command line interface does not offer this option.

To flush responses from a content group by using the command line interface

At the command prompt, type:

```
flush cache contentGroup <name> [-query <queryString> | [-selectorValue <
selectorExpressionIDList> -host <hostName>]]
```

To flush responses from a content group by using the GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**.
2. In the details pane, flush the responses as follows:
   - To flush all responses in all content groups, click Invalidate All, and flush all the responses.
   - To flush responses in a particular content group, select the content group, click **Invalidate**, and flush all the responses.

> **Note**:
> If this content group uses a selector, you can selectively flush responses by entering a string in the Selector value text box, entering a host name in the Host text box. Then click **Flush and OK**.

> The Selector value can be a query string of up to 2319 characters that is used for parameterized invalidation.

If the content group uses an invalidation parameter, you can selectively flush responses by entering a string in the **Query** field.

If the content group uses an invalidation parameter and Invalidate objects belonging to the target host is configured, enter strings in the **Query and Host** fields.

To flush a cached response by using the command line interface

At the command prompt, type:

```
flush cache object -locator <positiveInteger> | -url <URL> -host <hostName>
 [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET|POST]
```

To flush a cached response by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Cached Objects**, select the cached object, and flush.

Deleting a content group

You can remove a content group if it is not used by any policy that stores responses in the cache. If the content group is bound to a policy, you must first remove the policy. Removing the content group removes all responses stored in that group.

You cannot remove the Default, BASEFILE, or Deltas group. The Default group stores cached responses that do not belong in any other content group.

To delete a content group by using the command line interface

At the command prompt, type:

```
rm cache contentgroup <name>
```

To delete a content group by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Content Groups,** select the content group, and delete.

Configure policies for caching and invalidation

September 14, 2021

Policies enable the integrated cache to determine whether to try to serve a response from the cache or the origin. The NetScaler appliance provides built-in policies for integrated caching, and you can

configure more policies. When you configure a policy, you associate it with an action. An action either caches the objects to which the policy applies or invalidates (expires) the objects. Typically, you based caching policies on information in GET and POST requests. You typically base invalidation policies on the presence of the POST method in requests, along with other information. You can use any information in a GET or POST request in a caching or an invalidation policy.

You can view some of the built-in policies in the integrated cache's Policies node in the configuration utility. The built-in policy names begin with an underscore (_).

Actions determine what the NetScaler appliance does when traffic matches a policy. The following actions are available:

- **Caching actions**. Policies that you associate with the CACHE action store responses in the cache and serve them from the cache.
- **Invalidation actions**. Policies that you associate with the INVAL action immediately expire cached responses and refresh them from the origin server. For Web-based applications, invalidation policies often evaluate POST requests.
- **"Do not cache" actions**. Policies that you associate with a NOCACHE action never store objects in the cache.
- **Provisionally cache actions**. Policies that you associate with a MAYCACHE or MAYNOCACHE action depend on the outcome of more policy evaluations.

Although the integrated cache does not store objects specified by the LOCK method, you can invalidate cached objects upon receipt of a `LOCK` request. For invalidation policies only, you can specify `LOCK` as a method by using the expression `http.req.method.eq( "lock" )`. Unlike policies for `GET` and `POST` requests, you must enclose the LOCK method in quotes because the NetScaler appliance recognizes this method name as a string only.

After you create a policy, you bind it to a particular point in the overall processing of requests and responses. Although you create a policy before binding it, you must understand how the bind points affect the order of processing before you create your policies.

The policies bound to a particular bind point constitute a policy bank. You can use goto expressions to modify the order of execution in a policy bank. You can also invoke policies in other policy banks. In addition, you can create labels and bind policies to them. Such a label is not associated with a processing point, but the policies bound to it can be invoked from other policy banks.

Actions to associate with integrated caching policies

The following table describes actions for integrated caching policies.

Action	Specification
CACHE	Serves a response from the cache if the response has not expired. If the response must be fetched from the origin server, the NetScaler appliance caches the response before serving it. Even data that is updated and accessed frequently can be cached. For example, stock quotes are updated frequently, but they can be cached so that they can be served quickly to multiple users. If necessary, cached data can be refreshed immediately after it is downloaded. A CACHE action can be overridden by built-in policies.
NOCACHE	Always fetches the response from the origin server and marks the response as non-storable. You typically configure NOCACHE policies for data that is sensitive or personalized.

Action	Specification
MAY_CACHE	Used in a request-time policy, this setting provisionally enables a response to be stored in a content group, pending evaluation of response-time policies. The following are possible: 1. If a matching response-time policy has a CACHE action but does not specify a content group, the response is stored in the Default group unless built-in policies override this policy. 2. If a matching response-time policy has a CACHE action and specifies the same content group as the one in the request-time policy, the response is stored in the named content group unless built-in policies override this policy. 3. If a matching response-time policy has a CACHE action but specifies a different content group from the one in the request-time policy, a NOCACHE action is applied. 4. If a matching response-time policy has a NOCACHE action, perform a NOCACHE action. 5. If there is no matching response-time policy, a CACHE action is applied, unless a built-in policy overrides this policy.

Action	Specification
MAY_NOCACHE	For a request-time policy, this setting provisionally prevents caching the response. At response time, one of the following actions is taken: - If no response-time policy matches the request, the final action is NOCACHE. - If a matching response-time policy contains a CACHE action, the final action is CACHE, unless built-in policies override this policy. - If a matching response-time policy contains a NOCACHE action, the final action is NOCACHE. -If a matching response-time policy has a CACHE action but does not specify a content group, the final action is to CACHE the response in the Default content group, unless built-in policies override this policy.
INVAL	Expires cached responses. Depending on how the policy and the content group are configured, all responses in one or more content groups are expired, or selected objects in the content group are expired. Note: You can specify INVAL actions in request-time policies only.

Bind points for a policy

You can bind the policy to one of the following bind points:

- **A global policy bank**. These are the request-time default, request-time override, response-time default, and response-time override policy banks, as described in "Order of Policy Evaluation."
- **A virtual server**. Policies that you bind to a virtual server are processed after the global override policies and before the global default policies, as described in "Order of Policy Evaluation." When binding a policy to a virtual server, you bind it to either request-time or response-time processing.
- **An ad-hoc policy label**. A policy label is a name assigned to a policy bank. In addition to the global labels, the integrated cache has two built-in custom policy labels:
  - _reqBuiltinDefaults. This policy label, by default, is invoked from the request-time default policy bank.

  - _resBuiltinDefaults. This policy label, by default, is invoked from the response-time default policy bank.

You can also define new policy labels. Policies bound to a user-defined policy label must be invoked from within a policy bank for one of the built-in bind points.

> **Important:**
> You must bind a policy with an INVAL action to a request-time override or a response-time override bind point. To delete a policy, you must first unbind it.

Order of policy evaluation

For an advanced policy to take effect, you must ensure that the policy is invoked at some point during the NetScaler appliance's processing of traffic. To specify the invocation time, you associate the policy with a bind point. The following are the bind points, listed in order of evaluation:

- **Request-time override**. If a request matches a request-time override policy, by default the request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time load balancing virtual server**. If policy evaluation cannot be completed after all the request-time override policies are evaluated, the NetScaler appliance processes request-time policies that are bound to load balancing virtual servers. If the request matches one of these policies, evaluation ends and the NetScaler appliance store the action that is associated with the matching policy.
- **Request-time content switching virtual server**. Policies that are bound to this bind point are evaluated after the request-time policies that are bound to load balancing virtual servers.
- **Request-time default**. If policy evaluation cannot be completed after all the request-time, virtual server-specific policies are evaluated, the NetScaler appliance processes the request-time default policies. If the request matches a request-time default policy, by default the request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Response-time override**. Similar to request-time override policy evaluation.
- **Response-time load balancing virtual server**. Similar to request-time virtual server policy evaluation.
- **Response-time content switching virtual server**. Similar to request-time virtual server policy evaluation.
- **Response-time default**. Similar to request-time default policy evaluation.

You can associate multiple policies with each bind point. To control the order of evaluation of the policies associated with the bind point you configure a priority level. In the absence of any other flow control information, policies are evaluated according to priority level, starting with the lowest numeric priority value.

> **Note:**
> Request-time policies for POST data or cookie headers must be invoked during request-time override evaluation, because the built-in request-time policies in the integrated cache return a NOCACHE action for POST requests and a MAY_NOCACHE action for requests with cookies. You would associate MAY_CACHE or MAY_NOCACHE actions with a request-time policy that points to a parameterized content group. The response time policy determines whether the transaction is stored in the cache.

Configure a policy for integrated caching

You configure new policies to handle data that the built-in policies cannot process. You configure separate policies for caching, preventing caching from occurring, and for invalidating cached data. Following are the main components of a policy for integrated caching:

- Rule: A logical expression that evaluates an HTTP request or response.
- Action: You associate a policy with an action to determine what to do with a request or response that matches the policy rule.

Content groups: You associate the policy with one or more content groups to identify where the action is to be performed.

To configure a policy for caching by using the command line interface

At the command prompt, type:

```
add cache policy <policyName> -rule <expression> -actionCACHE|MAY_CACHE
|NOCACHE|MAY_NOCACHE [-storeInGroup <contentGroupName>] [-undefAction
NOCACHE|RESET]
> add cache policy image_cache -rule "http.req.url.contains(\"jpg\")|| http
.req.url.contains(\"jpeg\")"-action CACHE -storeingroup myImages_group -
undefaction NOCACHE

> add cache policy bugReportPolicy -rule "http.req.url.query.contains(\"
IssuePage\")"-action CACHE -storeInGroup bugReportGroup

> add cache policy my_form_policy -rule "http.req.header(\"Host\")contains
(\"my.company.com\")&& http.req.method.eq(\"GET\")&& http.req.url.query.
contains(\"v=7\")"-action CACHE -storeInGroup my_form_event

> add cache policy viewproducts_policy -rule "http.req.url.contains(\"
viewproducts.aspx\")"-action CACHE -storeInGroup Product_Details
```

To configure a policy for invalidation by using the command line interface

At the command prompt, type:


```
add cache policy <policyName> -rule <expression> -action INVAL [-
    invalObjects "\<contentGroupName1>[,<selectorName1>"]. . .]] | [-
    invalGroup \<contentGroupName1>[, <contentGroupName2>. . .]] [-
    undefAction NOCACHE|RESET]
<!--NeedCopy-->
```

```
> add cache policy invalidation_events_policy -rule "http.req.header("
    Host")contains("my.company.com") && http.req.method.eq("GET") &&
    http.req.url.query.contains("v=8") -action INVAL -invalObjects
    my_form_event -undefaction NOCACHE
<!--NeedCopy-->
```

```
> add cache policy inval_all -rule "http.req.method.eq("POST") && http.
    req.url.contains("jpeg")" -action INVAL -invalGroups myImages_group
    myApps_group PDF_group
<!--NeedCopy-->
```

```
> add cache policy bugReportInvalidationPolicy -rule "http.req.url.
    query.contains(\"TransitionForm\")" -action INVAL -invalObjects
    bugReport`
`> add cache policy editproducts_policy  - rule "http.req.url.contains
    (\"editproducts.aspx\")"  - action INVAL -invalObjects "
    Product_Details,batchnum_sel" "Products_In_Depots,batchid_sel"
<!--NeedCopy-->
```

To configure a policy for caching or invalidation by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Policies**, and create the new policy.

Globally binding an integrated caching policy

When you globally bind a policy, it is available to all virtual servers on the NetScaler appliance.

To bind an integrated caching policy globally by using the command line interface:

At the command prompt, type:

```
bind cache global <policy> -priority <positiveInteger> [-
    typeREQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] [-
    gotoPriorityExpression <expression>] [-invoke <labelType> <labelName
    >]
<!--NeedCopy-->
```



```
1 > bind cache global myCachePolicy -priority 100 -type req_default
2 <!--NeedCopy-->
```

> **Note:**
>
> The type argument is optional for globally bound policies, to maintain backward compatibility with policies that you defined using earlier versions of the NetScaler appliance. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. If the rule contains both request time and response time parameters, it is bound to RES_DEFAULT. Following is an example of a binding that omits the type

Following is an example of a binding that omits the type.

```
> bind cache global myCache Policy 200
```

To bind an integrated caching policy globally by using the configuration utility

Navigate to **Optimization** > **Integrated Caching**, click **Cache Policy Manager** and bind policies by specifying the relevant bind point and connection type (Request/Response).

Bind an integrated caching policy to a virtual server

When you bind a policy to a virtual server, it is available only to requests and responses that match the policy and that flow through the relevant virtual server.

When using the GUI, you can bind the policy using the configuration dialog box for the virtual server. This enables you to view all of the policies from all Citrix ADC modules that are bound to this virtual server. You can also use the **Policy Manager configuration** dialog for the integrated cache. This enables you to view only the integrated caching policies that are bound to the virtual server.

To bind an integrated caching policy to a virtual server by using the command line interface:

At the command prompt, type:

```
1 bind lb vserver <name>@ -policyName <policyName> -priority <
     positiveInteger> -type(REQUEST|RESPONSE)
2 <!--NeedCopy-->
```

```
1 bind cs vserver <name>@ -policyName <policyName> -priority <
     positiveInteger> -type(REQUEST|RESPONSE)
2 <!--NeedCopy-->
```

To bind an integrated caching policy to a virtual server by using the configuration utility (virtual server method)

- CS Virtual Server - Navigate toTraffic **Management > Content Switching > Virtual Servers**, select the virtual server, and bind relevant cache policies.
- LB Virtual Server - Navigate toTraffic **Management > Load Balancing > Virtual Servers**, select the virtual server, and bind relevant cache policies.

To bind an integrated caching policy to a virtual server by using the GUI (Policy Manager method).

Navigate to **Optimization** > **Integrated Caching**, click **Cache Policy Manager**, and bind cache policies by specifying the relevant bind point and connection type.

> **Note:**
> You can bind cache policies to both load balancing virtual server and content switching virtual server by selecting the appropriate bind point.

How to cache compressed and uncompressed versions of a file

By default, a client that can handle compression can be served uncompressed responses or compressed responses in gzip, deflate, compress, and pack200-gzip format. If the client handles compression, an `Accept-Encoding:compression` format header is sent in the request. The compression type accepted by the client must match the compression type of the cached object. For example, a `cached.gzip` file cannot be served in response to a request with an `Accept-Encoding:deflate` header.

A client that cannot handle compression is served a cache miss if the cached response is compressed.

For dynamic caching, you need to configure two content groups, one for compressed data and one for uncompressed versions of the same data. The following is an example of configuring the selectors, content groups, and policies for serving uncompressed files from the cache to clients that cannot handle compression, and serving compressed versions of the same files to the client that can handle compression.

```
add cache selector uncompressed_response_selector http.req.url "http.req.
header(\"Host\")"

add cache contentGroup uncompressed_group -hitSelector uncompressed_responst_selector
 -invalSelector uncomp_resp_sel

add cache policy cache_uncompressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\")&&
 !HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS"-action CACHE -storeInGroup
uncompressed_group

bind cache global cache_uncompressed -priority 100 -gotoPriorityExpression
END -type REQ_OVERRIDE

add cache selector compressed_response_selector HTTP.REQ.URL "HTTP.REQ.
HEADER(\"Host\")""HTTP.REQ.HEADER(\"Accept-Encoding\")"
```



```
add cache contentGroup compressed_group -hitSelector compressed_response_selector


add cache policy cache_compressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\")&&
 HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS"-action CACHE -storeInGroup
compressed_group

bind cache global cache_compressed -priority 200 -gotoPriorityExpression
END -type REQ_OVERRIDE
```

Configure a policy bank for caching

All of the policies that are associated with a particular bind point are collectively known as a policy bank. In addition to configuring priority levels for policies in a bank, you can modify the order of evaluation in a bank by configuring Goto expressions. You can further modify the evaluation order by invoking an external policy bank from within the current policy bank. You can also configure new policy banks, to which you assign your own labels. Because such policy banks are not bound to any point in the processing cycle, they can be invoked only from within other policy banks. For convenience, policy banks whose labels do not correspond to a built-in bind point are called policy labels.

In addition to controlling the order of policy evaluation by binding the policy and assigning a priority level, as described in "Binding Policies", you can establish the flow within a bank of policies by configuring a Goto expression. A Goto expression overrides the flow that is determined by the priority levels. You can also control the evaluation flow by invoking an external policy bank after evaluating an entry in the current bank. Evaluation always returns to the current bank after evaluation has completed.

The following table summarizes the entries to control evaluation in a policy bank.

Attribute	Specifies
Name	The name of a policy, or, to invoke another policy bank without evaluating the policy, the keyword NOPOLICY. You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.
Priority	An integer. The lower the integer, the higher the priority.

Attribute	Specifies
Goto Expression	Determines the next policy or policy bank to evaluate. You can provide one of the following values: 1. NEXT: Go to the policy with the next higher priority. 2. END: Stop evaluation. 3. USE_INVOCATION_RESULT: Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT. 4. Positive number: Priority number of the next policy to be evaluated. 5. Numeric expression: Expression that produces the priority number of the next policy to be evaluated. The Goto can only proceed forward in a policy bank. Omitting the Goto expression is the same as specifying END.
Invocation Type	Designates a policy bank type. The value can be one of the following - 1. Request virtual server: Invokes request-time policies that are associated with a virtual server. 2. Response virtual server: Invokes response-time policies that are associated with a virtual server. 3. Policy label: Invokes another policy bank, as identified by the policy label for the bank.
Invocation Name	Name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

The integrated cache has two built-in policy labels, and you can configure more policy labels:

`_reqBuiltInDefaults`: This policy label is invoked from the request-time default bind point.
`_resBuiltInDefaults`: This policy label is invoked from the response-time default bind point.

To invoke a policy label in a caching policy bank by using the command line interface

At the command prompt, type:

```
bind cache policylabel <labelName> -policname<policyName> -priority<
    priority> [-gotoPriorityExpression <gotopriorityExpression>] [-
```



```
    invoke <labelType> <labelName>]
<!--NeedCopy-->
```

To invoke a policy label in a caching policy bank by using the GUI:

1. Navigate to **Optimization** > **Integrated Caching**, click **Cache policy manager**, and specify the relevant bind point (Override Global or Default Global) and connection type to view the list of policies bound to this bind point.
2. If you want to invoke a policy label without evaluating a policy, click **NOPOLICY**.

> **Note:**
>
> To invoke an external policy bank, click the field in the Invoke Type column, and select the type of policy bank that you want to invoke at this point in the policy bank. This can be a global label or a virtual server bank. In the Invoke Name field, enter the label or virtual server name.

To invoke a caching policy label in a virtual server policy bank by using the command line interface

At the command prompt, type:

```
bind lb vserver <name>@ -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

```
bind cs vserver <name> -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

To invoke a caching policy label in a virtual server policy bank by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing/Content Switching** > **Virtual Servers**, select the virtual server, and click **Policies**.
2. If you are configuring an existing entry in this bank, skip this step. If you are adding a new policy to this policy bank, or you want to use the “dummy” NOPOLICY entry, click **Add** and do one of the following:
   - To configure a new policy, click Cache and configure the new policy as described in Configuring a Policy in the Integrated Cache.
   - To invoke a policy bank without processing a policy a rule, select the `NOPOLICY-CACHE` option.

> **Note:**
>
> To invoke an external policy bank, click the field in the Invoke Type column, and select the type

> of policy bank that you want to invoke at this point in the policy bank. This can be a global label or a virtual server bank. In the Invoke Name field, enter the label or virtual server name.

Configure a policy label in an integrated cache

In addition to configuring policies in a policy bank for one of the built-in bind points or a virtual server, you can create caching policy labels and configure banks of policies for these new labels.

A policy label for the integrated cache can be invoked only from one of the bind points that you can view in the Policy Manager in the **Integrated Caching** details pane (request override, request default, response override, or response default) or the built-in policy labels `\\_reqBuiltinDefaults` and `\\_resBuiltinDefaults`. You can invoke a policy label any number of times unlike a policy, which can only be invoked once.

The Citrix ADC GUI provides an option to rename a policy label. Renaming a policy label does not affect the process of evaluation of the policies bound to the label.

> **Note**:
>
> You can use the `NOPOLICY` "dummy" policy to invoke any policy label from another policy bank. The `NOPOLICY` entry is a placeholder that does not process a rule.

To configure a policy label for caching by using the command line interface

At the command prompt, type the following command to create a policy label and verify the configuration:

- `add cache policylabel <labelName> -evaluates (REQ|RES)`
- `show cache policylabel <labelName>`

Invoke this policy label from a policy bank.

To configure a policy label for caching by using the GUI:

Navigate to **Optimization** > **Integrated Caching** > **Policy Labels**, add a policy label, and bind the cached policies.

> **Note:**
> To ensure that the Citrix ADC processes the policy label at the right time, configure an invocation of this label in one of the policy banks that are associated with the built-in bind points.

To rename a policy label by using the GUI:

Navigate to **Optimization** > **Integrated Caching** > **Policy Labels** select the policy label, and rename.

Unbind and delete an integrated caching policy and policy label

You can unbind a policy from a policy bank, and you can delete it. To delete the policy, you must first unbind it. You can also remove a policy label invocation and delete a policy label. To delete the policy

label, you must first remove any invocations that you have configured for the label.

You cannot unbind or delete the labels for the built-in bind points (request default, request override, response default, and response override).

To unbind a global caching policy by using the command line interface

At the command prompt, type:

```
unbind cache global <policy>
```

To unbind a virtual server-specific caching policy by using the command line interface

At the command prompt, type:

```
(unbind lb vserver|unbind cs vserver)<vserverName> -policyName <policyName>
 -type(REQUEST|RESPONSE)
```

To delete a caching policy by using the command line interface

At the command prompt, type:

```
rm cache policy <policyName>
```

To unbind a caching policy by using the GUI:

Navigate to **Optimization** > **Integrated Caching**, click **Cache Policy Manager**, and unbind policies by specifying the relevant bind point and connection type (Request/Response).

To delete a policy label invocation by using the GUI:

1. Navigate to **Optimization** > **Integrated Caching**, click **Cache policy manager**, and specify the relevant binding point (load balancing virtual server or content switching virtual server) and connection type to view the list of cache policies bound to this virtual server.
2. In the policy Invoke column, clear the entry.

Cache support for database protocols

September 14, 2021

The integrated cache feature monitors and caches database request as determined by the cache policies. Users must configure the cache policies for MYSQL and MSSQL protocols as the Citrix ADC appliance does not provide any default policies. When configuring the default protocols, remember the request-based policies support only CACHE and INVAL actions, while the response-based policies support only “NOCACHE” action. After configuring the policies, you must bind them to virtual servers. MYSQL and MSSQL policies, both request and response, are bound only to virtual servers.

Before creating a cache policy, you must create a cache content group of type MYSQL or MSSQL. When you create a cache content group, associate at least one select selector with it. See Setting Up a Basic Content Group for setting up a cache content group.

The following example explains how to configure and verify cache support for SQL protocols.

```
> enable feature IC
> set cache parameter -memlimit 100
> add cache selector sel1 mssql.req.query.text

> add cache contentgroup cg1  -type "MSSQL" -hitselector "sel1" -
   invalselector "inval_sel" -relExpiry "500" -maxResSize
 "100"
> add cache policy cp1  -rule "mssql.req.query.command.contains(\"
   select\")" -action "CACHE" -storeInGroup "cg1"
> add cache policy cp2  -invalObjects "cg1" -rule "mssql.req.query.text
   .contains(\"insert\")" -action "INVAL"
> add db user user1  -password "Pass1"
> add service svc_sql_1 10.102.147.70 mssql 64834 -healthMonitor "NO" -
   downstateflush "ENABLED"
> add lb vserver lb_mssql1 mssql 10.102.147.77 1433  -lbmethod "
   roundrobin"
> bind lb vserver lb_mssql1 svc_sql_1
> bind lb vserver lb_mssql1 -policyName cp1  -type "REQUEST" -priority
   "2"
> bind lb vserver lb_mssql1 -policyName cp2  -type "REQUEST" -priority
   "1"

> show cache selector sel1
     Name:sel1
                    Expressions:
                    1)mssql.req.query.text
> show cache policy cp1
                    Name:cp1
     Rule:mssql.req.query.command.contains("select")
                    CacheAction:CACHE
                    Stored in group: cg1
                    UndefAction:Use Global
                Hits:2
                    Undef Hits:0
                    Policy is bound to following entities
                    1) Bound to:
                                REQ VSERVER lb_mssql1
                                Priority:2
                                GotoPriorityExpression: END
```



```
<!--NeedCopy-->
```

> **Note:**
>
> The methods for reducing flash crowds, as explained in Reducing Flash Crowds, are not supported for MYSQL and MSSQL protocols.

Configure expressions for caching policies and selectors

September 14, 2021

A request-time expression examines data in the request-time transaction, and a response-time expression examines data in a response-time transaction. In a policy for caching, if an expression matches data in a request or response, the Citrix ADC appliance takes the action associated with the policy. In a selector, request-time expressions are used to find matching responses that are stored in a content group.

Before configuring policies and selectors for the integrated cache, you need to know, at minimum, the host names, paths, and IP addresses that appear in HTTP request and response URLs. And you probably need to know the format of entire HTTP requests and responses. Programs such as Live HTTP Headers `http://livehttpheaders.mozdev.org/)or HTTPFox https://addons.mozilla.org/en-US/firefox/addon/6647` can help you investigate the structure of the HTTP data that your organization works with.

Following is an example of an HTTP GET request for a stock quote program:

```
GET /quote.dll?page=dynamic&mode=data&mode=stock&symbol=CTXS&page=multi
    &selected=CTXS&random=0.00792039478975548 HTTP/1.1

Host: quotes.mystockquotes.com

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
    Gecko/2008052906 Firefox/3.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate,compress,pack200-gzip

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300
```



```
16
17 Connection: keep-alive
18
19 Referer: http://quotes.mystockquotes.com/quote.dll?mode=stock&symbol=
      CTXS&page=multi&selected=CTXS
20
21 Cookie: __qca=1210021679-72161677-10297606
22 <!--NeedCopy-->
```

When configuring an expression, note the following limitations:

Expression Type	Restrictions
Request	Do not configure request-time expressions in a policy with a CACHE or NOCACHE action. Use MAY_CACHE or MAY_NOCACHE instead.
Response	Configure response-time expressions in caching policies only. Selectors can use only request-time expressions, Do not configure response-time expressions in a policy with an INVAL action. Note: Do not configure response-time expressions in a policy with a CACHE action and a parameterized content group. Use the MAY_CACHE action.

> **Note:**
> For a comprehensive discussion of advanced expressions, see Policies and Expression.

Expression syntax

Following are the basic components of the syntax:

- Separate keywords with periods (.), as follows:

```
http.req.url
```

- Enclose string values in parentheses and quotes, as follows:

```
http.req.url.query.contains("this")
```

- When configuring an expression from the command line, you must escape internal quote marks (the quotes that delimit the values in the expression, as opposed to the quotes that delimit the expression). One method is to use a slash, as followings:


```
\"abc\"
```

Selector expressions are evaluated in order of appearance, and multiple expressions in a selector definition are joined by a logical AND. Unlike selector expressions, you can specify Boolean operators and modify the precedence in an advanced expression for a policy rule.

Configure an expression in a caching policy or a selector

> **Note:**
> The syntax for a policy expression is different from a selector expression. For a comprehensive discussion of advanced expressions, see “Policies and Expressions.”

To configure a policy expression by using the command line interface

1. Start the policy definition as described in “Globally Binding an Integrated Caching Policy.”
2. To configure the policy rule, delimit the entire rule in quotes, and delimit string values within the rule in escaped quotes.

The following is an example:

“http.req.url.contains(”jpg”)”

To add Boolean values, insert , or ! operators. &&,

1.

The following are examples:

```
"http.req.url.contains(\"jpg\")|| http.req.url.contains(\"jpeg\")"
```

```
"http.req.url.query.contains(\"IssuePage\")"
```

```
"http.req.header(\"Host\")contains(\"my.company.com\")&& http.req.method.eq
(\"GET\")&& http.req.url.query.contains(\"v=7\")"
```

1. To configure an order of evaluation for the constituent parts of a compound

```
"http.req.url.contains(\"jpg\")|| (http.req.url.contains(\"jpeg\")&& http.
req.method.eq(\"GET\"))"
```

To configure a selector expression by using the command line interface:

1. Start the selector definition as described in “About Content Groups.”
2. To configure the selector expression, delimit the entire rule in quotes, and delimit string values within the rule in escaped quotes.

The following is an example:


```
"http.req.url.contains(\"jpg\")"
```

You cannot add Boolean values, insert &&,	, or ! operators. Enter each expression element delimited in quotes. Multiple expressions in the definition are treated as a compound expression joined by logical ANDs.

1.

The following are examples:

```
1 "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"
     BatchNum\")" "http.req.url.query.value(\"depotLocation\")"
2 <!--NeedCopy-->
```

To configure a policy or selector expression by using the GUI

1. Start the policy or selector definition as described in “To configure a policy for caching or invalidation by using the configuration utility” or “To configure a selector by using the configuration utility.”
2. In the **Expression** field, you can either manually type the default syntax by clicking Switch to Classic Syntax or create new expression using **Expression Editor**.

To insert an operator between two parts of a compound expression, click the Operators button and select the operator type. The following is an example of a configured expression with a Boolean OR (signaled by double vertical bars,	):

3.
4. Click **Frequently Used Expressions** drop-down list to insert the commonly used expressions.
5. To test the expression, click the **Evaluate**. In the **Expression Evaluator** dialog box, select the

Flow Type that matches the expression. In the data field, paste the HTTP request or response that you hope to parse using the expression, and click **Evaluate**.

Display cached objects and cache statistics

You can view particular cached objects, and you can view summary statistics on cache requests, misses, and memory usage. The statistics provide insight on the amount of data that is being served from the cache, what items are responsible for the largest performance benefit, and what you can tune to improve cache performance.

This section includes the following details:

- Viewing Cached Objects
- Finding Particular Cached Responses
- Viewing Cache Statistics

View cached objects

After enabling caching, you can view details for cached objects. For example, you can view the following items:

- Response sizes and header sizes
- Status codes
- Content groups
- ETag, Last-Modified, and Cache-Control headers
- Request URLs
- Hit parameters
- Destination IP addresses
- Request and response times

To view a list of cached objects by using the command line interface

At the command prompt, type:

```
show cache object
```

Properties	Description
Response size (bytes)	The size of the response header and body.
Response header size (bytes)	The size of the header portion of the response.
Response status code	The status code sent with the response.
ETag	The ETag header inserted in the response. Typically, this header indicates whether the response has changed recently.

Properties	Description
Last-Modified	The Last-Modified header inserted in the response. This header indicates the date that the response was last changed.
Cache-Control	The Cache-Control header inserted in the response.
Date	The Date header that indicates when the response was sent.
Contentgroup	The content group where the response is stored.
Complex match	If this object was cached based on parameterized values, this field value is YES.
Host	The host specified in the URL that requested this response.
Host port	The listen port for the host specified in the URL that requested this response
URL	The URL issued for the stored response.
Destination IP	The IP address of the server from which this response was fetched.
Destination port	The listen port for the destination server.
Hit parameters	If the content group that stores the response uses hit parameters, they are listed in this field.
Hit selector	If this content group uses a hit selector, it is listed in this field.
Inval selector	If this content group uses an invalidation selector, it is listed in this field.
Selector Expressions	If this content group uses a selector, this field displays the expression that defines the selection rule.
Request time	The time in milliseconds since the request was issued.
Response time	The time in milliseconds since the cache started to receive the response.
Age	Amount of time the object has been in the cache.

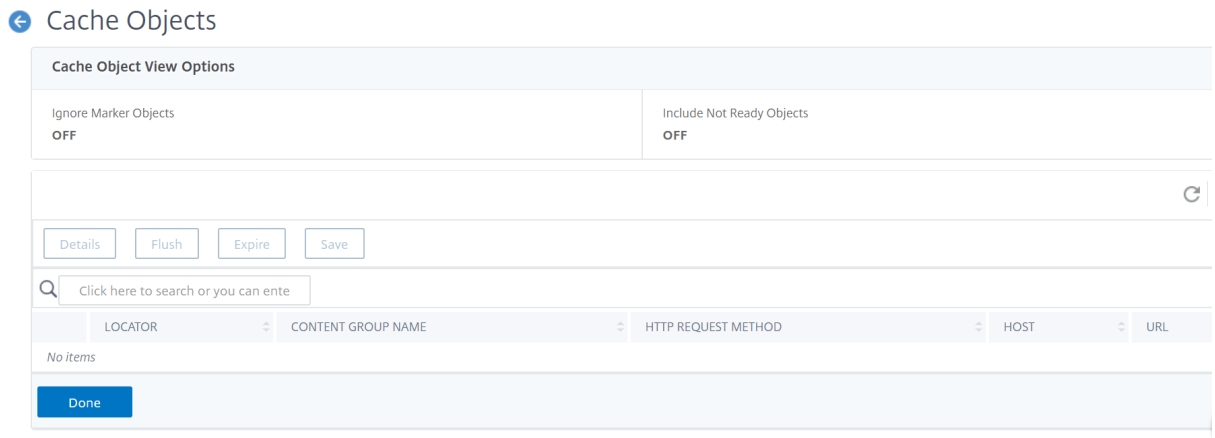
Properties	Description
Expiry	Amount of time after which the object is marked as expired.
Flushed	Whether the response has been flushed after expiry.
Prefetch	If Prefetch has been configured for this content group, the amount of time before expiry during which the object is fetched from the origin. Prefetch does not apply to negative objects (for example, 404 “object not found” responses).
Current readers	Approximately the current number of requests being served. When a response with a Content-Length header object is being downloaded, the current misses and the current readers values are each typically 1. When a chunked response object is being downloaded, the current misses value is typically 1, but the current readers value is typically 0, because the chunked response that is served to the client does not come from the integrated caching buffers.
Current misses	The current number of requests that resulted in a cache miss and fetching from the origin server. This value is typically 0 or 1. If Poll Every Time is enabled for a content group, the count can be greater than 1.
Hits	The number of cache hits for this object.
Misses	The number of cache misses for this object
Compression format	The type of compression applied to this object. Compression formats include gzip, deflate, compress, and pack200-gzip.
HTTP version in response	The version of HTTP that was used to send the response.

Properties	Description
Weak etag present in response	Strong etag headers change if the bits of an entity change. Strong headers are based on the octet values of an object. Weak etag headers change if the meaning of an entity changes. Weak etag values are based on semantic identity. Weak etags values start with a “W.”
Negative marker cell	A marker object is cacheable, but it does not yet meet all the criteria for being cached. For example, the object may exceed the maximum response size for the content group. A marker cell is created for objects of this type. The next time a user sends a request for this object, a cache miss is served.
Reason marker created	The reason a marker cell was created (for example, “Waiting for minhit,” “Content-length response data is not in group size limit”).
Auto poll every time	If the integrated cache receives an already expired 200 OK response with validators (either the Last-Modified or the ETag response headers) it stores the response and marks it as Auto-PET (automatically poll every time).
Citrix ADC Etag inserted in response	A variation of the ETag header generated by the Citrix ADC appliance. A value of YES appears if the Citrix ADC inserts an Etag in the response.
Full response present in cache	Indicates whether this is a complete response.
Destination IP verified by DNS	Indicates whether DNS resolution was performed when storing the object.
Object stored through a cache forward proxy	Indicates whether this response was stored due to a forward proxy that is configured in the integrated cache.
Object is a Delta basefile	A response that is delta-compressed.
Waiting for minhits	Indicates whether this content group requires a minimum number of origin servers hit before caching a response.

Properties	Description
Minhit count	If this content group requires a minimum number of origin server requests before caching an object, this field displays a count of the number of requests received so far.
HTTP Request Method	The method, GET or POST, used in the request that obtained this object.
Stored by policy	The name of the caching policy that caused this object to be stored. A value of NOT AVAILABLE indicates that the policy has been deactivated or deleted. A value of NONE indicates that the object did not match a visible policy, but was stored according to internal criteria for caching.
Application firewall metadata exists	This parameter is used when the application firewall and the integrated cache are both enabled. The application firewall analyzes the contents of a response page, stores its metadata (for example, URLs and forms contained in page), and exports the metadata with the response to the cache. The cache stores the page and the metadata, and when the cache serves the page, it sends the metadata back to the request's session.
HTTP callout object, name, type, response	These cells indicate whether this data was stored as a result of an HTTP Callout expression, and provide information about various aspects of the callout and the corresponding response. For more information about HTTP callouts, see "HTTP Callouts".

To view cached objects by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Cache Objects**. You can view all the cached objects and sort them accordingly as per your requirement.



Find particular cached responses

You can find individual items in the cache based on search criteria. There are different methods for finding cached items, depending on whether the content group that contains the data uses hit and invalidation selectors, as follows:

- If the content group uses selectors, you can only conduct the search using the Locator ID for the cached item.
- If the content group does not use selectors, you conduct the search using criteria such as URL, host, content group name.

When searching for a cached response, you can locate some items by URL and host. If the response is in a content group that uses a selector, you can find it only by using a Locator number (for example, 0x00000000ad7af00000050). To save a Locator number for later use, right-click the entry and select **Copy**. For more information about selectors, see "Configuring Selectors and Basic Content Groups."

To display cached responses in content groups that do not have a selector by using the command line interface

At the command prompt, type:

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET
 | POST ]))| [-httpStatus <positive integer>] | -group <contentGroupName> |
 -ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects (ON | OFF)]
```

To display cached responses in content groups that have a selector by using the command line interface

At the command prompt, type:

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )| -
includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer>]
```


To display cached responses in content groups that do not have a selector by using the configuration utility

Navigate to **Optimization > Integrated Caching > Cache Objects**, click Search, and set the search criteria to view the required cached response.

If you have not yet configured any content groups, all of the objects are in the Default group.

View cache statistics

The following table summarizes the detailed cache statistics that you can view.

Counter	Description
Hits	Responses that are found in and served from the integrated cache. Includes static objects such as image files, pages with status codes 200, 203, 300, 301, 302, 304, 307, 403, 404, 410, and responses that match a user-defined policy with a CACHE action.
Misses	Intercepted HTTP requests where the response was ultimately fetched from origin server.
Requests	Total cache requests plus total cache misses.
Non-304 hits	If the user requests an item more than once, and the item in the cache is unchanged since the last time the Citrix ADC appliance served it, the Citrix ADC appliance serves a 304 response instead of the cached object.
This statistic indicates how many items the Citrix ADC appliance served from the cache, excluding 304 responses.	
304 hits	Number of 304 (object not modified) responses the Citrix ADC appliance served from the cache.
304 hit ratio (%)	Percentage of 304 responses that the Citrix ADC appliance served, relative to other responses.
Hit ratio (%)	Percentage of responses that the Citrix ADC appliance served from the cache (cache requests) relative to responses that could not be served from the cache.

Counter	Description
Origin bandwidth saved (%)	An estimate of the processing capacity that the Citrix ADC appliance saved on the origin server due to serving responses from the cache.
Bytes served by the Citrix ADC	Total number of bytes that the Citrix ADC appliance served from the origin server and the cache.
Bytes served by cache	Total number of bytes that the Citrix ADC appliance served from the cache.
Byte hit ratio(%)	Percentage of data that the Citrix ADC appliance served from the cache, relative to all of the data in all served responses.
Compressed bytes from cache	Amount of data, in bytes, that the Citrix ADC appliance served in compressed form.
Storable misses	If the Citrix ADC appliance does not find a requested object in the cache, it fetches the object from the origin server. This is known as a cache miss. A storable cache miss can be stored in the cache.
Non-storable misses	A non-storable cache miss cannot be stored in the cache.
Misses	All cache misses.
Revalidations	Max-Age setting in a Cache-Control header determines, in number of seconds, when an intervening cache must revalidate the content with the integrated cache before serving it to the user.
For more information, see “Inserting a Cache-Control Header.”	
Successful revalidations	Number of revalidations that have been performed.
For more information, see “Inserting a Cache-Control Header.”	
Conversions to conditional req	A user-agent request for a cached PET object is always converted to a conditional request and sent to the origin server.

Counter	Description
For more information, see “Polling the Origin Server Every Time a Request Is Received.”	
Storable miss ratio (%)	Storable cache misses as a percentage of non-storable cache misses.
Successful reval ratio (%)	Successful revalidations as a percentage of all revalidation attempts.
For more information, see “Inserting a Cache-Control Header.”	
Expire at last byte	Number of times that the cache expired content immediately after receiving the last body byte. Only applicable to positive responses, as described in the table “Cache Hits and Misses.”
For more information, see “Example of Performance Optimization.”	
`Flashcache` misses	If you enable Flash Cache, the cache allows only one request to reach the server, eliminating flash crowds. This statistic indicates the number of Flash Cache requests that were cache misses.
For more information, “Queuing Requests to the Cache.”	
Flashcache hits	Number of Flash Cache requests that were cache hits.
For more information, see “Queuing Requests to the Cache.”	
Parameterized inval requests	Requests that match a policy with an invalidation (INVAL) action and a content group that uses an invalidation selector or parameters to selectively expire cached objects in the group.
Full inval requests	Requests that match an invalidation policy where the invalGroups parameter is configured and expires one or more content groups.

Counter	Description
Inval requests	Requests that match an invalidation policy and result in expiration of specific cached responses or entire content groups.
Parameterized requests	Number of cache requests that were processed using a policy with a parameterized content group.
Parameterized non-304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where full cached response was found, and the response was not a 304 (object not updated) response.
Parameterized 304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found, and the object was a 304 (object not updated) response.
Total parameterized hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found.
Parameterized 304 hit ratio (%)	Percentage of 304 (object not updated) responses that were found using a parameterized policy, relative to all cache hits.
Poll every time requests	If Poll Every Time is enabled, the Citrix ADC appliance always consults the origin server before serving a stored object.
For more information, see “Polling the Origin Server Every Time a Request Is Received.”	
Poll every time hits	Number of times a cache hit was found using the Poll Every Time method.
For more information, see “Polling the Origin Server Every Time a Request Is Received.”	

Counter	Description
Poll every time hit ratio (%)	Percentage of cache hits using the Poll Every Time method, relative to all searches for cached objects using Poll Every Time. For more information, see “Polling the Origin Server Every Time a Request Is Received.”
Maximum memory (KB)	Maximum amount of memory in the Citrix ADC appliance that is allocated to the cache. For more information, see “Configuring Global Attributes for Caching.”
Maximum memory active value (KB)	Maximum amount of memory (active value) that will be set after the memory is allocated to the cache. For more information, see “How to Configure the Integrated Caching Feature of a Citrix ADC Appliance for various Scenarios.”
Utilized memory (KB)	Amount of memory that is actually being used.
Memory allocation failures	Number of failed attempts to utilize memory for the purpose of storing a response in the cache.
Largest response so far	Largest response in bytes found in either the cache or the origin server and sent to the client.
Cached objects	Number of objects in the cache, including responses that have not yet been fully downloaded and responses that have been expired but not yet flushed.
Marker objects	Marker objects are created when a response exceeds the maximum or minimum response size for the content group, or has not yet received the minimum number of hits for the content group.
Hits being served	Number of hits that have been served from the cache.

Counter	Description
Misses being handled	Responses that were fetched from the origin server, stored in the cache, and then served. Should approximate the number for storable misses. Does not include non-storable misses.

To view summary cache statistics by using the command line interface:

At the command prompt, type:

```
stat cache
```

To view specific cache statistics by using the command line interface:

At the command prompt, type:

```
stat cache -detail
```

```
> stat cache -detail

Integrated Cache Statistics - Detail
Integrated Cache Statistics - Summary
                                                Rate (/s)
                                                   Total


Hits                                                  0
                        0

Misses                                                0
                        0

Requests                                              0
                        0

Hit ratio(%)                                         --
                        0

Origin bandwidth saved(%)                            --
                        0
Cached objects                                       --
                        0

Marker objects                                       --
                        0
```



```
20                                                Rate (/s)
                                                     Total
21
22 Requests                                                0
                         0
23
24
25 Hit Statistics
26
27                                                Rate (/s)
                                                     Total
28
29
30 Non-304 hits                                            0
                         0
31
32 304 hits                                                0
                         0
33
34
35 Sql hits                                                0
                         0
36
37
38 Hits                                                    0
                         0
39
40 304 hit ratio(%)                                       --
                         0
41
42 Hit ratio(%)                                           --
                         0
43
44 Origin bandwidth saved(%)                              --
                         0
45 Byte Statistics
46                                                Rate (/s)
                                                     Total
47
48
49 Bytes served by Citrix ADC                            648
    55379204
50
51 Bytes served by cache                                   0
                         0
```



```
52 Byte hit ratio(%)                                --
                        0
53 Compressed bytes from cache                       0
                        0
54
55 Miss Statistics
56
57                                                Rate (/s)
                                                     Total
58
59
60 Storable misses                                   0
                        0
61
62 Non-storable misses                               0
                        0
63
64 Misses                                            0
                        0
65
66 Revalidations                                     0
                        0
67
68 Successful revalidations                          0
                        0
69
70 Conversions to conditional req                    0
                        0
71
72
73 Storable miss ratio(%)                           --
                        0
74 Successful reval ratio(%)                        --
                        0
75
76 Flashcache Statistics
77                                                Rate (/s)
                                                     Total
78
79
80 Expire at last byte                               0
                        0
81
82 Flashcache misses                                 0
                        0
```



```
Flashcache hits                                        0
                        0

Invalidation Statistics

                                                 Rate (/s)
                                                    Total

Parameterized inval requests                           0
                        0


Full inval requests                                    0
                        0


Inval requests                                         0
                        0

Parameterized Caching Statistics

                                                 Rate (/s)
                                                    Total


Parameterized requests                                 0
                        0

Parameterized non-304 hits                             0
                        0

Parameterized 304 hits                                 0
                        0


Total parameterized hits                               0
                        0

Parameterized 304 hit ratio(%)                        --
                        0

Poll Every Time (PET) Statistics

                                                 Rate (/s)
```



```
                                                    Total
117
118
119 Poll every time requests                              0
                         0
120
121 Poll every time hits                                  0
                         0
122
123 Poll every time hit ratio(%)                         --
                         0
124
125 Memory Usage Statistics
126                                                  Total
127
128 Maximum memory(KB)                                    0
129
130 Maximum memory active value(KB)                       0
131
132 Utilized memory(KB)                                   0
133
134 Memory allocation failures                            0
135
136 Largest response so far(B)                            0
137
138 Cached objects                                        0
139
140 Marker objects                                        0
141
142 Hits being served                                     0
143 Misses being handled                                  0
144  Done
145 <!--NeedCopy-->
```

To view summary cache statistics by using the GUI

1. Click the **Dashboard** tab at the top of the page.
2. Scroll down to the **Integrated Caching** section of the window.
3. To see detailed statistics, click the More… link at the bottom of the table.

To view specific cache statistics by using the GUI

1. Click the **Reporting** tab at the top of the page.
2. Under Built-In Reports, expand **Integrated Cache**, and then click the report with the statistics you want to view.

3. To save the report as a template, click **Save As** and name the report. The saved report appears under **Custom** Reports.

Display cached objects and cache statistics

September 14, 2021

You can view particular cached objects, and you can view summary statistics on cache hits, misses, and memory usage. The statistics provide insight on the amount of data that is being served from the cache, what items are responsible for the largest performance benefit, and what you can tune to improve cache performance.

This section includes the following details:

- Viewing Cached Objects
- Finding Particular Cached Responses
- Viewing Cache Statistics

View cached objects

After enabling caching, you can view details for cached objects. For example, you can view the following items:

- Response sizes and header sizes
- Status codes
- Content groups
- ETag, Last-Modified, and Cache-Control headers
- Request URLs
- Hit parameters
- Destination IP addresses
- Request and response times

To view a list of cached objects by using the command line interface

At the command prompt, type:

```
show cache object
```

Properties	Specification
Response size (bytes)	The size of the response header and body.
Response header size (bytes)	The size of the header portion of the response.

Properties	Specification
Response status code	The status code sent with the response.
ETag	The ETag header inserted in the response. Typically, this header indicates whether the response has changed recently.
Last-Modified	The Last-Modified header inserted in the response. This header indicates the date that the response was last changed.
Cache-Control	The Cache-Control header inserted in the response.
Date	The Date header that indicates when the response was sent.
Contentgroup	The content group where the response is stored.
Complex match	If this object was cached based on parameterized values, this field value is YES.
Host	The host specified in the URL that requested this response.
Host port	The listen port for the host specified in the URL that requested this response
URL	The URL issued for the stored response.
Destination IP	The IP address of the server from which this response was fetched.
Destination port	The listen port for the destination server.
Hit parameters	If the content group that stores the response uses hit parameters, they are listed in this field.
Hit selector	If this content group uses a hit selector, it is listed in this field.
Inval selector	If this content group uses an invalidation selector, it is listed in this field.
Selector Expressions	If this content group uses a selector, this field displays the expression that defines the selection rule.
Request time	The time in milliseconds since the request was issued.

Properties	Specification
Response time	The time in milliseconds since the cache started to receive the response.
Age	Amount of time the object has been in the cache.
Expiry	Amount of time after which the object is marked as expired.
Flushed	Whether the response has been flushed after expiry.
Prefetch	If Prefetch has been configured for this content group, the amount of time before expiry during which the object is fetched from the origin. Prefetch does not apply to negative objects (for example, 404 “object not found” responses).
Current readers	Approximately the current number of hits being served. When a response with a Content-Length header object is being downloaded, the current misses and the current readers values are each typically 1. When a chunked response object is being downloaded, the current misses value is typically 1, but the current readers value is typically 0, because the chunked response that is served to the client does not come from the integrated caching buffers.
Current misses	The current number of requests that resulted in a cache miss and fetching from the origin server. This value is typically 0 or 1. If Poll Every Time is enabled for a content group, the count can be greater than 1.
Hits	The number of cache hits for this object.
Misses	The number of cache misses for this object.
Compression format	The type of compression applied to this object. Compression formats include gzip, deflate, compress, and pack200-gzip.

Properties	Specification
HTTP version in response	The version of HTTP that was used to send the response.
Weak `etag` present in response	Strong `etag` headers change if the bits of an entity change. Strong headers are based on the octet values of an object. Weak `etag` headers change if the meaning of an entity changes. Weak `etag` values are based on semantic identity. Weak `etags` values start with a “W.”
Negative marker cell	A marker object is cacheable, but it does not yet meet all the criteria for being cached. For example, the object may exceed the maximum response size for the content group. A marker cell is created for objects of this type. The next time a user sends a request for this object, a cache miss is served.
Reason marker created	The reason a marker cell was created (for example, “Waiting for minhit,” “Content-length response data is not in group size limit”).
Auto poll every time	If the integrated cache receives an already expired 200 OK response with validators (either the Last-Modified or the `ETag` response headers) it stores the response and marks it as Auto-PET (automatically poll every time).
Citrix ADC Etag inserted in response	A variation of the `ETag` header generated by the Citrix ADC appliance. A value of YES appears if the Citrix ADC inserts an `Etag` in the response.
Full response present in cache	Indicates whether this is a complete response.
Destination IP verified by DNS	Indicates whether DNS resolution was performed when storing the object.
Object stored through a cache forward proxy	Indicates whether this response was stored due to a forward proxy that is configured in the integrated cache.
Object is a Delta basefile	A response that is delta-compressed.

Properties	Specification
Waiting for minhits	Indicates whether this content group requires a minimum number of origin servers hit before caching a response.
Minhit count	If this content group requires a minimum number of origin servers hit before caching an object, this field displays a count of the number of hits received so far.
HTTP Request Method	The method, GET or POST, used in the request that obtained this object.
Stored by policy	The name of the caching policy that caused this object to be stored. A value of NOT AVAILABLE indicates that the policy has been deactivated or deleted. A value of NONE indicates that the object did not match a visible policy, but was stored according to internal criteria for caching.
Application firewall metadata exists	This parameter is used when the application firewall and the integrated cache are both enabled. The application firewall analyzes the contents of a response page, stores its metadata (for example, URLs and forms contained in page), and exports the metadata with the response to the cache. The cache stores the page and the metadata, and when the cache serves the page, it sends the metadata back to the request's session.
HTTP callout object, name, type, response	These cells indicate whether this data was stored as a result of an HTTP Callout expression, and provide information about various aspects of the callout and the corresponding response. For more information about HTTP callouts, see "HTTP Callouts".

Find particular cached responses

You can find individual items in the cache based on search criteria. There are different methods for finding cached items, depending on whether the content group that contains the data uses hit and invalidation selectors, as follows:

If the content group uses selectors, you can only conduct the search using the Locator ID for the cached item.

If the content group does not use selectors, you conduct the search using criteria such as URL, host, content group name.

When searching for a cached response, you can locate some items by URL and host. If the response is in a content group that uses a selector, you can find it only by using a Locator number (for example, 0x00000000ad7af00000050). To save a Locator number for later use, right-click the entry and select Copy. For more information about selectors, see "Configuring Selectors and Basic Content Groups."

To display cached responses in content groups that do not have a selector by using the command line interface

At the command prompt, type:

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET
 | POST ]))| [-httpStatus<positive integer>] | -group <contentGroupName> |
-ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects (ON | OFF)]
```

To display cached responses in content groups that have a selector by using the command line interface

At the command prompt, type:

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )| -
includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer>]
```

To display cached responses in content groups that do not have a selector by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Cache Objects**, click **Search**, and set the search criteria to view the required cached response.

If you have not yet configured any content groups, all of the objects are in the Default group.

To display cached responses in content groups that have a selector by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Cache Objects**, click **Search**, and set the selector search criteria to view the required cached response.

View cache statistics

The following table summarizes the cache statistics.

Counter

Specification

Viewing cache statistics

Updated: 2013-10-28

The following table summarizes the detailed cache statistics that you can view.

Counter	Specifies
Hits	Responses that are found in and served from the integrated cache. Includes static objects such as image files, pages with status codes 200, 203, 300, 301, 302, 304, 307, 403, 404, 410, and responses that match a user-defined policy with a CACHE action..
Misses	Intercepted HTTP requests where the response was ultimately fetched from origin server.
Requests	Total cache hits plus total cache misses.
Non-304 hits	If the user requests an item more than once, and the item in the cache is unchanged since the last time the Citrix ADC appliance served it, the Citrix ADC appliance serves a 304 response instead of the cached object. This statistic indicates how many items the Citrix ADC appliance served from the cache, excluding 304 responses.
304 hits	Number of 304 (object not modified) responses the Citrix ADC appliance served from the cache.
304 hit ratio (%)	Percentage of 304 responses that the Citrix ADC appliance served, relative to other responses.
Hit ratio (%)	Percentage of responses that the Citrix ADC appliance served from the cache (cache hits) relative to responses that could not be served from the cache.
Origin bandwidth saved (%)	An estimate of the processing capacity that the Citrix ADC appliance saved on the origin server due to serving responses from the cache.

Counter	Specifies
Bytes served by the Citrix ADC	Total number of bytes that the Citrix ADC appliance served from the origin server and the cache.
Bytes served by cache	Total number of bytes that the Citrix ADC appliance served from the cache.
Byte hit ratio(%)	Percentage of data that the Citrix ADC appliance served from the cache, relative to all of the data in all served responses.
Compressed bytes from cache	Amount of data, in bytes, that the Citrix ADC appliance served in compressed form.
Storable misses	If the Citrix ADC appliance does not find a requested object in the cache, it fetches the object from the origin server. This is known as a cache miss. A storable cache miss can be stored in the cache.
Non-storable misses	A non-storable cache miss cannot be stored in the cache.
Misses	All cache misses.
Revalidations	Max-Age setting in a Cache-Control header determines, in number of seconds, when an intervening cache must revalidate the content with the integrated cache before serving it to the user. For more information, see “Inserting a Cache-Control Header.”
Successful revalidations	Number of revalidations that have been performed. For more information, see “Inserting a Cache-Control Header.”
Conversions to conditional req	A user-agent request for a cached PET object is always converted to a conditional request and sent to the origin server. For more information, see “Polling the Origin Server Every Time a Request Is Received.”
Storable miss ratio (%)	Storable cache misses as a percentage of non-storable cache misses.

Counter	Specifies
Successful reval ratio (%)	Successful revalidations as a percentage of all revalidation attempts. For more information, see “Inserting a Cache-Control Header.”
Expire at last byte	Number of times that the cache expired content immediately after receiving the last body byte. Only applicable to positive responses, as described in the table “Cache Hits and Misses.”For more information, see “Example of Performance Optimization.”
Flash cache misses	If you enable Flash Cache, the cache allows only one request to reach the server, eliminating flash crowds. This statistic indicates the number of Flash Cache requests that were cache misses. For more information, “Queuing Requests to the Cache.”
`Flashcache` hits	Number of Flash Cache requests that were cache hits. For more information, see “Queuing Requests to the Cache.”
Parameterized inval requests	Requests that match a policy with an invalidation (INVAL) action and a content group that uses an invalidation selector or parameters to selectively expire cached objects in the group.
Full inval requests	Requests that match an invalidation policy where the invalGroups parameter is configured and expires one or more content groups.
Inval requests	Requests that match an invalidation policy and result in expiration of specific cached responses or entire content groups.
Parameterized requests	Number of cache requests that were processed using a policy with a parameterized content group.

Counter	Specifies
Parameterized non-304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where full cached response was found, and the response was not a 304 (object not updated) response.
Parameterized 304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found, and the object was a 304 (object not updated) response.
Total parameterized hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found.
Parameterized 304 hit ratio (%)	Percentage of 304 (object not updated) responses that were found using a parameterized policy, relative to all cache hits.
Poll every time requests	If Poll Every Time is enabled, the Citrix ADC appliance always consults the origin server before serving a stored object. For more information, see “Polling the Origin Server Every Time a Request Is Received.”
Poll every time hits	Number of times a cache hit was found using the Poll Every Time method. For more information, see “Polling the Origin Server Every Time a Request Is Received.”
Poll every time hit ratio (%)	Percentage of cache hits using the Poll Every Time method, relative to all searches for cached objects using Poll Every Time. For more information, see “Polling the Origin Server Every Time a Request Is Received.”
Maximum memory (KB)	Maximum amount of memory in the Citrix ADC appliance that is allocated to the cache. For more information, see “Configuring Global Attributes for Caching.”

Counter	Specifies
Maximum memory active value (KB)	Maximum amount of memory (active value) that will be set after the memory is actually allocated to the cache. For more information, see “How to Configure the Integrated Caching Feature of a Citrix ADC Appliance for various Scenarios.”
Utilized memory (KB)	Amount of memory that is actually being used.
Memory allocation failures	Number of failed attempts to utilize memory for the purpose of storing a response in the cache.
Largest response so far	Largest response in bytes found in either the cache or the origin server and sent to the client.
Cached objects	Number of objects in the cache, including responses that have not yet been fully downloaded and responses that have been expired but not yet flushed.
Marker objects	Marker objects are created when a response exceeds the maximum or minimum response size for the content group, or has not yet received the minimum number of hits for the content group.
Hits being served	Number of hits that have been served from the cache.
Misses being handled	Responses that were fetched from the origin server, stored in the cache, and then served. Should approximate the number for storable misses. Does not include non-storable misses.

To view summary cache statistics by using the command line interface

At the command prompt, type:

```
stat cache
```

To view specific cache statistics by using the command line interface

At the command prompt, type:


```
stat cache -detail

    > stat cache -detail
    Integrated Cache Statistics - Detail
    Integrated Cache Statistics - Summary
                                                Rate (/s)
                                                   Total
    Hits                                                0
                           0
    Misses                                              0
                           0
    Requests                                            0
                           0
    Hit ratio(%)                                       --
                           0
    Origin bandwidth saved(%)                          --
                           0
    Cached objects                                     --
                           0
    Marker objects                                     --
                           0
                                                Rate (/s)
                                                   Total
    Requests                                            0
                           0
    Hit Statistics
                                                Rate (/s)
                                                   Total
    Non-304 hits                                        0
                           0
    304 hits                                            0
                           0
    Sql hits                                            0
                           0
    Hits                                                0
                           0
    304 hit ratio(%)                                   --
                           0
    Hit ratio(%)                                       --
                           0
    Origin bandwidth saved(%)                          --
                           0

    Byte Statistics
```



```
27                                                  Rate (/s)
                                                       Total
28      Bytes served by Citrix ADC                          648
           55379204
29      Bytes served by cache                                 0
                              0
30      Byte hit ratio(%)                                    --
                              0
31      Compressed bytes from cache                           0
                              0
32      Miss Statistics
33                                                  Rate (/s)
                                                       Total
34      Storable misses                                       0
                              0
35      Non-storable misses                                   0
                              0
36      Misses                                                0
                              0
37      Revalidations                                         0
                              0
38      Successful revalidations                              0
                              0
39      Conversions to conditional req                        0
                              0
40      Storable miss ratio(%)                               --
                              0
41      Successful reval ratio(%)                            --
                              0
42      Flashcache Statistics
43                                                  Rate (/s)
                                                       Total
44      Expire at last byte                                   0
                              0
45      Flashcache misses                                     0
                              0
46      Flashcache hits                                       0
                              0
47
48      Invalidation Statistics
49                                                  Rate (/s)
                                                       Total
50      Parameterized inval requests                          0
                              0
51      Full inval requests                                   0
```



```
                              0
52      Inval requests                                      0
                              0
53
54      Parameterized Caching Statistics
55                                                  Rate (/s)
                                                       Total
56      Parameterized requests                              0
                              0
57      Parameterized non-304 hits                          0
                              0
58      Parameterized 304 hits                              0
                              0
59      Total parameterized hits                            0
                              0
60      Parameterized 304 hit ratio(%)                     --
                              0
61
62      Poll Every Time (PET) Statistics
63                                                  Rate (/s)
                                                       Total
64      Poll every time requests                            0
                              0
65      Poll every time hits                                0
                              0
66      Poll every time hit ratio(%)                       --
                              0
67      Memory Usage Statistics
68                                                     Total
69      Maximum memory(KB)                                  0
70      Maximum memory active value(KB)                     0
71      Utilized memory(KB)                                 0
72      Memory allocation failures                          0
73      Largest response so far(B)                          0
74      Cached objects                                      0
75      Marker objects                                      0
76      Hits being served                                   0
77      Misses being handled                                0
78       Done
79  <!--NeedCopy-->
```

To view summary cache statistics by using the GUI

1. Click the **Dashboard** tab at the top of the page.
2. Scroll down to the Integrated Caching section of the window.

3. To see detailed statistics, click the More… link at the bottom of the table.

To view specific cache statistics by using the GUI

1. Click the Reporting tab at the top of the page.
2. Under Built-In Reports, expand Integrated Cache, and then click the report with the statistics you want to view.
3. To save the report as a template, click Save As and name the report. The saved report appears under Custom Reports.

Improve cache performance

September 14, 2021

You can improve the performance of the integrated cache, including handling simultaneous requests for the same cached data, avoiding delays that are associated with refreshing cached responses from the origin server, and ensuring that a response is requested often enough to be worth caching.

Reduce flash crowds

Flash crowds occur when many users simultaneously request the same data. The requests in a flash crowd can become cache misses if you configured the cache to serve hits only after the entire object is downloaded.

The following techniques can reduce or eliminate flash crowds:

- **PREFETCH**: Refreshes a positive response before it expires to ensure that it never becomes stale or inactive. For more information, see “Refreshing a Response Prior to Expiration” section.
- **Cache buffering**: Starts serving a response to multiple clients when it receives the response header from the origin server, rather than waiting for the entire response to be downloaded. The only limit on the number of clients that can download a response simultaneously is the available system resources. The Citrix ADC appliance downloads and serves responses even if the client that initiated the download halts before the download is complete. If the response exceeds the cache size or if the response is chunked, the cache stops storing the response, but service to the clients is not disrupted.
- **Flash Cache:** Flash Cache queues requests to the cache, and allows only one request to reach the server at a time.

For more information, see “Queuing Requests to the Cache” section.

Refresh a response before expiration

To ensure that a cached response is fresh whenever it is needed, the PREFETCH option refreshes a response before its calculated expiration time. The prefetch interval is calculated after receiving the first client request. From that point onward, the Citrix ADC appliance refreshes the cached response at a time interval that you configure in the PREFETCH parameter.

This setting is useful for data that is updated frequently between requests. It does not apply to negative responses (for example, 404 messages).

To configure prefetch for a content group by using the command line interface

At the command prompt, type:

```
set cache contentgroup <name> -prefetch YES [-prefetchPeriod <seconds> | -
prefetchPeriodMilliSec <milliseconds>] [-prefetchMaxPending <positiveInteger
>]
```

*To configure prefetch for a content group by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the **content group**.

On **Others** tab, in the Flash Crowd and Prefetch group, select **Prefetch** option, and specify the values in the Interval and Maximum number of pending prefetches text boxes.

Queue requests to the cache

The Flash Cache option queues requests that arrive simultaneously (a flash crowd), retrieves the response, and distributes it to all the clients whose requests are in the queue. If, during this process, the response becomes non-cacheable, the Citrix ADC appliance stops serving the response from the cache and instead serves the origin server's response to the queued clients. If the response is not available, the clients receive an error message.

Flash Cache is disabled by default. You cannot enable Poll Every Time (PET) and Flash Cache on the same content group.

One disadvantage of Flash Cache is if the server replies with an error (for example, a 404 that is quickly remedied), the error is fanned out to the waiting clients.

> **Note**:
> If Flash Cache is enabled, in some situations the Citrix ADC appliance is unable to correctly match the Accept-Encoding header in the client request with the Content-Encoding header in the response. The Citrix ADC appliance can assume that these headers match and mistakenly serve a hit. As a work-around, you can configure Integrated Caching policies to disallow serving hits to clients that do not have an appropriate Accept-Encoding header.

To enable Flash Cache by using the command line interface

At the command prompt, type:

```
set cache contentgroup <contentGroupName> -flashcache yes
```

To enable Flash Cache by using the GUI

Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the content group.

On **Others** tab, in the Flash Crowd and Prefetch group, select **Prefetch** option.

Cache a response after a client halts a download

You can set the Quick Abort parameter to continue caching a response, even if the client halts a request before the response is in the cache.

If the downloaded response size is less than or equal to the Quick Abort size, the Citrix ADC appliance stops downloading the response. If you set the Quick Abort parameter to 0, all downloads are halted.

To configure quick abort size by using the command line interface

At the command prompt, type:

```
set cache contentgroup <name> -quickAbortSize <integerInKBytes>
```

To configure quick abort size by using the GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the content group.
2. On **Memory** tab, set the relevant value in Quick Abort: Continue caching if more than text box.

Requiring a minimum number of server hits before caching

You can configure the minimum number of times that a response must be found on the origin server before it can be cached. You must consider increasing the minimum hits if the cache memory fills up quickly and has a lower-than-expected hit ratio.

The default value for the minimum number of hits is 0. This value caches the response after the first request.

To configure the minimum number of hits that are required before caching by using the command line interface

At the command prompt, type:

```
set cache contentgroup <name> -minhits <positiveInteger>
```

To configure the minimum number of hits that are required before caching by using the GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the content group.
2. On **Memory** tab, set the relevant value in Do not cache, if hits are less than the text box.

Example for performance optimization

In this example, a client accesses a stock quote. Stock quotes are highly dynamic. You configure the integrated cache to serve the same stock quote to concurrent clients without sending multiple requests to the origin server. The stock quote expires after it is downloaded to the clients, and the next request is fetched from the origin server. This ensures that the quote is always up to date.

The following task overview describes the steps to configure the cache for the stock quote application.

Configure caching for a stock quote application

Create a content group for stock quotes

For more information, see “About Content Groups.”

Configure the following for this content group:

1. On the **Expiry Method** tab, select the Expire after complete response received check box.
2. On the **Others** tab, select the **Flash Cache** check box, and click **Create**.
3. Add a cache policy to cache the stock quotes.

For more information, see “Configuring a Policy in the Integrated Cache.”

Configure the following for the policy

1. In the **Action and Store in Group lists**, select **CACHE** and select the group that you defined in the previous step.
2. Click **Add**, and in the **Add Expression** dialog box configure an expression that identifies stock quote requests, for example: http.req.url.contains(“cgi-bin/stock-quote.pl”)
3. Activate the policy.

For more information, see “Globally Binding an Integrated Caching Policy.” In this example, you bind this policy to request-time override processing and set the priority to a low value.

Configure cookies, headers, and polling

October 27, 2021

This topic explains how to configure cache manages cookies, HTTP headers, and origin server polling. This includes modifying the default behavior that causes the cache to diverge from documented standards, overriding HTTP headers that might cause cacheable content to not be stored in the cache, and configuring the cache to always poll the origin for updated content.

Divergence of cache behavior from the standards

By default, the integrated cache adheres to the following RFC standards:

- RFC 2616, “HTTP HTTP/1.1”
- The caching behaviors described in RFC 2617, “HTTP Authentication: Basic and Digest Access Authentication”
- The caching behavior described in RFC 2965, “HTTP State Management Mechanism”

The built-in policies and the Default content group attributes ensure conformance with most of these standards.

The default integrated cache behavior diverges from the specification as follows:

- There is a limited support for the Vary header. By default, any response containing a Vary header is considered to be non-cacheable unless it is compressed. A compressed response contains content-encoding: gzip, content-encoding: deflate, or content-encoding: pack200-gzip and is cacheable even if it contains the Vary: Accept-encoding header.
- The integrated cache ignores the values of the headers cache-control: no-cache and cache-control: private. For example, a response that contains cache-control: no-cache=”Set-Cookie” is treated as if the response contained Cache-Control: no-cache. By default, the response is not cached.
- An image (content-type = image/*) is always considered cacheable, even if an image response contains set-cookie or set-cookie2 headers, or if an image request contains a cookie header. The integrated cache removes set-cookie and set-cookie2 headers from a response before caching it. This diverges from RFC 2965. You can configure RFC-compliant behavior as follows:

```
1 add cache policy rfc_compliant_images_policy -rule "http.res.header.set
     -cookie2.exists || http.res.header.set-cookie.exists" -action
     NOCACHE
2
3
4 bind cache global rfc_compliant_images_policy -priority 100 -type REQ\
     _OVERRIDE
5 <!--NeedCopy-->
```

- The following cache-control headers in a request force an RFC-compliant cache to reload a cached response from the origin server:

```
Cache-control: max-age=0
```

```
Cache-control: no-cache
```

To guard against denial-of-service attacks, this behavior is not the default.

- By default, the caching module considers a response to be cacheable unless a response header state otherwise. To make this behavior RFC 2616 compliant, set `-weakPosRelExpiry` and `-weakNegResExpiry` to 0 for all content groups.

Remove cookies from a response

Cookies are often personalized for a user, and typically should not be cached. The `Remove Response Cookies` parameter removes `Set-Cookie and Set-Cookie2` headers before caching a response. By default, the `Remove Response Cookies` option for a content group prevents caching of responses with `Set-Cookie` or `Set-Cookie2` headers.

> **Note:**
> When images are cached, the built-in behavior is to remove the `Set-Cookie` and `Set-Cookie2` headers before caching, no matter how the content group is configured.
>
> Citrix recommends that you accept the default `Remove Response Cookies` for every content group that stores embedded responses, for example, images.

To configure `Remove Response Cookies` for a content group by using the command line interface:

At the command prompt, type:

```
set cache contentgroup <name> -removeCookies YES
```

Configure Remove Response Cookies for a content group by using Citrix ADC GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the content group.
2. On the **Others** tab, in the **Settings** group, select Remove response cookies option.

Inserting HTTP headers at response time

The integrated cache can insert HTTP headers in responses that result from cache requests. The Citrix ADC appliance does not alter headers in responses that result from cache misses.

The following table describes headers that you can insert in a response.

Header	Specification
Age	Provides the age of the response in seconds, calculated from the time the response was generated at the origin server. By default, the cache inserts an Age header for every response that is served from the cache.

Header	Specification
via	Lists protocols and recipients between the start and end points for a request or a response. The Citrix ADC appliance inserts a Via header in every response that it serves from the cache. The default value of the inserted header is `NS-CACHE-10.0`: last octet of the Citrix ADC IP address." For more information, see "Configuring Global Attributes for Caching."
`Tag`	The cache supports response validation using Last-Modified and `Tag` headers to determine if a response is stale. The cache inserts an `Tag` in a response only if it caches the response and the origin server has not inserted its own `Tag` header. The `Tag` value is an arbitrary unique number. The `Tag` value for a response changes if it is refreshed from the origin server, but it stays the same if the server sends a 304 (object not updated) response. Origin servers typically do not generate validators for dynamic content because dynamic content is considered non-cacheable. You can override this behavior. With `Tag` header insertion, the cache is permitted to not serve full responses. Instead, the user agent is required to cache the dynamic response sent by the integrated cache the first time. To force a user agent to cache a response, you configure the integrated cache to insert an `Tag` header and replace the origin-provided Cache-Control header.

Header	Specification
Cache-Control	The Citrix ADC appliance typically does not modify cacheability headers in responses that is serves from the origin server. If the origin server sends a response that is labeled as non-cacheable, the client treats the response as non-cacheable even if the Citrix ADC appliance caches the response. To cache dynamic responses in a user agent, you can replace Cache-Control headers from the origin server. This applies only to user agents and other intervening caches. They do not affect the integrated cache.

Header	Specification
Age	Provides the age of the response in seconds, calculated from the time the response was generated at the origin server. By default, the cache inserts an Age header for every response that is served from the cache.
via	Lists protocols and recipients between the start and end points for a request or a response. The Citrix ADC appliance inserts a Via header in every response that it serves from the cache. The default value of the inserted header is “NS-CACHE-9.2: last octet of the Citrix ADC IP address.” For more information, see “Configuring Global Attributes for Caching.”

Header	Specification
Tag	The cache supports response validation using the Last-Modified and Tag headers to determine if a response is stale. The cache inserts an Tag in a response only if it caches the response and the origin server has not inserted its own Tag header. The Tag value is an arbitrary unique number. The Tag value for a response changes if it is refreshed from the origin server, but it stays the same if the server sends a 304 (object not updated) response. Origin servers typically do not generate validators for dynamic content because dynamic content is considered non-cacheable. You can override this behavior. With Tag header insertion, the cache is permitted to not serve full responses. Instead, the user agent is required to cache the dynamic response sent by the integrated cache the first time. To force a user agent to cache a response, you configure the integrated cache to insert an Tag header and replace the origin-provided Cache-Control header.
Cache-Control	The Citrix ADC appliance typically does not modify cacheability headers in responses that is serves from the origin server. If the origin server sends a response that is labeled as non-cacheable, the client treats the response as non-cacheable even if the Citrix ADC appliance caches the response. To cache dynamic responses in a user agent, you can replace Cache-Control headers from the origin server. This applies only to user agents and other intervening caches. They do not affect the integrated cache.

Insert an age, via, or Tag header

The following procedures describe how to insert Age, Via, and ETag headers.

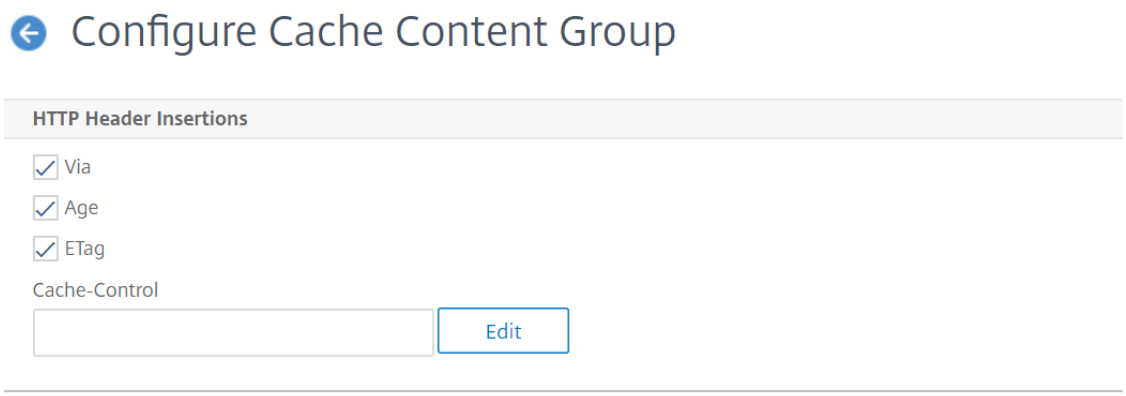
Insert an Age, Via, or Etag header by using the Citrix ADC command interface:

At the command prompt, type:

```
set cache contentgroup <name> -insertVia YES -insertAge YES -insertETag YES
```

Configure the Age, Via, or Etag header by using the Citrix ADC GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups,** and select the **content group**.
2. On the **Others** tab, in the HTTP Header Insertions group, select the **Via**, **Age**, or **ETag** options, as appropriate.
3. The values for the other header types are calculated automatically. You configure the Via value in the main settings for the cache.



Insert a cache-control header

When the integrated cache replaces a Cache-Control header that the origin server inserted, it also replaces the Expires header. The new Expires header contains an expiration time in the past. This ensures that HTTP/1.0 clients and caches (that do not understand the Cache-Control header) do not cache the content.

Insert a cache-control header by using the Citrix ADC command interface

At the command prompt, type:

```
set cache contentgroup <name> -cacheControl <value>
```


Insert a cache-control header by using the Citrix ADC GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and
   a) Click **Expiry Method** to clear the heuristic and default expiry settings and set the relevant value in the Expire content after text box.
   b) Click **Others** tab and type the header you want to insert in the Cache-Control text box. Alternatively, click Configure to set the Cache-Control directives in cached responses.

Ignore cache-control and pragma headers in requests

By default, the caching module processes Cache-Control and Pragma headers. The following tokens in the Cache-Control headers are processed as described in RFC 2616.

- max-age
- max-stale
- only-if-cached
- no-cache

A Pragma: no-cache header in a request is treated in the same way as a Cache-Control: no-cache header.

If you configure the caching module to ignore the Cache-Control and Pragma headers, a request that contains a Cache-Control: No-Cache header causes the Citrix ADC appliance to retrieve the response from the origin server, but the cached response is not updated. If the caching module processes Cache-Control and Pragma headers, the cached response is refreshed.

The following table summarizes the implications of various settings for these headers and the Ignore Browser's Reload Request setting.

Setting for Ignore Cache-Control and Pragma Headers	Setting for Ignore Browser's Reload Request	Outcome
Yes	Yes or No	Ignore the Cache-Control and Pragma headers from the client, including the Cache-Control: no-cache directive.
No	Yes	The Cache-Control: no-cache header produces a cache miss, but a response that is already in the cache is not refreshed.

Setting for Ignore Cache-Control and Pragma Headers	Setting for Ignore Browser's Reload Request	Outcome
No	No	A request that contains a Cache-Control: no-cache header causes a cache miss and the stored response is refreshed.

To ignore Cache-Control and Pragma headers in a request by using the command line interface

At the command prompt, type:

```
set cache contentgroup <name> -ignoreReqCachingHdrs YES
```

To ignore browser reload requests by using the command line interface

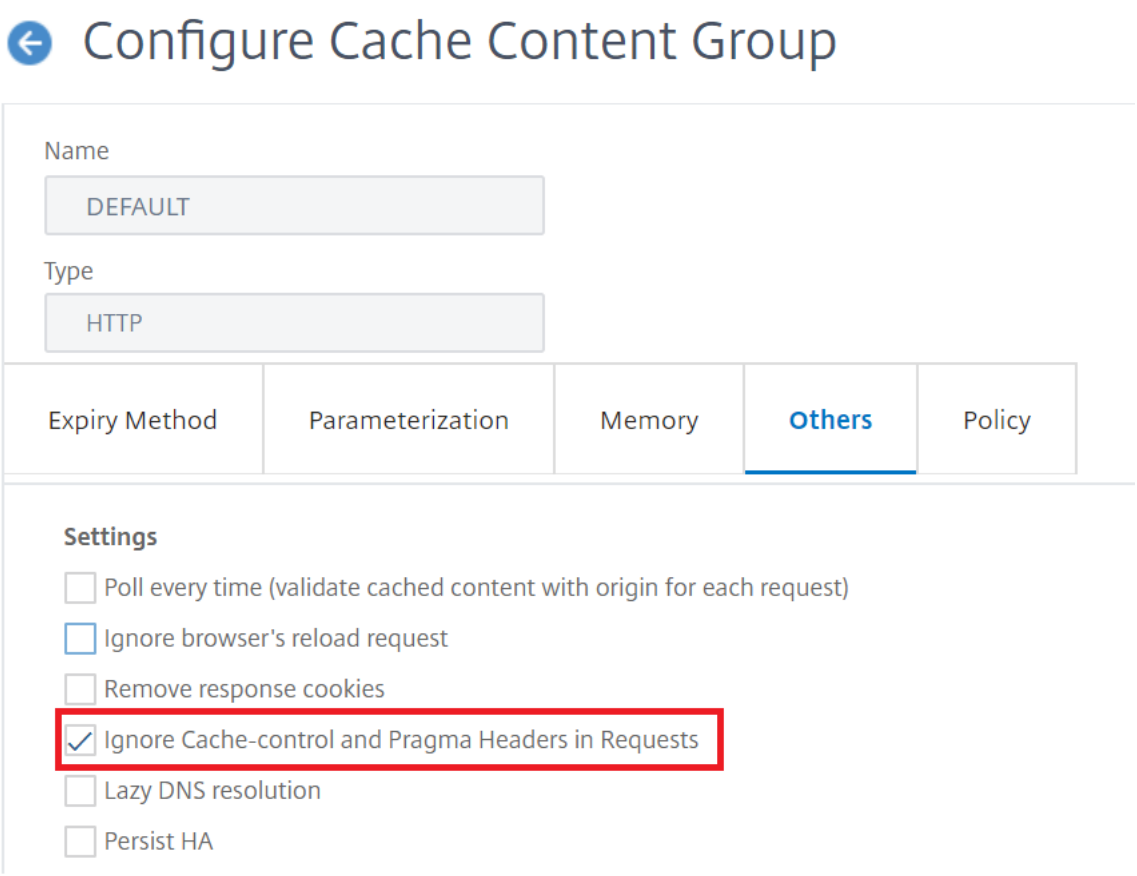
At the command prompt, type:

```
set cache contentgroup <name> -ignoreReloadReq NO
```

> **Note:**
> By default, the -ignoreReloadReq parameter is set to YES.

Ignore Cache-Control and Pragma headers in a request by using the GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the content group.
2. On the **Others** tab, in the **Settings** group, select **Ignore Cache-control and Pragma Headers** in **Requests** option.



Example of a policy to ignore Cache-Control headers:

In the following example, you configure a request-time override policy to cache responses that contain Content-type: image/* regardless of the Cache-Control header in the response.

To configure a request-time override policy to cache all responses with image/*

Flush the cache using the Invalidate All option.

Configure a new cache policy, and direct the policy to a particular content group. For more information, see "Configuring a Policy in the Integrated Cache."

Ensure the content group that the policy uses is configured to ignore Cache-Control headers, as described in "Ignoring Cache-Control and Pragma Headers in Requests."

Bind the policy to the request-time override policy bank.

For more information, see Globally Binding an Integrated Caching Policy topic.

Poll origin server every time a request is received

You can configure the Citrix ADC appliance to always consult the origin server before serving a stored response. This is known as Poll Every Time (PET). When the Citrix ADC appliance consults the origin

server and the PET response has not expired, a full response from the origin server does not overwrite cached content. This property is useful when serving client-specific content.

After a PET response expires, the Citrix ADC appliance refreshes it when the first full response arrives from the origin server.

The Poll Every Time (PET) function works as follows:

For a cached response that has validators in the form of a Tag or a Last-Modified header, if the response expires it is automatically marked PET and cached.

You can configure PET for a content group.

If you configure a content group as PET, every response in the content group is marked PET. The PET content group can store responses that do not have validators. Responses that are automatically marked PET are always expired. Responses that belong to a PET content group can expire after a delay, based on how you configure the content group.

Two types of requests are affected by polling:

- Conditional Requests: A client issues a conditional request to ensure that the response that it has is the most recent copy. A user-agent request for a cached PET response is always converted to a conditional request and sent to the origin server. A conditional request has validators in If-Modified-Since or If-None-Match headers. The If-Modified-Since header contains the time from the Last-Modified header. An If-None-Match header contains the response's Tag header value. If the client's copy of the response is fresh, the origin server replies with 304 Not Modified. If the copy is stale, a conditional response generates a 200 OK that contains the entire response.
- Non-Conditional Requests: A non-conditional request can only generate a 200 OK that contains the entire response.

Origin Server Response	Action
Send the full response	The origin server sends the response as-is to the client. If the cached response has expired, it is refreshed.
304 Not Modified	The following header values in the 304 response are merged with the cached response and the cached response is served to the client: Date, Expires, Age, Cache-Control header Max-Age, and S-Maxage tokens
401 Unauthorized; 400 Bad Request; 405 Method Not Allowed; 406 Not Acceptable; 407 Proxy Authentication Required	The origin's response is served as-is to the client. The cached response is not changed.

Origin Server Response	Action
Any other error response, for example, 404 Not Found	The origin's response is served as-is to the client. The cached response is removed.

> **Note:**
> The Poll Every Time parameter treats the affected responses as non-storable.

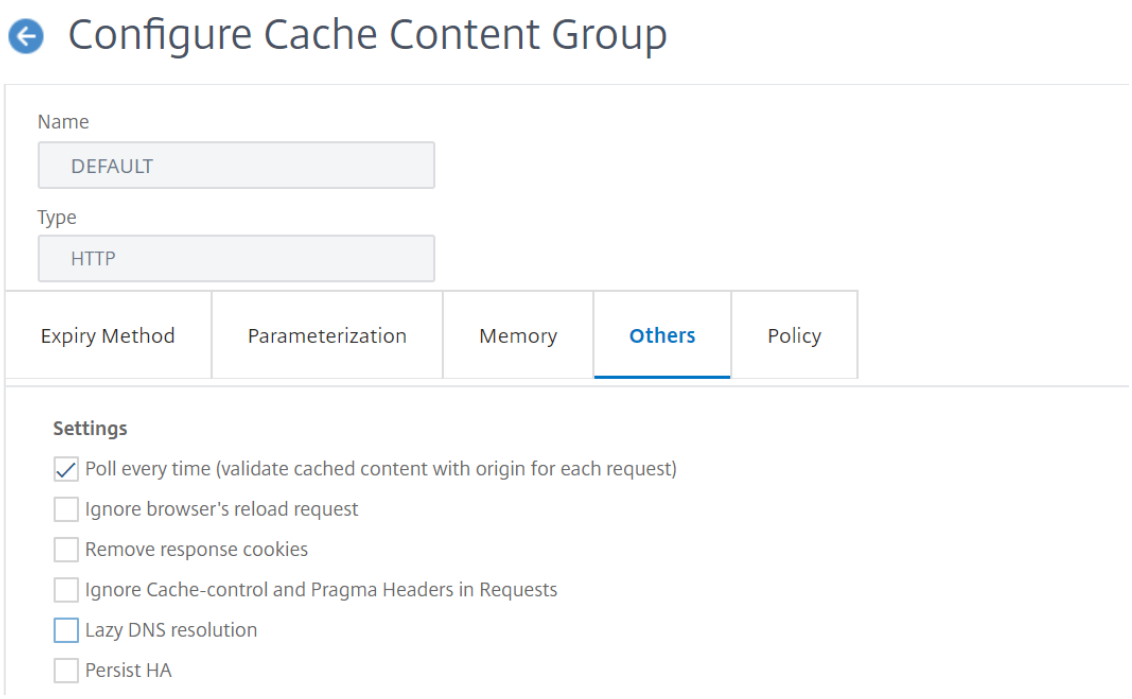
To configure poll every time by using the command line interface

At the command prompt, type:

```
add cache contentgroup <contentGroupName> -pollEveryTime YES
```

Poll by using the GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Content Groups**, and select the content group.
2. On the **Others** tab, in the Settings group, select Poll every time (validate cached content with origin for every request) option.



PET and client-specific content

The PET function can ensure that content is customized for a client. For example, a website that serves content in multiple languages examines the Accept-Language request header to select the language

for the content that it is serving. For a multi-language website where English is the predominant language, all English language content can be cached in a PET content group. This ensures that every request goes to the origin server to determine the language for the response. If the response is English, and the content has not changed, the origin server can serve a 304 Not Modified to the cache.

The following example shows commands to cache English responses in a PET content group, configure a named expression that identifies English responses in the cache, and configure a policy that uses this content group and named expression. Bold is used for emphasis:

```
add cache contentgroup EnglishLanguageGroup -pollEveryTime YES
add expression containsENExpression  - rule "http.res.header(\\"Content-
    Language\\").contains(\\"en\\")"
add cache policy englishPolicy -rule containsENExpression -action CACHE
     -storeInGroup englishLanguageGroup
bind cache policy englishPolicy -priority 100 -precedeDefRules NO
<!--NeedCopy-->
```

PET and authentication, authorization, and auditing

Outlook Web Access (OWA) is a good example of dynamically generated content that benefits from PET. All mail responses (*.EML objects) have an `ETag` validator that enables them to be stored as PET responses.

Every request for a mail response travels to the origin server, even if the response is cached. The origin server determines whether the requestor is authenticated and authorized. It also verifies that the response exists in the origin server. If all results are positive, the origin server sends a 304 Not Modified response.

Configure integrated cache as a forward proxy

September 14, 2021

The integrated cache can service as a forward proxy device that passes requests to other Citrix ADC appliances or to other types of cache servers. You configure the integrated cache as a forward proxy by identifying the IP addresses of the cache server or servers. After configuring the forward proxy, the Citrix ADC appliance sends requests that contain the configured IP address on to the cache server instead of involving the integrated cache.

To configure the Citrix ADC as a forward cache proxy by using the command line interface

At the command prompt, type:

```
add cache forwardProxy <IPAddress> <port>
```


To configure the Citrix ADC as a forward cache proxy by using the GUI

1. Navigate to **Optimization** > **Integrated Caching** > **Forward Proxy**, and add a forward proxy by specifying the IP address and port number.

Default settings for the integrated cache

September 14, 2021

The Citrix ADC integrated cache feature provides built-in policies with default settings and initial settings for the Default content group. The information in this section defines the parameters for the built-in policies and Default content group.

Default caching policies

The integrated cache has built-in policies. The Citrix ADC appliance evaluates the policies in a particular order, as discussed in the following sections.

You can override these built-in policies with a user-defined policy that is bound to a request-time override or response-time override policy bank.

> **Note**
> If you configured policies prior to release 9.0 and specified the -precedeDefRules parameter when binding the policies, they are automatically assigned to override-time bind points during migration.

View default policies

The built-in policy names start with an underscore (_). You can view the built-in policies from the command line and the administrative console using the show cache policy command.

Default request policies

You can override the following built-in request time policies by configuring new policies and binding them to the request-time override processing point. In the following policies, note that the MAY_NOCACHE action stipulates that the transaction is cached only when there is a user-configured or built-in CACHE directive at response time.

The following policies are bound to the _reqBuiltinDefaults policy label. They are listed in priority order.

Do not cache a response for a request that uses any method other than GET.

The policy name is _nonGetReq. The following is the policy rule:

```
!HTTP.REQ.METHOD.eq(GET)
```

Set a NOCACHE action for a request with header value that contains If-Match or If-Unmodified-Since.

The policy name is _advancedConditionalReq. The following is the policy rule:

```
HTTP.REQ.HEADER("If-Match").EXISTS || HTTP.REQ.HEADER("If-Unmodified-Since"
).EXISTS
```

Set a MAY_NOCACHE action for a request with the following header values: Cookie, Authorization, Proxy-authorization, or a request which contains the NTLM or Negotiate header.

The policy name is _personalizedReq. The following is the policy rule:

```
HTTP.REQ.HEADER("Cookie").EXISTS || HTTP.REQ.HEADER("Authorization").EXISTS
 || HTTP.REQ.HEADER("Proxy-Authorization").EXISTS || HTTP.REQ.IS_NTLM_OR_NEGOTIATE
```

Default response policies

You can override the following default response-time policies by configuring new policies and binding them to the response-time override processing point.

The following policies are bound to the **_resBuiltinDefaults** policy label and are evaluated in the order in which they are listed:

1. Do not cache HTTP responses unless they are of type 200, 304, 307, 203 or if the types are between 400 and 499 or between 300 and 302.

   The policy name is **_uncacheableStatusRes**. The following is the policy rule:

   ```
   !((HTTP.RES.STATUS.EQ(200))|| (HTTP.RES.STATUS.EQ(304))|| (HTTP.RES.
   STATUS.BETWEEN(400,499))|| (HTTP.RES.STATUS.BETWEEN(300, 302))|| (HTTP.
   RES.STATUS.EQ(307))|| (HTTP.RES.STATUS.EQ(203)))
   ```

2. Do not cache an HTTP response if it has a Vary header with a value of anything other than Accept-Encoding.

   The compression module inserts the Vary: Accept_Encoding header. The name of this expression is **_uncacheableVaryRes**. The following is the policy rule:

   ```
   ((HTTP.RES.HEADER("Vary").EXISTS)&& ((HTTP.RES.HEADER("Vary").INSTANCE
   (1).LENGTH > 0)|| (!HTTP.RES.HEADER("Vary").STRIP_END\\_WS.SET_TEXT_MODE
   (IGNORECASE).eq("Accept-Encoding"))))
   ```

3. Do not cache a response if its Cache-Control header value is No-Cache, No-Store, or Private, or if the Cache-Control header is not valid.

The policy name is **_uncacheableCacheControlRes**. The following is the policy rule:

```
((HTTP.RES.CACHE\\_CONTROL.IS\\_PRIVATE)|| (HTTP.RES.CACHE\\_CONTROL.IS
\\_NO\\_CACHE)|| (HTTP.RES.CACHE\\_CONTROL.IS\\_NO\\_STORE)|| (HTTP.RES
.CACHE\\_CONTROL.IS\\_INVALID))
```

4. Cache responses if the Cache-Control header has one of the following values: Public, Must-Revalidate, Proxy-Revalidate, Max-Age, S-Maxage.

   The policy name is **_cacheableCacheControlRes**. The following is the policy rule:

   ```
   ((HTTP.RES.CACHE_CONTROL.IS_PUBLIC)|| (HTTP.RES.CACHE_CONTROL.IS_MAX_AGE
   )|| (HTTP.RES.CACHE_CONTROL.IS_MUST_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL
   .IS_PROXY_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL.IS_S_MAXAGE))
   ```

5. Do not cache responses that contain a Pragma header.

   The name of the policy is **_uncacheablePragmaRes**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Pragma").EXISTS
   ```

6. Cache responses that contain an Expires header.

   The name of the policy is **_cacheableExpiryRes**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Expires").EXISTS
   ```

7. If the response contains a Content-Type header with a value of Image, remove any cookies in the header and cache it.

   The name of the policy is **_imageRes**. The following is the policy rule:

   ```
   HTTP.RES.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).STARTSWITH("
   image/")
   ```

   You can configure the following content group to work with this policy:

   ```
   add cache contentgroup nocookie -group -removeCookies YES
   ```

8. Do not cache a response that contains a Set-Cookie header.

   The name of the policy is **_ personalizedRes**. The following is the policy rule:

HTTP.RES.HEADER(“Set-Cookie”).EXISTS	HTTP.RES.HEADER(“Set-Cookie2”).EXISTS

Restrictions on default policies

You cannot override the following built-in request time policies with user-defined policies.

These policies are listed in priority order.

1. Do not cache any responses if the corresponding HTTP request lacks a GET or POST method.
2. Do not cache any responses for a request if the HTTP request URL length plus host name exceeds 1744 byes.
3. Do not cache a response for a request that contains an If-Match header.
4. Do not cache a request that contains an If-Unmodified-Since header.

> **Note**
> This is different from the If-Modified-Since header.

1. Do not cache a response if the server does not set an expiry header.

You cannot override the following built-in response time policies. These policies are evaluated in the order in which they are listed:

1. Do not cache responses that have an HTTP response status code of 201, 202, 204, 205, or 206.
2. Do not cache responses that have an HTTP response status code of 4xx, with the exceptions of status codes 403, 404, and 410.
3. Do not cache responses if the response type is FIN terminated, or the response does not have one of the following attributes: Content-Length, or Transfer-Encoding: Chunked.
4. Do not cache the response if the caching module cannot parse its Cache-Control header.

Initial settings for the default content group

When you first enable integrated caching, the Citrix ADC appliance provides one predefined content group named the Default content group. For detailed information, see Default content group settings table.

Troubleshooting

September 14, 2021

If the integrated cache feature does not work as expected after you have configured it, you can use some common tools to access Citrix ADC resources and diagnose the problem.

Resources for troubleshooting

For more information about the resources available for troubleshooting and sample configurations, see Resource for troubleshooting PDF file.

Front end optimization

September 14, 2021

> **Note:** Front end optimization is available if you have an Advanced or Premium Citrix ADC license and are running Citrix ADC release 10.5 or later.

The HTTP protocols that underlie web applications were originally developed to support the transmission and rendering of simple webpages. New technologies such as JavaScript and cascading style sheets (CSS), and new media types such as Flash videos and graphics-rich images, place heavy demands on front-end performance, that is, on performance at the browser level.

The Citrix ADC front end optimization (FEO) feature addresses such issues and reduces the load time and render time of webpages by:

- Reducing the number of requests.
- Required for rendering each page.
- Reducing the number of bytes in page responses.

Simplifying and optimizing the content served to the client browser.

You can customize your FEO configuration to provide the best results for your users. Citrix ADCs support numerous web content optimizations for both desktop and mobile users. The following tables describe the front-end optimizations provided by the FEO feature, and the operations performed on different types of files.

Optimizations performed by the FEO feature

Web Optimization	Problem	What Citrix ADC FEO feature does	Benefits
Inlining	Client browsers often send multiple requests to servers for loading external CSS, images, and JavaScript associated with the webpage.	CSS inline, JavaScript inline, CSS combine	Loading the external CSS, images, and JavaScript inline with the HTML files improves page-rendering time. This optimization is beneficial for content that is viewed only once, and for mobile devices that have limited cache sizes.
Minification	Data fetched from servers includes inessential characters such as white spaces, comments, and newline characters. The time that browsers spend in processing such data creates website latency.	CSS minification, JavaScript minification, Removal of HTML comments	Minified files consume less bandwidth and avoid the latency caused by special processing.
Image optimization	Mobile browsers often have slow connection speeds and limited cache memory. Downloading the images on mobile clients consumes more bandwidth, processing time, and cache space, resulting in website latency.	JPEG optimization, CSS image inlining, **Image shrink-to** attributes, GIF to PNG conversion, HTML image inlining, WebP image conversion, JPEG, GIF, PNG to JPEG-XR image conversion	Reduces the image to the size indicated in the image tag by Citrix ADC, enabling client browsers to load images faster.

Web Optimization	Problem	What Citrix ADC FEO feature does	Benefits
Repositioning	Inefficient processing of external CSS, images, and JavaScript increases page-load time.	Image lazy loading, CSS move to Head, JavaScript move to end	Repositions HTML elements, to reduce rendering time for webpages and enable client browsers to load the objects faster.
Connection Management	Many browsers set limits on the number of simultaneous connections that can be established to a single domain. This can cause browsers to download webpage resources one at a time, resulting in higher browsers time.	Domain sharding	Overcomes the connection limitation, which improves page-rendering time by enabling client browsers to download more resources in parallel.

Web Optimizations on different file types:

Citrix ADC can perform web optimizations on CSS, images, Javascript, and HTML. For more information, see Web Optimizations PDF.

> **Note:**
> The front end optimization feature supports ASCII characters only. It does not support the Unicode character set.

How front end optimization works

After the Citrix ADC receives the response from the server:

1. Parses the contents of the page, creates an entry in the cache (wherever applicable), and applies the FEO policy.

   For example, a Citrix ADC can apply the following optimization rules:

   - Remove white spaces or comments present within a CSS or JavaScript.
   - Combine one or more CSS files to one file.

- Convert GIF image format to PNG format.

2. Rewrites the embedded objects and saves the optimized content in the cache, with a different signature than the one used for the initial cache entry.
3. For subsequent requests, fetches the optimized objects from the cache, not from the server, and forwards the responses to the client.

**

Remove extraneous information such as white spaces and comments.

The period during which the browser can use the cached resource without checking to see if fresh content is available on the server.

Configure front end optimization

Optionally, you can change the values of the front end optimization global settings. Otherwise, begin by creating actions that specify the optimization rules to be applied to the embedded objects.

After configuring actions, create policies, each with a rule specifying a type of request for which to optimize the response, and associate the actions with the policies.

> **Note:** The Citrix ADC evaluates front end optimization policies at request time only, not at response time.

To put the policies into effect, bind them to bind points. You can bind a policy globally, so that it applies to all traffic that flows through the Citrix ADC, or you can bind the policy to a load balancing or content switching virtual server of type HTTP or SSL. When you bind a policy, assign it a priority. A lower priority number indicates a higher value. The Citrix ADC applies the policies in the order of their priorities.

Prerequisites

Front end optimization requires the Citrix ADC integrated caching feature to be enabled. Also, you must perform the following integrated caching configurations:

- Allocate cache memory.
- Set the maximum response size and memory limit for a default cache content group.

For more information on configuring integrated caching, see Integrated Caching.

Note: The term Integrated Cache can be interchangeably used with AppCache; note that from a functionality point of view, both terms mean the same.

Configure front end optimization by using Citrix ADC command interface

At the command prompt, do the following:

1. Enable the front end optimization feature.

```
enable ns feature FEO
```

1. Create one or more front end optimization actions.

```
add feo action <name> [-imgShrinkToAttrib] [-imgGifToPng] ...
```

Example: To add a front end optimization action for converting images in GIF format to PNG format and to extend the cache expiry period:

```
add feo action allact -imgGifToPng -pageExtendCache
```

1. [Optional] Specify non-default values for front end optimization global settings.

```
set feo parameter [-cacheMaxage <integer>] [-JpegQualityPercent <integer>]
[-cssInlineThresSize <integer>] [-inlineJsThresSize <integer> [-inlineImgThresSize
 <integer>]
```

Example: To specify the cache maximum expiry period:

```
set feo parameter -cacheMaxage 10
```

1. Create one or more front end optimization policies.

```
add feo policy <name> <rule> <action>
```

Example: To add a front end optimization policy and associate it with the above specified allact action:

```
1 >add feo policy pol1 TRUE all act
2 >add feo policy pol1 "(HTTP.REQ.URL.CONTAINS(\"testsite\"))" allact1
3 <!--NeedCopy-->
```

1. Bind the policy to a load balancing or content switching virtual server, or bind it globally.

```
bind lb vserver <name> -policyName <string> -priority <num>
```

```
bind cs vserver <name> -policyName <string> -priority <num>
```

```
bind feo global <policyName> <priority> -type <type> <gotoPriorityExpression
>
```

Example: To apply the front end optimization policy to a virtual server named “abc”:

```
> bind lb vserver abc -policyName pol1 -priority 1 -type NONE
```

Example: To apply the front end optimization policy for all the traffic reaching the ADC:

```
> bind feo global pol1 100 -type REQ_DEFAULT
```

1. Save the configuration. save ns config

Configure front end optimization by using the GUI

1. Navigate to **Optimization** > **Front End Optimization** > **Actions**, and click **Add** and create a front end optimization action by specifying the relevant details.
2. [Optional] Specify the front end optimization global settings.
3. Navigate to **Optimization** > **Front End Optimization**, and on the right-pane, under Settings, click **Change Front End Optimization** settings and specify the front end optimization global settings.
4. Create a front end optimization policy.
5. Navigate to **Optimization** > **Front End Optimization** > **Policies**, click **Add** and create a front end optimization policy by specifying the relevant details.
6. Bind the policy to a load balancing or content switching virtual server.
   a) Navigate to **Optimization** > **Front End optimization** > **Policies**.
   b) Select a front end optimization policy and click **Policy Manager**.
   c) Under **Front End Optimization Policy Manager**, bind the front end optimization policy to a load balancing or content switching virtual server.

Verify front end optimization configuration

The dashboard utility displays summary and detailed statistics in tabular and graphic formats. You can view the FEO statistics to evaluate your FEO configuration.

Optionally, you can also display statistics for an FEO policy, including the number of select that the policy counter increments during the policy based FEO.

> **Note:**
> For more information about statistics and charts, see the Dashboard help on the Citrix ADC appliance.

View FEO statistics by using the CLI

At the command prompt, type the following commands to display a summary of FEO statistics, FEO policy select and details, and detailed FEO statistics, respectively:

- `stat feo` Note: The stat feo policy command displays statistics only for advanced FEO policies.
- `show feo policy name`
- `stat feo -detail`

View FEO statistics on Citrix ADC dashboard

In the dashboard GUI, you can:

- Select Front End Optimization to display a summary of `FEO` statistics.

- Click the **Graphical View** tab to display the rate of requests processed by the FEO feature.

Sample optimization:

Refer to the Sample PDF for some examples of content optimization actions that are applied on HTML content and the embedded objects within the HTML content.

Content accelerator

September 14, 2021

> **Important:**
>
> The content accelerator feature is no longer supported on the Citrix ADC appliance.

Content accelerator is a Citrix ADC feature that you can use in a Citrix ByteMobile T1100 deployment, to store data on a Citrix ByteMobile T2100 appliance.

Storing data on a T2100 appliance saves bandwidth and provides faster response times, because the Citrix ADC does not have to connect to the server for repeated requests of the same data.

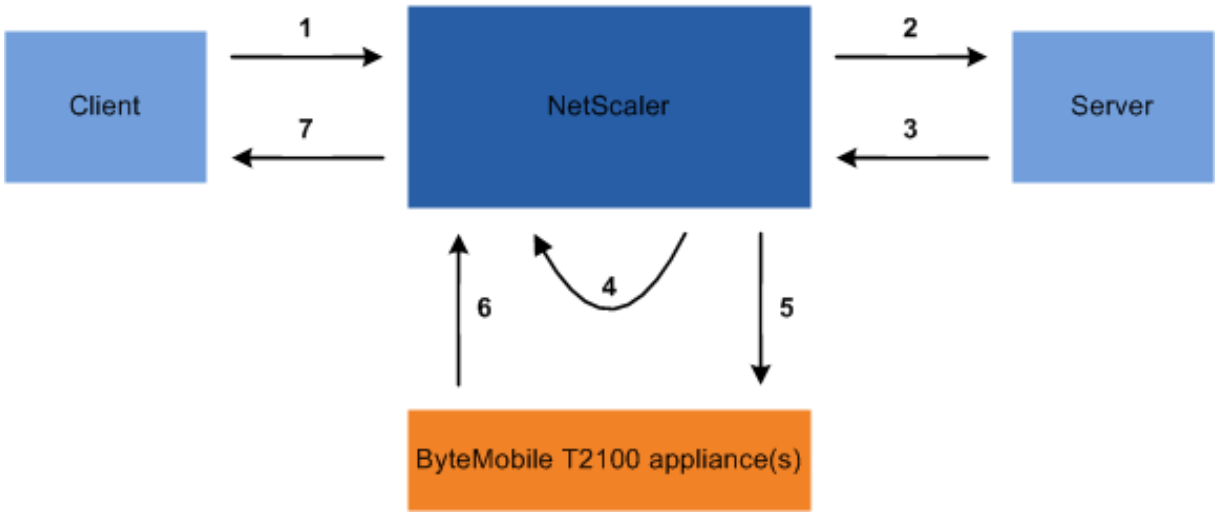
> **Note:** Content accelerator works with a Citrix ByteMobile Premium license. Contact customer support for more information and for obtaining the license.

How the content accelerator works

When a load balancing or content switching virtual server receives a client request, the Citrix ADC appliance evaluates a content accelerator policy that you have bound to the virtual server. The policy filters the requests to identify the ones to which to apply the content accelerator feature.

> **Note:**
>
> For HTTP requests, the content accelerator feature can serve partial content in response to single byte-range requests.

The following figure illustrates the operations that the appliance performs when a client request arrives at a virtual server configured to use the content accelerator feature:



The process flow is as follows:

1. Client sends request.
2. Citrix ADC forwards the request to the server.
3. Server responds with the predefined size of the response (specified by the accumResSize parameter of the add ca action command).
4. Citrix ADC computes a hash of the response sent by the server.
5. Citrix ADC looks up the hash on the T2100 appliance.
6. A successful lookup indicates that the data is available and the T2100 appliance sends the data to the Citrix ADC.

   > **Note:**
   > When the database lookup does not succeed, the appliance fetches the requested data from the server, and serves the data to the client and updates the data on the T2100 appliance.
   > The T2100 appliance can be configured to specify the number of requests for which to cache data.

7. Citrix ADC sends the response to the client.

Configure content accelerator

Before you configure the content accelerator feature, you must enable it on the Citrix ADC appliance.

You must configure the content accelerator feature to use one or more T2100 appliances. You must add each T2100 appliance as a service and bind these services to a load balancing virtual server that is dedicated for distributing the load between the configured T2100 appliances.

You must configure a content accelerator action to look up the data on the T2100 appliance. The action must specify the T2100 load balancing virtual server and the size of data (in KB) to be fetched from the server to calculate the hash.

The action must be bound to a content accelerator policy that defines the traffic on which to perform content acceleration. The content accelerator policy must be bound to a content switching or load balancing virtual server that receives client traffic. Alternatively, you can bind the policy globally to all applicable virtual servers.

To configure the content accelerator by using the command line interface

At the command prompt, do the following:

1. Enable the content accelerator feature.

   ```
   enable ns feature ca
   ```

2. Identify the T2100 appliances and add each as a service on the Citrix ADC appliance.

   ```
   add service <name> <IPAddress> <serviceType> <port>
   ```

   Example:

   ```
   > add service T2100-A 10.102.29.61 HTTP 30
   > add service T2100-B 10.102.29.62 HTTP 40
   > add service T2100-C 10.102.29.63 HTTP 50
   <!--NeedCopy-->
   ```

   > **Note:**
   > The services must be of type HTTP only.

3. Create a load balancing virtual server for the T2100 appliances. Specify the token load balancing method and the rule shown in the following syntax.

   ```
   add lb vserver <name> <serviceType> <IPAddress> <port> -lbMethod
      TOKEN -rule "http.req.url.after_str(\"/lookup/\") alt http.req.
      url.path.SKIP(1).PREFIX(64)"
   <!--NeedCopy-->
   ```

   Example:

   ```
   add lb vserver T2100-lbvserver HTTP 10.102.29.64 99 -lbMethod
      TOKEN -rule "http.req.url.after_str(\"/lookup/\") alt http.req.
      url.path.SKIP(1).PREFIX(64)"
   <!--NeedCopy-->
   ```

4. Bind the T2100 services to the load balancing virtual server that you created for them.

   ```
   bind lb vserver <name> <serviceName>
   ```


Example:

```
> bind lb vserver T2100-lbvserver T2100-A
> bind lb vserver T2100-lbvserver T2100-B
> bind lb vserver T2100-lbvserver T2100-C
<!--NeedCopy-->
```

5. Define a content accelerator action.

   ```
   add ca action <name> accumResSize <KBytes> -lbvserver <string> -type lookup
   ```

   Example:

   ```
   > add ca action ca_action1 -type lookup -lbvserver T2100-lbvserver -accumResSize 60
   ```

6. Define a content accelerator policy.

   ```
   add ca policy <name> -rule <expression> -action <name>
   ```

   Example:

   To create a content accelerator policy that caches all video formats.

   ```
   > add ca policy ca_mp4_pol -rule ns_video -action ca_action1
   ```

   where ns_video is a built-in expression.

7. Bind the content accelerator policy to either a virtual server that receives traffic or globally to the Citrix ADC system.

   ```
   bind lb vserver <name> -policyName <string>
   ```

   ```
   bind cs vserver <name> -policyName <string>
   ```

   ```
   bind ca global -policyName <string> -priority <num> -type <type>
   ```

   Example: To apply the content accelerator policy to a virtual server named “traf_rec”

   ```
   bind lb vserver traf_rec -policyName ca_mp4_pol
   ```

   Example: To apply the content accelerator policy for all traffic reaching the Citrix ADC.

   ```
   bind ca global -policyName ca_mp4_pol -priority 100 -type RES_DEFAULT
   ```

8. Save the configuration.

   ```
   save ns config
   ```

Configuring content accelerator by using the GUI

1. Navigate to **System** > **Settings** > **Configure Advanced Features** and select **Content Accelerator**.

2. Create a service for each of the T2100 appliances.
   a) Navigate to **Traffic Management** > **Load Balancing** > **Services**.
   b) Click **Add** and specify the relevant details. In the **Server** field, make sure you specify the IP address of the T2100 appliance. In the **Protocol** field select HTTP.
3. Create a virtual server and bind the T2100 services to it.
   a) Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
   b) Click **Add** and specify the relevant details.
   c) In the **Method and Persistence** tab, specify the Method as **Token**.
   d) In the **Policies** tab, specify the rule as http.req.url.after_str("/lookup/") alt http.req.url.path.SKIP(1).PRE
   e) In the **Services** tab, select the T2100 services that you want to bind to the virtual server.
4. Create a content accelerator action.
   a) Navigate to **Optimization** > **Content Accelerator** > **Actions**.
   b) Specify the relevant details.
5. Create a content accelerator policy.
   a) Navigate to **Optimization** > **Content Accelerator** > **Policies**.
   b) Click **Add**, specify the policy rule, and associate the content accelerator action.
6. Bind the content accelerator policy globally or to a virtual server.
   a) Navigate to **Optimization** > **Content Accelerator**.
   b) Under the **Content Accelerator Policy Manager** [REQUEST] or Content Accelerator Policy Manager [RESPONSE] sections, bind the content accelerator policy globally or to a virtual server.

Media classification

September 14, 2021

Understanding the type of traffic in the network helps network administrators to manage bandwidth consumption for optimal network performance. The media classification mode monitors and displays the statistics of media traffic going through the Citrix ADC appliance.

With this mode enabled, a network administrator can collect stats showing the amount of data accessed, and the types of devices from which the media files have been accessed. The Citrix ADC appliance also supports byte-range requests in this mode.

Currently the Citrix ADC appliance can monitor and display statistics for the following media file types:

Media	File type
Microsoft Smooth Streaming	Video
Apple Live Streaming	Video

Media	File type
Audio Data Transport Stream (ADTS)	Audio
Advanced Audio Coding (AAC)	Audio
Flash Video (FLV)	Audio and Video
3GP	Audio and Video

The appliance can display stats for the following devices:

Device Platform	Device Type
iOS	iPad and iPod
Android	Mobiles and tablets
Laptop or Desktop	Windows laptop and desktop computers
Others	Other mobile devices (mobiles and tablets)

The network administrators can check the following stats counters to know the amount of data accessed through the Citrix ADC appliance for various media traffic types.

Media File Name	Stats Counter
Microsoft Smooth Streaming	`mcmssmthstrmvid`—This counter records the total number of Microsoft Smooth Streaming videos served by the Citrix ADC appliance; `Mcmssmthstrvidpl`—This counter records the total number of Microsoft Smooth Streaming video playlists served by the Citrix ADC appliance; `Mcmssmthstrmvidbytes`—This counter records the total number of data bytes served for Microsoft Smooth Streaming media traffic on the Citrix ADC appliance; `Mcmssmthstrmplvidbytespl`—This counter records the total number of Microsoft Smooth Streaming playlist bytes served by the Citrix ADC appliance.

Media File Name	Stats Counter
Apple Live Streaming	mccaplelivestrmngvid—This counter records the total number of Apple Live Streaming videos served by the Citrix ADC appliance. Mccapplelivestrmngvidpl—This counter records the total number of Apple Live Streaming video playlists served by the Citrix ADC appliance. Mcapplelivestreamingvidbytes—This counter records the total number of data bytes served for Apple Live Streaming media traffic on the Citrix ADC appliance. Mcapplelivestreamingplaylistvidbytespl—This counter records the total number of Apple Live Playlist bytes served by the Citrix ADC appliance.
Audio Data Transport Stream (ADTS)	mcadtsaudio—This counter records the total number of ADTS audio clips served by the Citrix ADC appliance. Mcadtsaudiobytes—This counter records the total number of data bytes served for ADTS media traffic on the Citrix ADC appliance.
Advanced Audio Coding (AAC)	Mcaacaudio—This counter records the total number of AAC audio clips served by the Citrix ADC appliance. Mcaacaudiobytes—This counter records the total number of data bytes served for AAC media traffic on the Citrix ADC appliance.
Flash Video (FLV)	Mcflvvid—This counter records the total number of flash videos served by the Citrix ADC appliance. Mcflvvidbytes—This counter records the total number of data bytes served for flash videos on the Citrix ADC appliance.
3GP	mc3gpvidbytes—This counter records the total number of data bytes served for 3GP media traffic on the Citrix ADC appliance.

The Citrix ADC appliance detects media file types by their signatures in the *initial body bytes* of the responses. For example, the initial body bytes for an mp4 file have the following signature in the response:

```
**....ftypmp42** ....isommp42....moov...lmvhd.....c.\!.c.\!..
```

The Citrix ADC appliance detects the client device type by the *user agent string* that the client device includes in the HTTP GET request. For example, a window phone using a UC browser has the following user agent string in the HTTP GET request:

```
User-Agent: **UCWEB**/2.0 (**Windows**; U; wds 8.10; en-US; HTC; 8X by HTC)
U2/1.0.0
```

Enable media classification

By default, media classification is disabled on the Citrix ADC appliance. You have to enable the mode before using it.

To enable media classification by using the command line interface

At the command prompt, type:

```
enable ns mode Mediaclassification
```

To enable media classification by using the GUI

Enable media classification on Citrix ADC appliance

Navigate to **System** > **Settings** > **Configure Modes** and select **Media Classification**.

To view media traffic statistics on the Citrix ADC appliance

Navigate to **Optimization** and click **Media Classification** to view the media traffic statistics.

Verify media classification statistics

You can view the media traffic statistics in the dashboard utility or using the command line interface. The dashboard utility displays summary and detailed statistics in a tabular and graphic format.

> **Note**
> For more information about statistics and charts, see the Dashboard help on your Citrix ADC appliance.

To View media classification statistics by using the command line interface

At the command prompt, type one of the following commands to display a summary of media classification statistics, display detailed statistics, or clear the display:

```
stat Mediaclassification
```



```
stat Mediaclassification -detail

stat Mediaclassification -clearstats
```

To view Media Classification statistics on the Dashboard

In the **Dashboard** utility, you can display the following types of media classification statistics:

1. Select **Media Classification** to display a summary of the media traffic statistics.
2. To display detailed media traffic statistics, click the **Details**.
3. To clear the media traffic statistics, click **Clear**.

Reputation

September 14, 2021

Citrix offers reputation based security. Using reputation assessment to determine the risk of processing requests, you can take actions such as blocking or dropping certain requests to improve the performance of your application.

The Citrix ADC IP reputation feature uses IP reputation checks to prevent Zero day attacks and provide protection against malicious sources associated with web attacks, phishing activity, or web scanning.

For more details, see IP Reputation.

IP Reputation

October 25, 2021

IP reputation is a tool that identifies IP addresses that send unwanted requests. Using the IP reputation list you can reject requests that are coming from an IP address with a bad reputation. Optimize Web Application Firewall performance by filtering requests that you do not want to process. Reset, drop a request, or even configure a responder policy to take a specific responder action.

Following are some attacks that you can prevent by using IP Reputation:

- **Virus Infected personal computers**. (home PCs) are the single biggest source of Spam on the internet. IP Reputation can identify the IP address that is sending unwanted requests. IP reputation can be especially useful for blocking large scale DDoS, DoS, or anomalous SYN flood attacks from known infected sources.
- **Centrally managed and automated botnet**. Attackers have gained popularity for stealing passwords, because it doesn't take long when hundreds of computers work together to crack

your password. It is easy to launch botnet attacks to figure out passwords that use commonly used dictionary words.

- **Compromised web-server**. Attacks are not as common because awareness and server security have increased, so hackers and spammers look for easier targets. There are still web servers and online forms that hackers can compromise and use to send spam (such as viruses and porn). Such activity is easier to detect and quickly shut down, or block with a reputation list such as SpamRats.
- **Windows Exploits**. (such as Active IPs offering or distributing malware, shell code, rootkits, worms, or viruses).
- **Known spammers and hackers**.
- **Mass e-mail marketing campaigns**.
- **Phishing Proxies** (IP addresses hosting phishing sites, and other fraud such as ad click fraud or gaming fraud).
- **Anonymous proxies** (IPs providing proxy and anonymization services including The Onion Router aka TOR).

A Citrix ADC appliance uses **Webroot** as the service provider for a dynamically generated malicious IP database and the metadata for those IP addresses. Metadata might include geolocation details, threat category, threat count, and so on. The Webroot threat Intelligence engine receives real-time data from millions of sensors. It automatically and continuously captures, scans, analyses and scores the data, using advanced machine learning and behavioral analysis. Intelligence about a threat is continually updated.

The Citrix ADC appliance validates an incoming request for its bad reputation using the Webroot's uses IP reputation database. The database has a huge collection of IP address classified based IP threat categories. Following are the IP threat categories and its description.

- Spam Sources. Spam Sources includes Tunneling Spam messages through proxy, anomalous SMTP activities, Forum Spam activities.
- Windows Exploits. Windows exploit category includes active IP Address offering or distributing malware, shell code, rootkits, worms or viruses
- Web Attacks. Web attacks category includes cross site scripting, iFrame injection, SQL injection, cross domain injection, or domain password brute force attack
- Botnets. Botnet category includes Botnet C&C channels, and infected zombie machine controlled by Bot master
- Scanners. Scanners category includes all reconnaissance such as probes, host scan, domain scan and password brute force attack
- Denial of Service. Denial of Services category includes DOS, DDOS, anomalous sync flood, anomalous traffic detection
- Reputation. Deny access from IP addresses currently known to be infected with malware. This category also includes IPs with average low Webroot Reputation Index score. Enabling this category prevents access from sources identified to contact malware distribution points

- Phishing. Phishing category includes IP addresses hosting phishing sites, other kind of fraud activities such as Ad Click Fraud or Gaming fraud
- Proxy. Proxy category includes IP addresses providing proxy and def services.
- Mobile Threats. Mobile Threat category includes IP addresses of malicious and unwanted mobile applications. This category leverages data from the Webroot mobile threat research team.
- Tor Proxy. Tor proxy category includes IP addresses acting as exit nodes for the Tor Network. Exit nodes are the last point along the proxy chain and make a direct connection to the originator's intended destination.

When a threat is detected anywhere in the network, the IP address is flagged as malicious and all appliances connected to the network are immediately protected. The dynamic changes in the IP addresses are processed with high speed and accuracy by using advanced machine learning.

As stated in the data sheet from Webroot, the Webroot's sensor network identifies many key IP threat types, including spam sources, Windows exploits, botnets, scanners, and others. (See the flow diagram on the data sheet.)

The Citrix ADC appliance uses an `iprep` client process to get the database from Webroot. The `iprep` client uses the HTTP GET method to get the absolute IP list from Webroot for the first time. Later, it checks delta changes once every 5 minutes.

> **Important:**
>
> - Make sure that the Citrix ADC appliance has Internet access and DNS is configured before you use the IP Reputation feature.
> - To access the Webroot database, the Citrix ADC appliance must be able to connect to **api.bcti.brightcloud.com** on **port 443**. Each node in the HA or cluster deployment gets the database from Webroot and must be able to access this Fully Qualified Domain Name (FQDN).
> - Webroot hosts its reputation database in AWS currently. Therefore, Citrix ADC must be able to resolve AWS domains for downloading the reputation db. Also, the firewall must be open for AWS domains.
>
> **Note:**
>
> Each packet engine requires at least 4 GB to function properly when the IP Reputation feature is enabled.

Advanced policy Expressions. Configure the IP Reputation feature by using advanced policy expressions (default syntax expressions) in the policies bound to supported modules, such as Web Application Firewall and responder. Following are two examples showing expressions that can be used to detect whether the client IP address is malicious.

1. **CLIENT.IP.SRC.IPREP_IS_MALICIOUS**: This expression evaluates to TRUE if the client is included in the malicious IP list.

2. **CLIENT.IP.SRC.IPREP_THREAT_CATEGORY (CATEGORY)**: This expression evaluates to TRUE if the client IP is malicious IP and is in the specified threat category.

Following are the possible values for the threat category:

SPAM_SOURCES, WINDOWS_EXPLOITS, WEB_ATTACKS, BOTNETS, SCANNERS, DOS, REPUTATION, PHISHING, PROXY, NETWORK, CLOUD_PROVIDERS, MOBILE_THREATS, TOR_PROXY.

> **Note:**
>
> The IP reputation feature checks both source and destination IP addresses. It detects malicious IPs in the header. If the PI Expression in a policy can identify the IP address, the IP reputation check determines whether it is malicious.

IPRep log message. The `/var/log/iprep.log` file contains useful messages that capture information about communication with the Webroot database. The information can be about the credentials used during Webroot communication, failure to connect with Webroot, information included in an update (such as the number of IP addresses in the database).

Creating a blocklist or allowlist of IPs using a policy data set. You can maintain an allow list to allow access to specific IP addresses that are blocklisted in the Webroot database. You can also create a customized block list of IP addresses to supplement the Webroot reputation check. These lists can be created by using a policy **data set**. A data set is a specialized form of pattern set that is ideally suited for IPv4 address matching. To use data sets, first create the data set and bind IPv4 addresses to it. When configuring a policy for comparing a string in a packet, use an appropriate operator and pass the name of the pattern set or data set as an argument.

To create an allow list of addresses to treat as exceptions during IP reputation evaluation:

- Configure the policy so that the PI expression evaluates to False even if an address in the allow list is listed as malicious by Webroot (or any service provider).

Enabling or disabling IP reputation. IP reputation is a part of the general reputation feature, which is license based. When you enable or disable the reputation feature, it enables or disables IP Reputation.

General procedure. Deploying IP reputation involves the following tasks

- Verify that the license installed on the Citrix ADC appliance has IP reputation support. Premium and standalone application firewall licenses support the IP reputation feature.
- Enable the IP reputation and application firewall features.
- Add an application firewall profile.
- Add an application firewall policy using the PI expressions to identify the malicious IP addresses in the IP Reputation database.
- Bind the application firewall policy to an appropriate bind point.
- Verify that any request received from a malicious address gets logged in the `ns.log` file to show that the request was processed as specified in the profile.

Configure the IP reputation feature using the CLI

At the command prompt, type:

- `enable feature reputation`
- `disable feature reputation`

The following examples show how you can add an application firewall policy using the PI expression to identify malicious addresses. You can use the built-in profiles, or add a profile, or configure an existing profile to invoke the desired action when a request matches a policy match.

Examples 3 and 4 show how to create a policy dataset to generate a block list or an allow list of IP addresses.

Example 1:

The following command creates a policy that identifies malicious IP addresses and block the request if a match is triggered:

```
add appfw policy pol1 CLIENT.IP.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
```

Example 2:

The following command creates a policy that uses the reputation service to check the client IP address in the `X-Forwarded-For` header and reset the connection if a match is triggered.

```
> add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").TYPECAST_IP_ADDRESS_AT
.IPREP_IS_MALICIOUS"APPFW_RESET**
```

Example 3:

The following example shows how to add a list to add exceptions that allow specified IP addresses:

```
> add policy dataset Allow_list1 ipv4
> bind policy dataset Allow_list1 10.217.25.17 -index 1
> bind policy dataset Allow_list1 10.217.25.18 -index 2
```

Example 4:

The following example shows how to add the customized list to flag specified IP addresses as malicious:

```
> add policy dataset Block_list1 ipv4
> bind policy dataset Block_list1 10.217.31.48 -index 1
> bind policy dataset Block_list1 10.217.25.19 -index 2
```

Example 5:

The following example shows a policy expression to block the client IP in the following conditions:

- It matches an IP address configured in the customized Block_list1 (example 4)
- It matches an IP address listed in the Webroot database unless relaxed by inclusion in the Allow_list1 (example 3).

```
1 > add appfw policy "Ip_Rep_Policy" "((CLIENT.IP.SRC.IPREP_IS_MALICIOUS
     || CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(\"Block_list1\")) && !
      (CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(\"Allow_list1\")))"
     APPFW_BLOCK
2 <!--NeedCopy-->
```

Using Proxy server:

If the Citrix ADC appliance does not have direct access to the internet and is connected to a proxy, configure the IP Reputation client to send requests to the proxy.

At the command prompt, type:

```
set reputation settings -proxyServer <proxy server ip> -proxyPort <proxy server port>
```

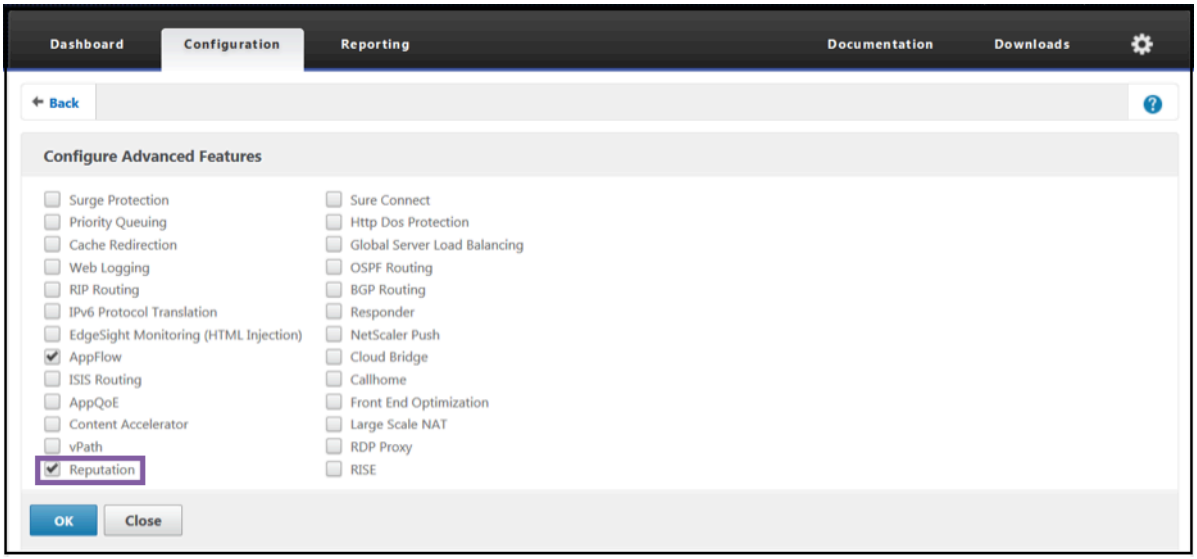
Example:

```
> set reputation settings proxyServer 10.102.30.112 proxyPort 3128
> set reputation settings -proxyServer testproxy.citrite.net -proxyPort 3128
> unset reputation settings -proxyserver -proxyport
> sh reputation settings
```

> **Note:**
>
> The Proxy Server IP can be an IP address or a fully qualified domain name (FQDN).

Configure IP reputation by using Citrix ADC GUI

1. Navigate to the **System > Settings**. In the **Modes and Features** section, click the link to access the **Configure Advanced Features** pane and enable the **Reputation** check box.
2. Click **OK**.

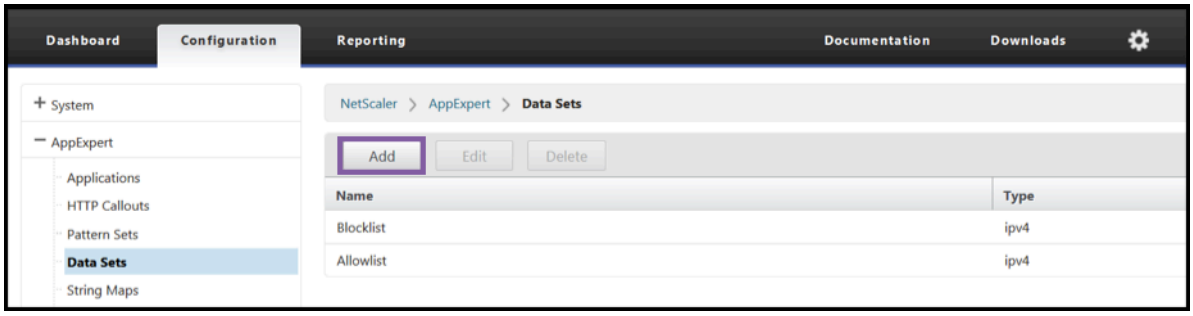


To configure a proxy server by using the Citrix ADC GUI

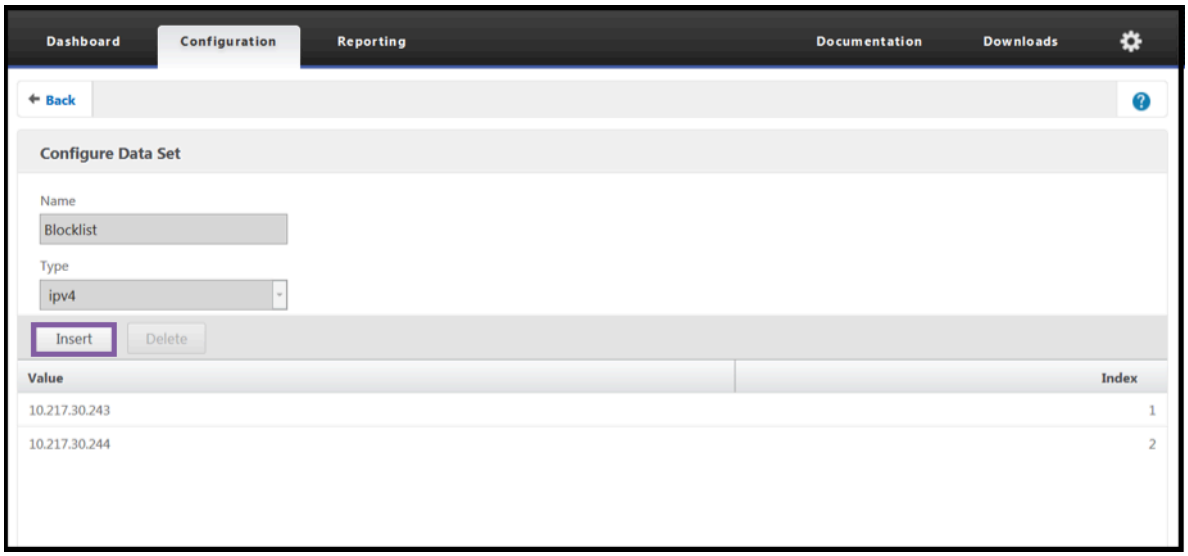
1. On the configuration tab, navigate to **Security > Reputation**. Under **Settings**, click **Change Reputation Settings** to configure a proxy server. You can also enable or disable the reputation feature. **Proxy Server** can be an IP address or a fully qualified domain name (FQDN). **Proxy port** accepts values between [1–65535].

Create an allow list and a block list of client IP addresses using the GUI

1. On the **Configuration** tab, navigate to **AppExpert > Data Sets**.
2. Click **Add**.



- In the **Create Data Set** (or **Configure Data set**) pane, provide a meaningful name for the list of the IP addresses. The name must reflect the purpose of the list.
- Select **Type** as **IPv4**.
- Click **Insert** to add an entry.



- In the **Configure Policy dataset binding** pane, add an IPv4 format IP address in the Value input box.
- Provide an index.
- Add a comment that explains the purpose of the list. This step is optional, but is recommended because a descriptive comment is helpful in managing the list.

Similarly, you can create a block list and add the IP addresses that are to be considered malicious.

Also see, Pattern sets and data sets for more details regarding using data sets and configuring default syntax policy expressions.

Configure an application firewall policy by using the Citrix ADC GUI

1. On the **Configuration** tab, navigate to **Security > Application Firewall > Policies > Firewall**. Click **Add** to add a policy using the PI expressions to use IP reputation.

You can also use the Expression editor to build your own policy expression. The list shows preconfigured options that are useful for configuring an expression using the threat categories.

Highlights

- Quickly and accurately stop bad traffic at the network's edge from known malicious IP addresses posing different types of threats. You can block the request without parsing the body.
- Dynamically configure IP reputation functionality for multiple applications.
- Secure your network against data breach without a performance penalty, and consolidate protections onto a single services fabric using fast and easy deployments.
- You can do IP Reputation checks on source and destination IPs.
- You can also inspect the headers to detect malicious IPs.
- IP reputation check is supported in both forward proxy and reverse proxy deployments.
- The IP Reputation process connects with Webroot and updates the database every 5 minutes.
- Each node in the High Availability (HA) or Cluster deployment gets the database from Webroot.
- The IP reputation data is shared across all partitions in admin-partition deployments.
- You can use an AppExpert data set to create lists of IP addresses to add exceptions for IPs blocklisted in the Webroot database. You can also create your own customized block list to designate specific IPs as malicious.
- The iprep.db file is created in the `/var/nslog/iprep` folder. Once created, it is not deleted even if the feature is disabled.
- When the reputation feature is enabled, the Citrix ADC Webroot database is downloaded. After that, it is updated every 5 minutes.
- The Webroot database version is 1.
- The minor version gets updated every day. The update version is incremented after every 5 minutes and is reset back to 1 when the minor version is incremented.
- PI expressions enable you to use IP reputation with other features, such as responder and rewrite.
- The IP addresses in the database are in decimal notation.

Debugging tips

- If you cannot see the reputation feature in the GUI, verify that you have the right license.
- Monitor the messages in `var/log/iprep.log` for debugging.
- **Webroot connectivity**: If you see the `ns iprep: Not able to connect/resolve WebRoot` message, make sure that the appliance has internet access and DNS is configured.
- **Proxy server**: If you see the `ns iprep: iprep_curl_download: 88 curl_easy_perform failed. Error code: 5 Err msg:couldnt resolve proxy name` message, make sure that the proxy server configuration is accurate.
- **IP Reputation feature not working**: The IP Reputation process takes about five minutes to start after you enable the reputation feature. The IP reputation feature might not work for that duration.
- **Database download**: If the IP DB data download is failing after enabling the IP Reputation fea-

ture, the following error is seen in the logs.

```
iprep: iprep_curl_download:86 curl_easy_perform failed. Error code:7 Err
msg:Couldn't connect to server
```

Solution: Allow the out-bound traffic to the following URLs or configure a proxy to resolve the issue.

```
1 localdb-ip-daily.brightcloud.com:443
2 localdb-ip-rtu.brightcloud.com:443
3 api.bcti.brightcloud.com:443
4 <!--NeedCopy-->
```

SSL offload and acceleration

September 14, 2021

A Citrix ADC appliance configured for SSL acceleration transparently accelerates SSL transactions by offloading SSL processing from the server. To configure SSL offloading, you configure a virtual server to intercept and process SSL transactions, and send the decrypted traffic to the server (unless you configure end-to-end encryption, in which case the traffic is re-encrypted). Upon receiving the response from the server, the appliance completes the secure transaction with the client. From the client's perspective, the transaction seems to be directly with the server. A Citrix ADC configured for SSL acceleration also performs other configured functions, such as load balancing.

Configuring SSL offloading requires an SSL certificate and key pair, which you must obtain if you do not already have an SSL certificate. Other SSL-related tasks that you might need to perform include managing certificates, managing certificate revocation lists, configuring client authentication, and managing SSL actions and policies.

A non-FIPS Citrix ADC appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as a hardware security module (HSM).

All Citrix ADC appliances that do not support a FIPS card (including virtual appliances) support the Thales nShield® Connect and SafeNet external HSMs. (MPX 9700/10500/12500/15500 appliances do not support an external HSM.)

Note: FIPS-related options for some of the SSL configuration procedures described in this document are specific to a FIPS-enabled Citrix ADC appliance.

SSL offloading configuration

September 14, 2021

To configure SSL offloading, you must enable SSL processing on the Citrix ADC appliance and configure an SSL based virtual server. The virtual server will intercept SSL traffic, decrypt the traffic, and forward it to a service that is bound to the virtual server. To secure time-sensitive traffic, such as media streaming, you can configure a DTLS virtual server. To enable SSL offloading, you must import a valid certificate and key and bind the pair to the virtual server.

Enable SSL

To process SSL traffic, you must enable SSL processing. You can configure SSL based entities, such as virtual servers and services, without enabling SSL processing. However, they do not work until SSL processing is enabled.

Enable SSL processing by using the CLI

At the command prompt, type:

```
enable ns feature ssl

show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature SSL
 Done
show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 9)     SSL Offloading                  SSL                 ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```


Enable SSL processing by using the GUI

Navigate to **System > Settings** and, in the **Modes and Features** group, click **Configure Basic Features**, and click **SSL Offloading**.

Configure services

On the Citrix ADC appliance, a service represents a physical server or an application on a physical server. Once configured, services are in the disabled state until the appliance can reach the physical server on the network and monitor its status.

Add a service by using the CLI

At the command prompt, type the following commands to add a service and verify the configuration:

```
add service <name> (<IP> | <serverName>) <serviceType> <port>
show service <serviceName>
<!--NeedCopy-->
```

Example:

```
add service sslsvc 198.51.100.225 SSL 443

Done

sh ssl service sslsvc

            Advanced SSL configuration for Back-end SSL Service sslsvc:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Server Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
```



```
22              Zero RTT Early Data: ???
23              DHE Key Exchange With PSK: ???
24              Tickets Per Authentication Context: ???
25
26              ECC Curve: P_256, P_384, P_224, P_521
27
28      1)      Cipher Name: DEFAULT_BACKEND
29              Description: Default cipher list for Backend SSL session
30       Done
31  <!--NeedCopy-->
```

Modify or remove a service by using the CLI

To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service.

To remove a service, use the rm service command, which accepts only the <name> argument.

```
1  rm service <servicename>
2  <!--NeedCopy-->
```

Example:

```
1  rm service sslsvc
2  <!--NeedCopy-->
```

To modify a service, use the set service command, select any parameter, and change it's setting.

```
1  set service <name> (<IP> | <serverName>) <serviceType> <port>
2  <!--NeedCopy-->
```

Example:

```
1  set service sslsvc 198.51.100.225 SSL 443
2  <!--NeedCopy-->
```

Configure a service by using the GUI

Navigate to **Traffic Management > Load Balancing > Services**, create a service, and specify the protocol as SSL.

SSL virtual server configuration

Secure sessions require establishing a connection between the client and an SSL-based virtual server on the Citrix ADC appliance. The SSL virtual server intercepts SSL traffic, decrypts it and processes it before sending it to services that are bound to the virtual server.

Note: The SSL virtual server is marked as down on the Citrix ADC appliance until a valid certificate / key pair and at least one service are bound to it. An SSL based virtual server is a load balancing virtual server of protocol type SSL or SSL_TCP. The load balancing feature must be enabled on the Citrix ADC appliance.

Add an SSL-based virtual server by using the CLI

At the command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

```
add lb vserver <name> (serviceType) <IPAddress> <port>
show ssl vserver <name>
<!--NeedCopy-->
```

Example:

```
add lb vserver sslvs SSL 192.0.2.240 443
Done

sh ssl vserver sslvs

            Advanced SSL configuration for VServer sslvs:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED          Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
```



```
                Push Encryption Trigger: Always
                Send Close-Notify: YES
                Strict Sig-Digest Check: DISABLED
                Zero RTT Early Data: DISABLED
                DHE Key Exchange With PSK: NO
                Tickets Per Authentication Context: 1
                ECC Curve: P_256, P_384, P_224, P_521

     1)         Cipher Name: DEFAULT
                Description: Default cipher list with encryption strength
                    >= 128bit
      Done
<!--NeedCopy-->
```

Modify or remove an SSL-based virtual server by using the CLI

To modify the load balancing properties of an SSL virtual server, use the `set lb vserver` command. The set command is similar to the `add lb vserver` command, except that you enter the name of an existing virtual server. To modify the **SSL** properties of an SSL-based virtual server, use the `set ssl vserver` command. For more information, see the "SSL virtual server parameters" section later in this page.

To remove an SSL virtual server, use the `rm lb vserver` command, which accepts only the `<name>` argument.

Configure an SSL-based virtual server by using the GUI

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, create a virtual server, and specify the protocol as SSL.

Bind services to the SSL virtual server

The ADC appliance forwards decrypted SSL data to servers in the network. To forward data, services representing these physical servers must be bound to the virtual server that receives the SSL data.

Typically, the link between the ADC appliance and the physical server is secure. Therefore, data transfer between the appliance and the physical server does not have to be encrypted. However, you can provide end-to end-encryption by encrypting data transfer between the appliance and the server. For details, see Configure SSL offloading with end to end encryption.

Note: Enable the load balancing feature on the ADC appliance before you bind services to the SSL based virtual server.

Bind a service to a virtual server by using the CLI

At the command prompt, type the following commands to bind the service to the virtual server and verify the configuration:

```
bind lb vserver <name> <serviceName>
show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind lb vserver sslvs sslsvc
     Done

sh lb vserver sslvs

            sslvs (192.0.2.240:443) - SSL       Type: ADDRESS
            State: DOWN[Certkey not bound]
            Last state change was at Wed May  2 11:43:04 2018
            Time since last state change: 0 days, 00:13:21.150
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Appflow logging: ENABLED
            No. of Bound Services :  1 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION      BackupMethod:
               ROUNDROBIN
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none
            L2Conn: OFF
            Skip Persistency: None
            Listen Policy: NONE
            IcmpResponse: PASSIVE
            RHIstate: PASSIVE
            New Service Startup Request Rate: 0 PER_SECOND, Increment
               Interval: 0
            Mac mode Retain Vlan: DISABLED
            DBS_LB: DISABLED
            Process Local: DISABLE
            Traffic Domain: 0
```



```
            TROFS Persistence honored: ENABLED
            Retain Connections on Cluster: NO
    1) sslsvc (198.51.100.225: 443) - SSL State: DOWN       Weight: 1
     Done
<!--NeedCopy-->
```

Unbind a service from a virtual server by using the CLI

At the command prompt, type the following command:

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
unbind lb vserver sslvs sslsvc
     Done
<!--NeedCopy-->
```

Bind a service to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server and click the **Load Balancing Virtual Server Service Bindings** tile under the **Services and Service Groups** section.
3. In the **Load Balancing Virtual Server Service Binding** page, click **Add Bindings** tab, click **Click to select** under **Select Service**, and select the check box next to the service to be bound.
4. Click **Select** and click **Bind**.

Configure a server name indication (SNI) virtual server for secure hosting of multiple sites

Virtual hosting is used by web servers to host more than one domain name with the same IP address. The appliance supports hosting of multiple secure domains by offloading SSL processing from the web servers using transparent SSL services or virtual server-based SSL offloading. However, when multiple websites are hosted on the same virtual server, the SSL handshake is completed before the expected host name is sent to the virtual server. As a result, the appliance cannot determine which certificate to present to the client after a connection is established. This problem is resolved by enabling SNI on the virtual server. SNI is a Transport Layer Security (TLS) extension used by the client to provide the host name during handshake initiation. The ADC appliance compares this host name to the common name

and, if it does not match, compares it to the subject alternative name (SAN). If the name matches, the appliance presents the corresponding certificate to the client.

A wildcard SSL certificate helps enable SSL encryption on multiple subdomains if the same organization controls these domains and the second-level domain name is the same. For example, a wildcard certificate issued to a sports network using the common name "*.sports.net" can be used to secure domains, such as "login.sports.net" and "help.sports.net". It cannot secure the "login.ftp.sports.net" domain.

Note:
On an ADC appliance, only domain name, URL, and email ID DNS entries in the **SAN** field are compared.

You can bind multiple server certificates to a single SSL virtual server or transparent service using the -SNICert option. The virtual server or service issues these certificates if SNI is enabled on the virtual server or service. You can enable SNI at any time.

Bind multiple server certificates to a single SSL virtual server by using the CLI

At the command prompt, type the following commands to configure SNI and verify the configuration:

```
set ssl vserver <vServerName>@ [-SNIEnable ( ENABLED | DISABLED )]

bind ssl vserver <vServerName>@ -certkeyName <string> -SNICert

show ssl vserver <vServerName>
<!--NeedCopy-->
```

To bind multiple server certificates to a transparent service by using the CLI, replace `vserver` with service and `vservername` with service name in the preceding commands.

Note: Create the SSL service with the `-clearTextPort 80` option.

Bind multiple server certificates to a single SSL virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open an SSL virtual server and, in **Certificates**, select **Server Certificate**.
3. Add a certificate or select a certificate from the list, and click **Server Certificate for SNI**.
4. In **Advanced Settings**, select **SSL Parameters**.
5. Click **SNI Enable**.

Support for SNI on the back-end service

> **Note**: SNI is not supported on a DTLS back-end service.

The Citrix ADC appliance supports Server Name Indication (SNI) at the back end. That is, the common name is sent as the server name in the client hello to the back-end server for successful completion of the handshake. This support helps meet federal system integrator customer security requirements. Also, SNI provides the advantage of using only one port instead of opening hundreds of different IP addresses and ports on a firewall.

Federal system integrator customer security requirements include support for Active Directory Federation Services (ADFS) 3.0 in 2012R2 and WAP servers. To meet this requirement, support for SNI at the back end on a Citrix ADC appliance is required.

> **Note:**
>
> For SNI to work, the server name in the client hello must match the host name configured on the back-end service that is bound to an SSL virtual server. For example, if the host name of the back-end server is www.mail.example.com, the SNI-enabled back-end service must be configured with the server name as https://www.mail.example.com. And this host name must match the server name in the client hello.

Support for dynamic SNI on the back-end service

The Citrix ADC appliance supports dynamic SNI on the back-end TLS connections. That is, the appliance learns the SNI in the client connection and uses it in the server-side connection. You no longer need to specify a common name in the SSL service, service group, or profile. The common name received in the SNI extension of the Client Hello message is forwarded to the back-end SSL connection.

Earlier, you had to configure static SNI on SSL services, service groups, and SSL profiles. As a result, only the configured static SNI extension was sent to the server. If a client needed to access multiple domains at the same time, the ADC appliance was not able to send the SNI received from the client to the back-end service. Instead, it sent the static common name that was configured. Now, if the back-end server is configured for multiple domains, the server can respond with the correct certificate based on the SNI received in the Client Hello message from the appliance.

Point to Note:

- SNI must be enabled on the front end and the correct SNI certificate bound to the SSL virtual server. If you don't enable SNI on the front end, the SNI information is not passed to the back end.
- When server authentication is enabled, the server certificate is verified by the CA certificate and the common name/SAN entries in the server certificate are matched with the SNI. Therefore, the CA certificate must be bound to the service.
- Reuse of back-end connection and SSL session is based on SNI when dynamic SNI is enabled.

SSL monitors do not send SNI when dynamic SNI is enabled. For SNI based probing, attach a back-end profile on which static SNI is configured to the SSL monitors. The monitor must be configured with

the same custom header as SNI.

Configure SNI on the back-end service by using the CLI

At the command prompt, type:

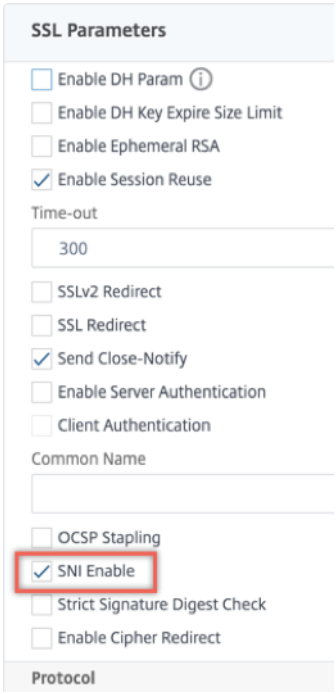
```
add service <name>  <IP>  <serviceType>  <port>

add lb vserver <name>  <IPAddress> <serviceType>  <port>

bind lb vserver <name> <serviceName>

set ssl service <serviceName> -SNIEnable ENABLED -commonName <string>

set ssl profile <name> -SNIEnable ENABLED
<!--NeedCopy-->
```

Example:

```
    add service service_ssl 198.51.100.100 SSL 443

    add lb vserver ssl-vs 203.0.113.200 SSL 443

    bind lb vserver ssl-vs service_ssl

    set ssl service service_ssl -SNIEnable ENABLED  - commonName www.
       example.com

    set ssl profile sslprof -SNIEnable ENABLED
<!--NeedCopy-->
```

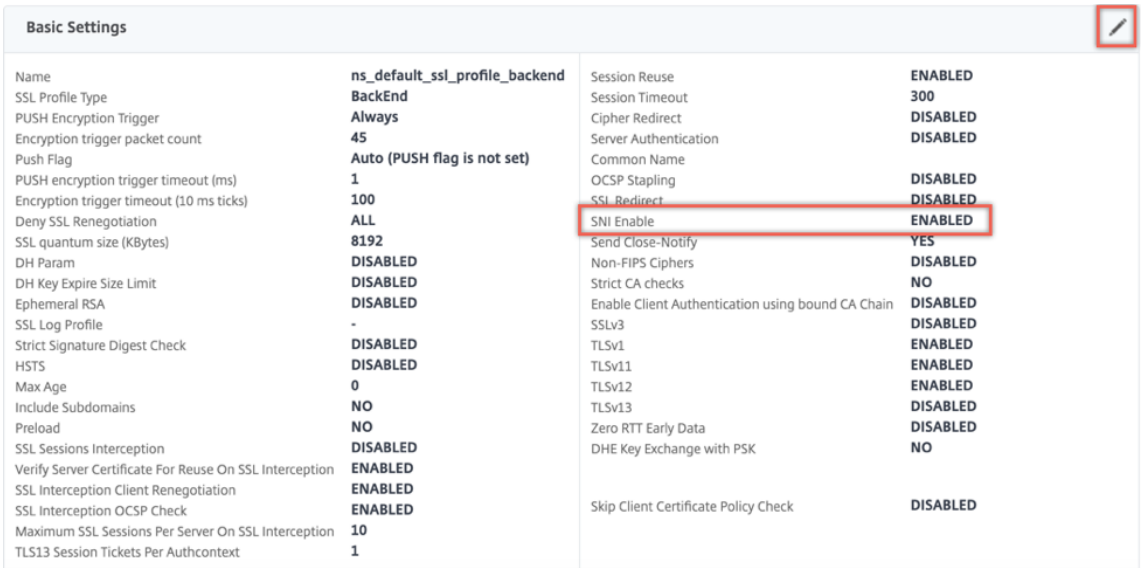
Configure SNI on the back-end service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Select an SSL service, and in **Advanced Settings**, click **SSL Parameters**.
3. Click **SNI Enable**.



Configure SNI on the SSL profile by using the GUI

1. Navigate to **System > Profiles > SSL Profile**.
2. Click **Add**.
3. In **Basic Settings**, select **SNI Enable**.



4. Click **OK**.

Bind a secure monitor to an SNI-enabled back-end service

You can bind secure monitors of type HTTP, HTTP-ECV, TCP, or TCP-ECV to the back-end services and service groups that support SNI. However, the monitor probes do not send the SNI extension if dynamic SNI is enabled. To send SNI probes, enable static SNI in the back-end SSL profile and bind the profile to the monitor. Set the custom header in the monitor to the server name that is sent as the SNI extension in the client hello of the monitor probe.

Configure and bind a secure monitor to an SNI-enabled back-end service by using the CLI

At the command prompt, type:

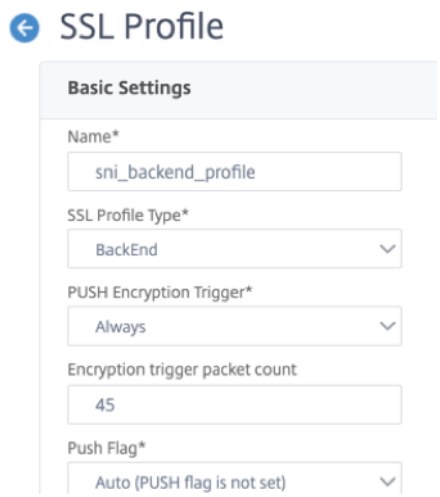
```
add lb monitor <monitorName> <type> -secure YES
add ssl profile <name> -sslProfileType BackEnd
set lb monitor <monitorName> <type> -customHeaders <string> -sslprofile
    <backend ssl profile>
set ssl profile <name> -sniEnable ENABLED -commonName <string>
bind service <name> -monitorName <string>
<!--NeedCopy-->
```

Example:

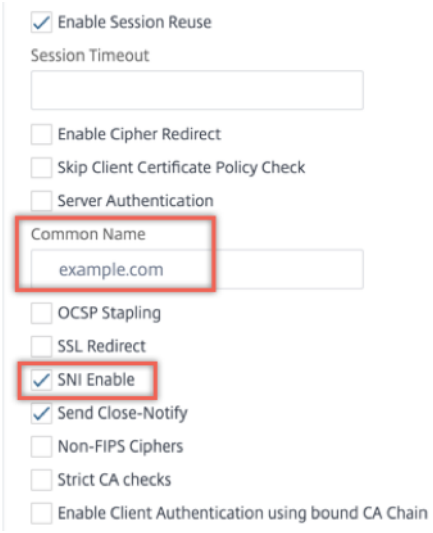
```
add ssl profile sni_backend_profile -sslProfileType BackEnd
set ssl profile sni_backend_profile -sniEnable ENABLED -commonName
   example.com
add lb monitor http-ecv-mon HTTP-ECV -secure YES
set monitor http-ecv-mon HTTP-ECV -customHeaders "Host: example.com\r\n
   " -sslprofile sni_backend_profile
bind service ssl_service  - monitorName http-ecv-mon
<!--NeedCopy-->
```

Configure and bind a secure monitor to an SNI enabled back-end service by using the GUI

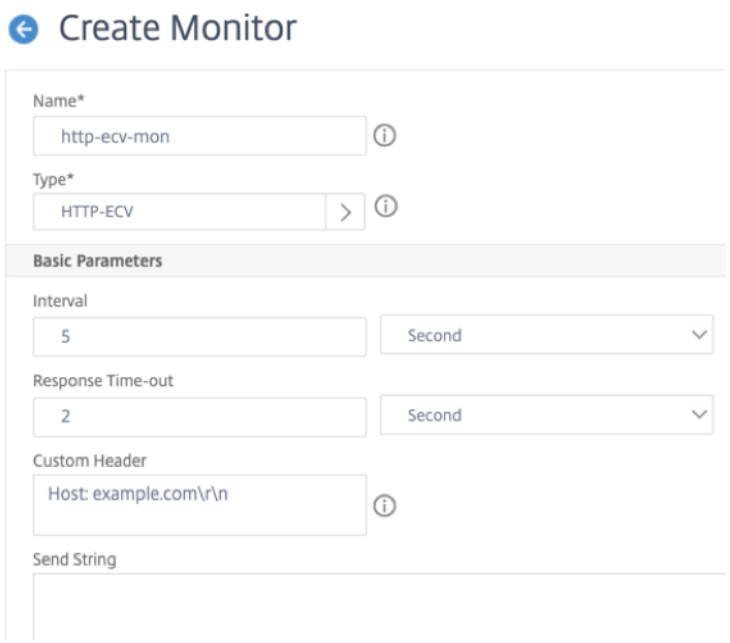
1. Navigate to **System > Profiles > SSL Profiles**.
2. Click **Add**.
3. Specify a name for the profile and in **SSL Profile Type**, select **Backend**.



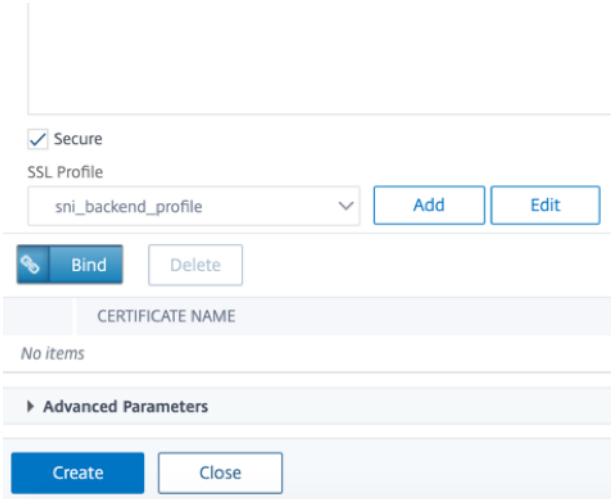
4. Specify the common name (same as host header) and select **SNI Enable**.



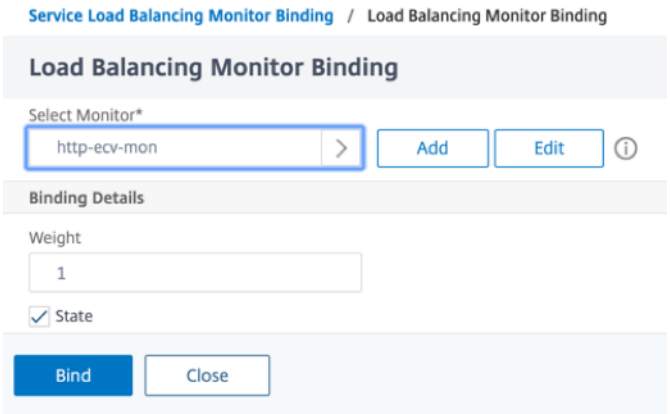
5. Click **OK**.
6. Navigate to **Traffic Management > Load Balancing > Monitor**.
7. Click **Add**.
8. Specify a name for the monitor. In **Type**, select HTTP, HTTP-ECV, TCP, or TCP-ECV.
9. Specify a **Custom Header**.



10. Select **Secure**.
11. In **SSL Profile**, select the back-end SSL profile created in the preceding steps.
12. Click **Create**.



13. Navigate to **Traffic Management > Load Balancing > Services**.
14. Select an SSL service and click **Edit**.
15. In **Monitors**, click **Add Binding**, select the monitor created in the preceding steps, and click **Bind**.



Configure and bind a secure monitor to an SNI-enabled back-end service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitor**.
2. Add a monitor of type **HTTP-ECV** or **TCP-ECV**, and specify a **Custom Header**.
3. Select **Create**.
4. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
5. Select an SSL service and click **Edit**.
6. In **Monitors**, click **Add Binding**, select the monitor created in step 3, and click **Bind**.

Add or update a certificate-key pair

> **Notes:**
>
> If you don't have an existing certificate and key, see Create a certificate.
>
> To create an ECDSA certificate-key pair, click Create an ECDSA certificate-key pair.
>
> From build 41.x, certificate names of up to 63 characters are supported.
>
> From release 13.0 build 79.x, password protected certificate-key pairs are always added successfully. Earlier, if strong password option was enabled on a Citrix ADC appliance, sometimes the password protected certificate-key pairs were not added. However, the certificate-key configuration is lost if you downgrade to an earlier build. Also, in the NITRO API response for certificate-key pairs, the `passplain` variable is sent instead of the `passcrypt` variable.

For any SSL transaction, the server needs a valid certificate and the corresponding private and public key pair. The SSL data is encrypted with the server's public key, which is available through the server's certificate. Decryption requires the corresponding private key. The password of the private key used while adding an SSL certificate-key pair is saved using a unique encryption key for each Citrix ADC appliance.

The ADC appliance offloads SSL transactions from the server. Therefore, the server's certificate and private key must be present on the appliance, and the certificate must be paired with its correspond-

ing private key. This certificate-key pair must be bound to the virtual server that processes the SSL transactions.

Note: The default certificate on a Citrix ADC appliance is 2048 bits. In earlier builds, the default certificate was 512 bits or 1024 bits. After upgrading to release 11.0, you must delete all your old certificate-key pairs starting with `"ns-"`, and then restart the appliance to automatically generate a 2048-bit default certificate.

Both the certificate and the key must be in local storage on the Citrix ADC appliance before they can be added to the appliance. If your certificate or key file is not on the appliance, upload it to the appliance before you create the pair.

Important: Certificates and keys are stored in the /nsconfig/ssl directory by default. If your certificates or keys are stored in any other location, you must provide the absolute path to the files on the Citrix ADC appliance. The Citrix ADC FIPS appliances do not support external keys (non-FIPS keys). On a FIPS appliance, you cannot load keys from a local storage device such as a hard disk or flash memory. The FIPS keys must be present in the Hardware Security Module (HSM) of the appliance.

Only RSA keys are supported on Citrix ADC appliances.

Set the notification period and enable the expiry monitor to issue a prompt before the certificate expires.

The Citrix ADC appliance supports the following input formats of the certificate and the private-key files:

- PEM - Privacy Enhanced Mail
- DER - Distinguished Encoding Rule
- PFX - Personal Information Exchange

The software automatically detects the format. Therefore, you are no longer required to specify the format in the inform parameter. If you do specify the format (correct or incorrect), the software ignores it. The format of the certificate and the key file must be the same.

Note: A certificate must be signed by using one of the following hash algorithms:

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384 (supported only on the front end)
- SHA-512 (supported only on the front end)

An MPX appliance supports certificates of 512 or more bits, up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service

- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back-end server
- 4096-bit client certificate (if client authentication is enabled on the virtual server)

A VPX virtual appliance supports certificates of 512 or more bits, up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back-end server
- 4096-bit client certificate (if client authentication is enabled on the virtual server)

> **Note**
>
> A Citrix ADC SDX appliance supports certificates of 512 or more bits. Each Citrix ADC VPX instance hosted on the appliance supports the preceding certificate sizes for a VPX virtual appliance. However, if an SSL chip is assigned to an instance, that instance supports the certificate sizes supported by an MPX appliance.

Add a certificate-key pair by using the CLI

At the command prompt, type the following commands to add a certificate-key pair and verify the configuration:

```
add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password
    ]) | -fipsKey <string>] [-inform ( DER | PEM )] [<passplain>] [-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]]

show ssl certKey [<certkeyName>]
<!--NeedCopy-->
```

Example:

```
add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -
    password ssl -expiryMonitor ENABLED -notificationPeriod 30
 Done
Note: For FIPS appliances, replace -key with -fipskey

show ssl certKey sslckey
        Name: sslckey           Status: Valid,   Days to expiration
            :8418
        Version: 3
        Serial Number: 01
        Signature Algorithm: md5WithRSAEncryption
```



```
        Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
        Validity
                Not Before: Jul 15 02:25:01 2005 GMT
                Not After : Nov 30 02:25:01 2032 GMT
        Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
 Done
<!--NeedCopy-->
```

Update or remove a certificate-key pair by using the CLI

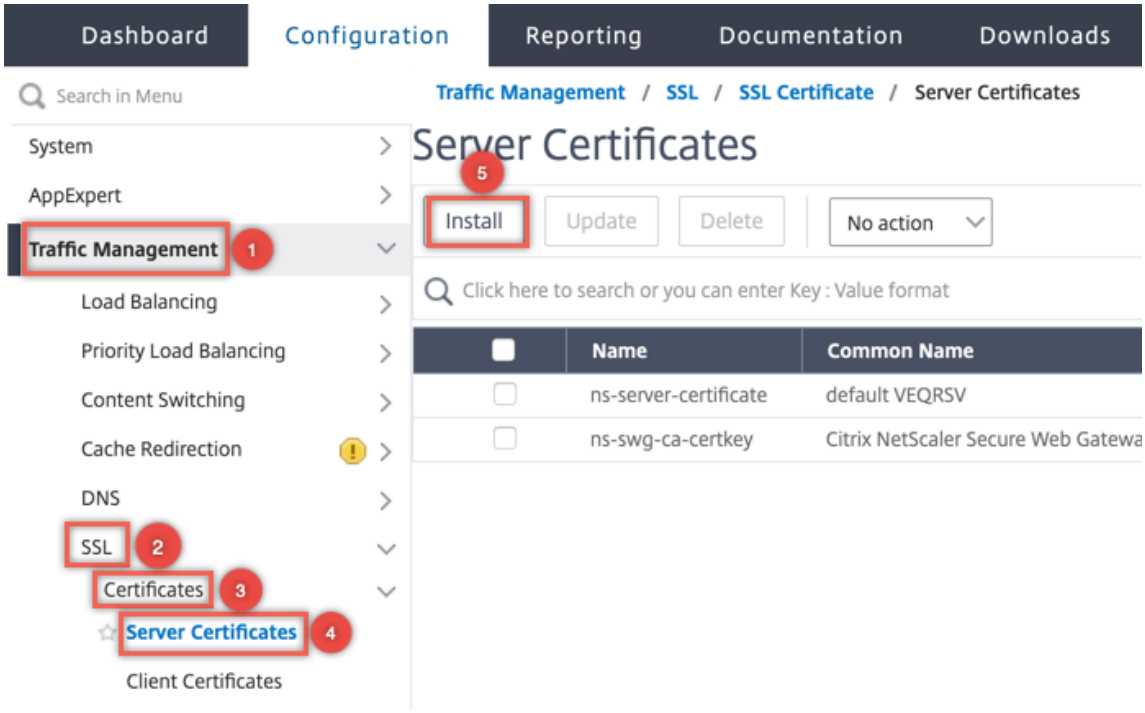
To modify the expiry monitor or notification period in a certificate-key pair, use the `set ssl certkey` command. To replace the certificate or key in a certificate-key pair, use the `update ssl certkey` command. The `update ssl certkey` command has an extra parameter for overriding the domain check. For both commands, enter the name of an existing certificate-key pair. To remove an SSL certificate-key pair, use the `rm ssl certkey` command, which accepts only the `<certkeyName>` argument.

Example:

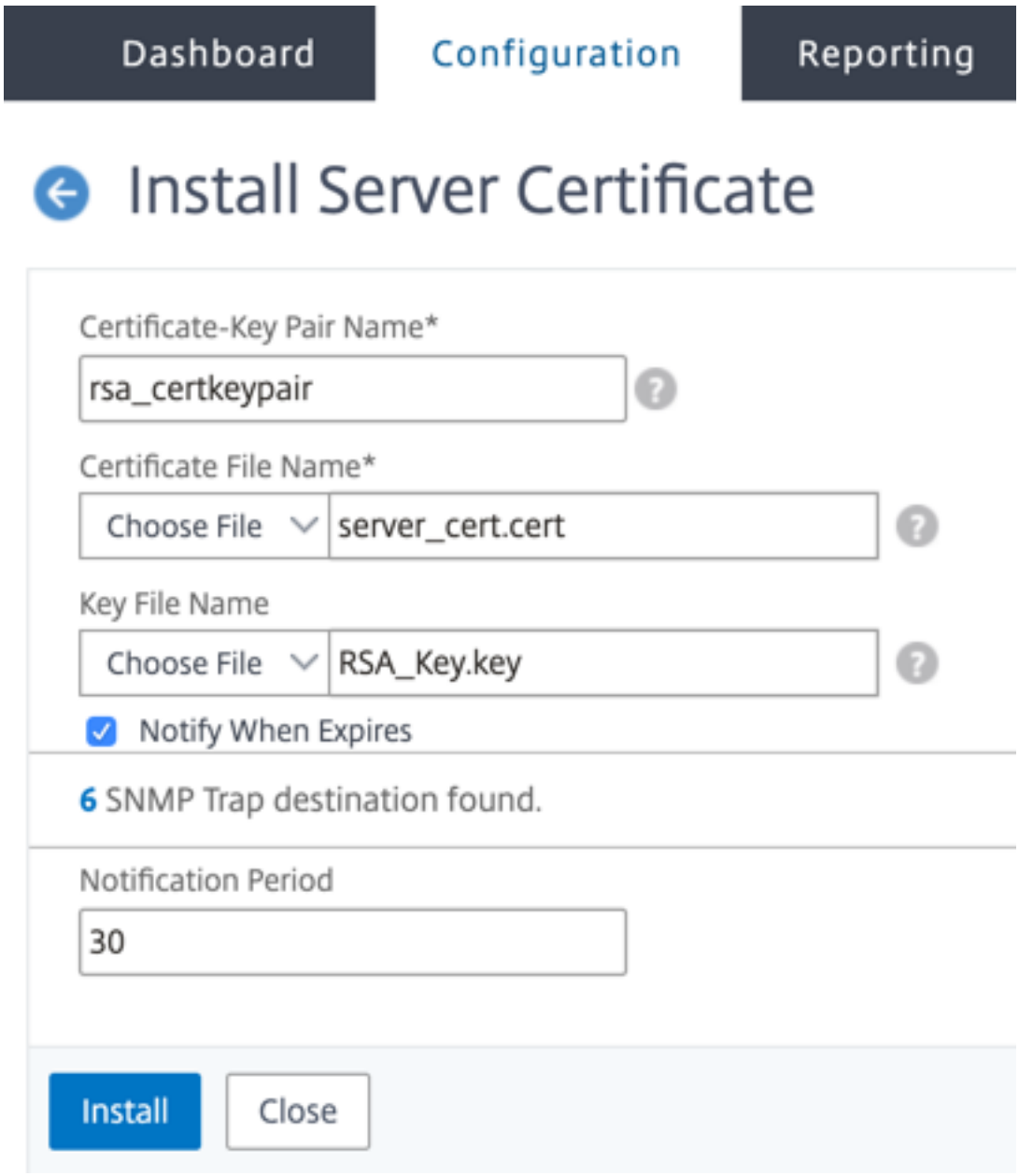
```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED )
       [-notificationPeriod <positive_integer>]]

update ssl certKey <certkeyName> [-cert <string> [-password]] [-key
       <string> | -fipsKey <string>] [-inform <inform>] [-noDomainCheck
          ]
<!--NeedCopy-->
```

Add or update a certificate-key pair by using the GUI

1. Navigate to **Traffic Management > SSL > Certificates > Server**.



2. Enter the values for the following parameters and click **Install**.
   - Certificate-Key Pair Name - Name for the certificate and private-key pair.
   - Certificate File Name - Signed certificate received from the certificate authority.
   - Key File Name - Name of and, optionally, path to the private-key file that is used to form the certificate-key pair.



Bind the certificate-key pair to the SSL virtual server

Important: Link any intermediate certificates to this certificate before binding the certificate to an SSL virtual server. For information about linking certificates, see Create a chain of certificates.

The certificate being used for processing SSL transactions must be bound to the virtual server that receives the SSL data. If you have multiple virtual servers receiving SSL data, a valid certificate-key pair must be bound to each of them.

Use a valid, existing SSL certificate that you have uploaded to the Citrix ADC appliance. As an alternative for testing purposes, create your own SSL certificate on the appliance. Intermediate certificates

created by using a FIPS key on the appliance cannot be bound to an SSL virtual server.

During the SSL handshake, in the certificate request message during client authentication, the server lists the distinguished names (DN) of all the certificate authorities (CA) bound to the server. The server accepts a client certificate only from this list. If you do not want the DN name of a specific CA certificate to be sent to the SSL client, set the `skipCA` flag. This setting indicates that the particular CA certificate's distinguished name must not be sent to the SSL client.

For details on how to create your own certificate, see Managing Certificates.

Note: Citrix recommends that you use only valid SSL certificates issued by a trusted certificate authority.

Bind an SSL certificate-key pair to a virtual server by using the CLI

At the command prompt, type the following commands to bind an SSL certificate-key pair to a virtual server and verify the configuration:

```
1 -  bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName
     > -CA -skipCAName
2 -  show ssl vserver <vServerName>
3 <!--NeedCopy-->
```

Example:

```
1 bind ssl vs vs1 -certkeyName cert2 -CA -skipCAName
2  Done
3 sh ssl vs vs1
4
5  Advanced SSL configuration for VServer vs1:
6
7  DH: DISABLED
8
9  Ephemeral RSA: ENABLED Refresh Count: 0
10
11  Session Reuse: ENABLED Timeout: 120 seconds
12
13  Cipher Redirect: DISABLED
14
15  SSLv2 Redirect: DISABLED
16
17  ClearText Port: 0
18
19  Client Auth: DISABLED
20
```



```
 SSL Redirect: DISABLED

 Non FIPS Ciphers: DISABLED

 SNI: DISABLED

 OCSP Stapling: DISABLED

 HSTS: DISABLED

 IncludeSubDomains: NO

 HSTS Max-Age: 0

 SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED
    TLSv1.2: DISABLED

 Push Encryption Trigger: Always

 Send Close-Notify: YES

 Strict Sig-Digest Check: DISABLED

ECC Curve: P_256, P_384, P_224, P_521

 1) CertKey Name: cert1 CA Certificate OCSPCheck: Optional CA_Name Sent
 2) CertKey Name: cert2 CA Certificate OCSPCheck: Optional CA_Name
    Skipped
 1) Cipher Name: DEFAULT

Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

Unbind an SSL certificate-key pair from a virtual server by using the CLI

If you try to unbind a certificate-key pair from a virtual server by using the `unbind ssl certKey <certkeyName>` command, an error message appears. The error appears because the syntax of the command has changed. At the command prompt, type the following command:

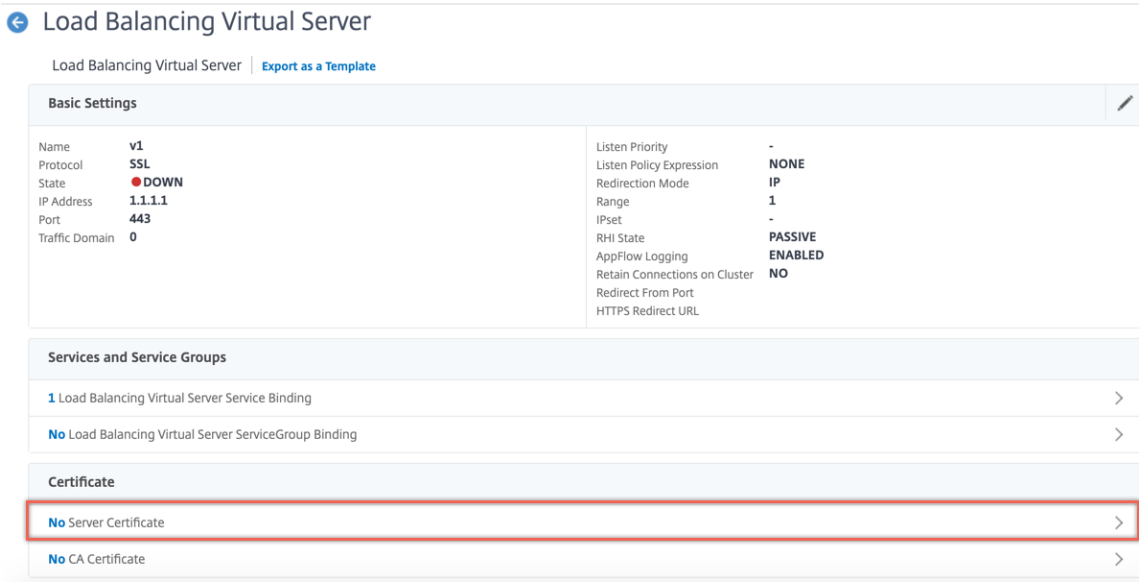
```
unbind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```

Example:

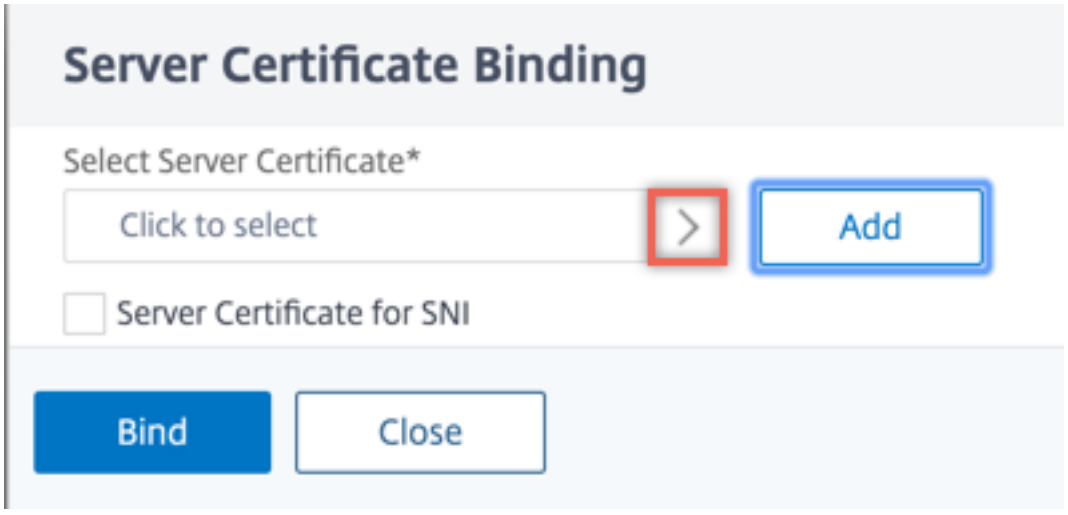

```
unbind ssl vserver vssl -certkeyName sslckey
<!--NeedCopy-->
```

Bind an SSL certificate-key pair to a virtual server by using the GUI

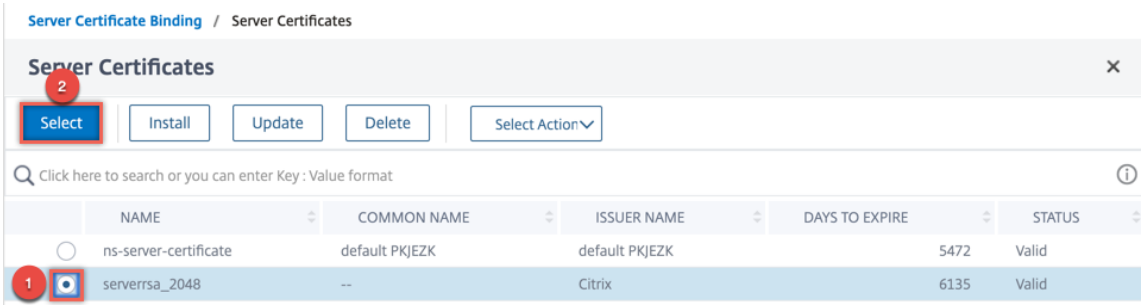
1. Navigate to **Traffic Management > Load Balancing > Virtual Servers** and open an SSL virtual server. Click inside the **Certificate** section.



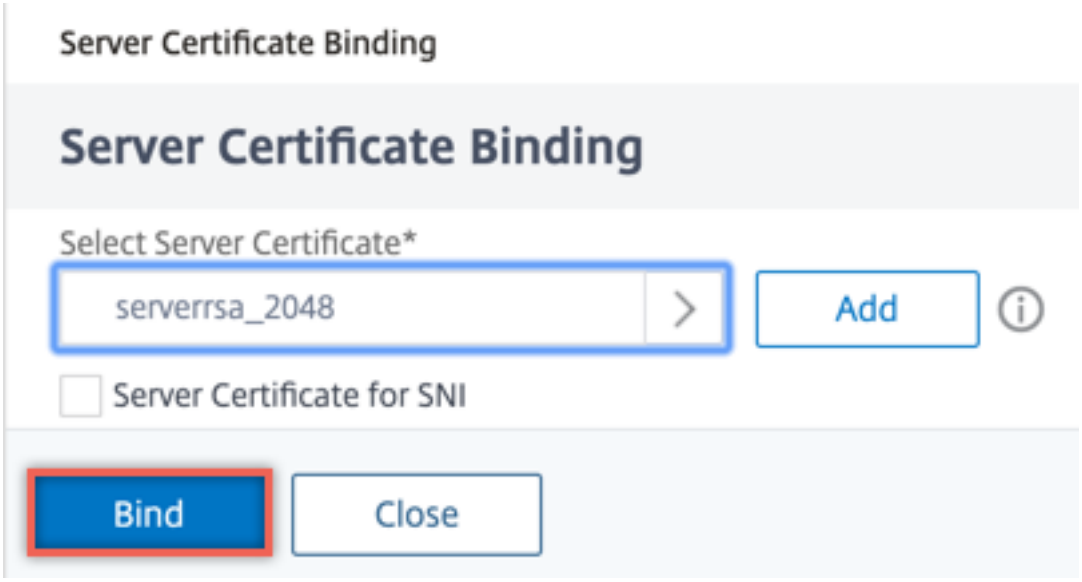
2. Click the arrow to select the certificate-key pair.



3. Select the certificate-key pair from the list.



4. Bind the certificate-key pair to the virtual server. To add a server certificate as an SNI certificate, select **Server Certificate for SNI**.



SSL virtual server parameters

Set the advanced SSL configuration for an SSL virtual server. You can also set many of these parameters in an SSL profile. For information about the parameters that can be set in an SSL profile, see SSL profile parameters.

Set SSL virtual server parameters by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName>@ [-clearTextPort <port>] [-dh ( ENABLED |
    DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-
    dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED |
    DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED |
    DISABLED )[-sessTimeout <positive_integer>]] [-cipherRedirect (
    ENABLED | DISABLED ) [-cipherURL <URL>]] [-sslv2Redirect ( ENABLED |
     DISABLED )[-sslv2URL <URL>]] [-clientAuth ( ENABLED | DISABLED ) [-
```



```
    clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
    DISABLED )][-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl2 (
    ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 (
    ENABLED | DISABLED )] [-tls11 ( ENABLED | DISABLED )] [-tls12 (
    ENABLED | DISABLED )][-tls13 ( ENABLED | DISABLED )] [-SNIEnable (
    ENABLED | DISABLED )][-ocspStapling ( ENABLED | DISABLED )] [-
    pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES | NO )] [-
    dtlsProfileName <string>] [-sslProfile <string>] [-HSTS ( ENABLED |
    DISABLED )][-maxage <positive_integer>] [-IncludeSubdomains ( YES |
    NO )][-strictSigDigestCheck ( ENABLED | DISABLED )] [-
    zeroRttEarlyData (ENABLED | DISABLED )] [-
    tls13SessionTicketsPerAuthContext <positive_integer>] [-
    dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

Diffie-Hellman (DH) parameters

To use ciphers on the appliance that require a DH key exchange to set up the SSL transaction, enable DH key exchange on the appliance. Configure other settings based on your network.

To list the ciphers for which DH parameters must be set by using the CLI, type: sh cipher DH.

To list the ciphers for which DH parameters must be set by using the configuration utility, navigate to **Traffic Management > SSL > Cipher Groups**, and double-click **DH**.

For details on how to enable DH key exchange, see Generate a Diffie-Hellman (DH) key.

Configure DH parameters by using the CLI

At the command prompt, type the following commands to configure DH parameters and verify the configuration:

```
- `set ssl vserver <vserverName> -dh <Option> -dhCount <
    RefreshCountValue> -filepath <string>
- show ssl vserver <vServerName>`
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.
    cert -dhCount 1000
Done

show ssl vserver vs-server

```



```
6               Advanced SSL configuration for VServer vs-server:
7               DH: ENABLED
8               Ephemeral RSA: ENABLED          Refresh Count: 1000
9               Session Reuse: ENABLED          Timeout: 120 seconds
10              Cipher Redirect: DISABLED
11              SSLv2 Redirect: DISABLED
12              ClearText Port: 0
13              Client Auth: DISABLED
14              SSL Redirect: DISABLED
15              Non FIPS Ciphers: DISABLED
16              SNI: DISABLED
17              OCSP Stapling: DISABLED
18              HSTS: DISABLED
19              HSTS IncludeSubDomains: NO
20              HSTS Max-Age: 0
21              SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
                    ENABLED  TLSv1.2: ENABLED
22
23      1)      Cipher Name: DEFAULT
24              Description: Predefined Cipher Alias
25  Done
26  <!--NeedCopy-->
```

Configure DH parameters by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable DH Param**, and specify a refresh count and file path.

Ephemeral RSA

Ephemeral RSA allows export clients to communicate with the secure server even if the server certificate does not support export clients (1024-bit certificate). If you want to prevent export clients from accessing the secure web object or resource, you need to disable ephemeral RSA key exchange.

By default, this feature is enabled on the Citrix ADC appliance, with the refresh count set to zero (infinite use).

Note:

The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused later when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the system restarts.

Configure ephemeral RSA by using the CLI

At the command prompt, type the following commands to configure ephemeral RSA and verify the configuration:

```
set ssl vserver <vServerName> -eRSA (enabled | disabled) -eRSACount <
   positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
 Done

show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
           ENABLED  TLSv1.2: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

Configure ephemeral RSA by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable Ephemeral RSA**, and specify a refresh count.

Session reuse

For SSL transactions, establishing the initial SSL handshake requires CPU-intensive public key encryption operations. Most handshake operations are associated with the exchange of the SSL session key (client key exchange message). When a client session is idle for some time and is then resumed, the SSL handshake is typically conducted all over again. With session reuse enabled, session key exchange is avoided for session resumption requests received from the client.

Session reuse is enabled on the Citrix ADC appliance by default. Enabling this feature reduces server load, improves response time, and increases the number of SSL transactions per second (TPS) that the server can support.

Configure session reuse by using the CLI

At the command prompt, type the following commands to configure session reuse and verify the configuration:

```
set ssl vserver <vServerName> -sessReuse ( ENABLED | DISABLED ) -
   sessTimeout <positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
 Done

show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
```



```
21          SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
                ENABLED  TLSv1.2: ENABLED
22
23  1)      CertKey Name: Auth-Cert-1       Server Certificate
24
25  1)      Cipher Name: DEFAULT
26          Description: Predefined Cipher Alias
27   Done
28  <!--NeedCopy-->
```

Configure session reuse by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable Session Reuse**, and specify a time for which to keep the session active.

SSL protocol settings

The Citrix ADC appliance supports the SSLv3, TLSv1, TLSv1.1, and TLSv1.2 protocols. Each of these protocols can be set on the appliance as required by your deployment and the type of clients that connect to the appliance.

TLS protocol versions 1.0, 1.1, and 1.2 are more secure than older versions of the TLS/SSL protocol. However, to support legacy systems, many TLS implementations maintain backward compatibility with the SSLv3 protocol. In an SSL handshake, the highest protocol version common to the client and the SSL virtual server configured on the Citrix ADC appliance is used.

In the first handshake attempt, a TLS client offers the highest protocol version that it supports. If the handshake fails, the client offers a lower protocol version. For example, if a handshake with TLS version 1.1 is not successful, the client attempts to renegotiate by offering the TLSv1.0 protocol. If that attempt is unsuccessful, the client reattempts with the SSLv3 protocol. A “man in the middle” (MITM) attacker can break the initial handshake and trigger renegotiation with the SSLv3 protocol, and then exploit a vulnerability in SSLv3. To mitigate such attacks, you can disable SSLv3 or not allow renegotiation using a downgraded protocol. However, this approach might not be practical if your deployment includes legacy systems. An alternative is to recognize a signaling cipher suite value (TLS_FALLBACK_SCSV) in the client request.

A TLS_FALLBACK_SCSV value in a client hello message indicates to the virtual server that the client has previously attempted to connect with a higher protocol version and that the current request is a fallback. If the virtual server detects this value, and it supports a version higher than the one indicated by the client, it rejects the connection with a fatal alert. The handshake succeeds if one of the following conditions is met:

- TLS_FALLBACK_SCSV value is not included in the client hello message.
- The protocol version in the client hello is the highest protocol version supported by the virtual server.

Configure SSL protocol support by using the CLI

At the command prompt, type the following commands to configure SSL protocol support and verify the configuration:

```
set ssl vserver <vServerName> -ssl2 ( ENABLED | DISABLED ) -ssl3 (
    ENABLED | DISABLED ) -tls1 ( ENABLED | DISABLED ) -tls11 ( ENABLED |
     DISABLED ) -tls12 ( ENABLED | DISABLED )

show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-ssl -tls11 ENABLED -tls12 ENABLED
Done

sh ssl vs vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
            DH: DISABLED
            Ephemeral RSA: ENABLED                              Refresh
               Count: 0
            Session Reuse: ENABLED                                Timeout
               : 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            SSLv2: DISABLED        SSLv3: ENABLED    TLSv1.0: ENABLED
               TLSv1.1: ENABLED  TLSv1.2: ENABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
            1 bound certificate:

     1)     CertKey Name: mycert  Server Certificate
            1 configured cipher:

```



```
25      1)      Cipher Name: DEFAULT
26              Description: Predefined Cipher Alias
27
28  Done
29  <!--NeedCopy-->
```

Configure SSL protocol support by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select a protocol to enable.

Close-notify

A close-notify is a secure message that indicates the end of SSL data transmission. A close-notify setting is required at the global level. This setting applies to all virtual servers, services, and service groups. For information about the global setting, see the "Global SSL parameters" section later in this page.

In addition to the global setting, you can set the close-notify parameter at the virtual server, service, or service group level. You therefore have the flexibility of setting the parameter for one entity and unsetting it for another entity. However, make sure that you set this parameter at the global level. Otherwise, the setting at the entity level does not apply.

Configure close-notify at the entity level by using the CLI

At the command prompt, type any of the following commands to configure the close-notify feature and verify the configuration:

1. To configure at the virtual server level, type:

```
1  set ssl vserver <vServerName> -sendCloseNotify ( YES | NO )
2  show ssl vserver <vServerName>
3  <!--NeedCopy-->
```

1. To configure at the service level, type:

```
1  set ssl service <serviceName> -sendCloseNotify ( YES | NO )
2  show ssl service <serviceName>
3  <!--NeedCopy-->
```

1. To configure at the service group level, type:


```
set ssl serviceGroup <serviceGroupName> -sendCloseNotify ( YES | NO )
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver sslvsvr -sendCloseNotify YES

Done
<!--NeedCopy-->
```

Configure the close-notify feature at the entity level by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Send Close-Notify**.

Global SSL parameters

Advanced customization of your SSL configuration addresses specific issues. You can use the `set ssl parameter` command or the configuration utility to specify the following:

- Quantum size to be used for SSL transactions.
- CRL memory size.
- OCSP cache size.
- Deny SSL renegotiation.
- Set the PUSH flag for decrypted, encrypted, or all records.
- Drop requests if the client initiates the handshake for one domain and sends an HTTP request for another domain.
- Set the time after which encryption is triggered.
  Note: The time that you specify applies only if you use the `set ssl vserver` command or the configuration utility to set timer-based encryption.
- NDCPP compliance certificate check – Applies when the appliance acts a client (back-end connection). During certificate verification, ignore the common name if SAN is present in the SSL certificate.
- Enable a heterogeneous cluster of Cavium chip based appliances, such as MPX 14000, and Intel Coleto chip based appliances, such as MPX 15000 appliances with a different number of packet engines. (Support added in release 13.0 build 47.x).
- Enable secure renegotiation at the back end (Support added from release 1.0 build 58.x).
- Adaptive SSL traffic control (Support added in release 13.0 build 58.x).

Configure global SSL parameters by using the CLI

At the command prompt, type the following commands to configure advanced SSL settings and verify the configuration:

```
set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB <
    positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout
    <positive_integer>] [-sendCloseNotify (YES | NO)] [-
    encryptTriggerPktCount <positive_integer>] [-denySSLReneg <
    denySSLReneg>] [-insertionEncoding (Unicode|UTF-8)] [-ocspCacheSize
    <positive_integer>][- pushFlag <positive_integer>] [-
    dropReqWithNoHostHeader (YES | NO)] [-pushEncTriggerTimeout <
    positive_integer>] [-ndcppComplianceCertCheck ( YES | NO)] [-
    heterogeneousSSLHW (ENABLED | DISABLED )]
show ssl parameter
<!--NeedCopy-->
```

Example:

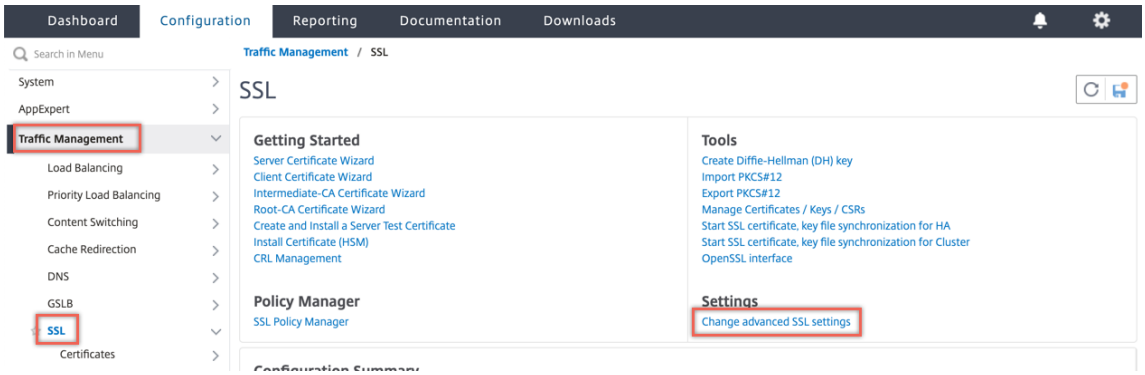
```
set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks
    no -ssltriggerTimeout 100 -sendClosenotify no -
    encryptTriggerPktCount 45 -denySSLReneg NONSECURE -insertionEncoding
     unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES
     -pushEncTriggerTimeout 100 ms -ndcppComplianceCertCheck YES
Done

show ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                    : 8 KB
    Max CRL memory size                                 : 256 MB
    Strict CA checks                                    : NO
    Encryption trigger timeout                          : 100 ms
    Send Close-Notify                                   : NO
    Encryption trigger packet count                     : 45
    Deny SSL Renegotiation                              : NONSECURE
    Subject/Issuer Name Insertion Format                : Unicode
    OCSP cache size                                     : 10 MB
    Push flag                                           : 0x3 (On
        every decrypted and encrypted record)
    Strict Host Header check for SNI enabled SSL sessions : YES
    PUSH encryption trigger timeout                     : 100 ms
    Crypto Device Disable Limit                         : 0
    Global undef action for control policies            : CLIENTAUTH
    Global undef action for data policies               : NOOP
```



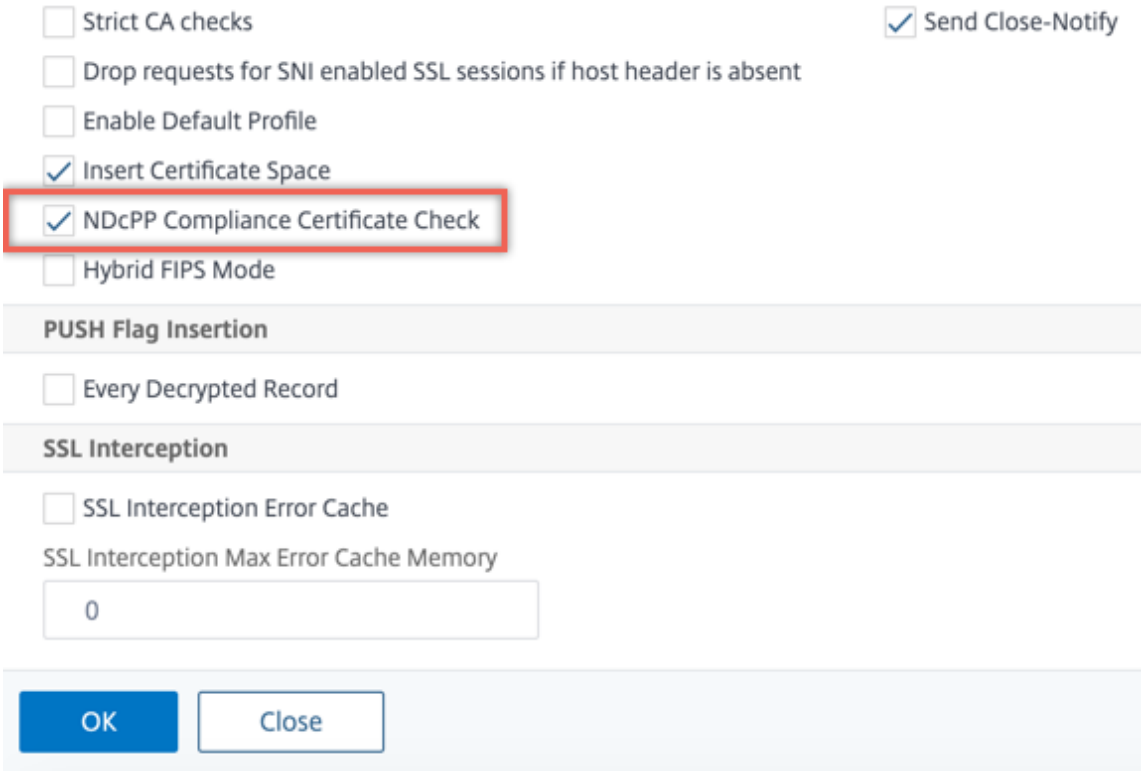
```
    Default profile                                          : DISABLED
    SSL Insert Space in Certificate Header                   : YES
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors       : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services         : NO
    Software Crypto acceleration CPU Threshold               : 0
    Hybrid FIPS Mode                                         : DISABLED
    Signature and Hash Algorithms supported by TLS1.2        : ALL
    SSL Interception Error Learning and Caching              : DISABLED
    SSL Interception Maximum Error Cache Memory              : 0 Bytes
    NDCPP Compliance Certificate Check                       : YES
    Heterogeneous SSL HW (Cavium and Intel Based)            : ENABLED
 Done
<!--NeedCopy-->
```

Configure NDcPP compliance certificate check by using the GUI

1. Navigate to **Traffic Management > SSL** and, in the **Settings** group, select **Change advanced SSL settings**.



2. Select **NDCPP Compliance Certificate Check**. Click **OK**.



Support for secure renegotiation at the back end of a Citrix ADC appliance

Note: This feature is supported in release 13.0 build 58.x and later. In earlier releases and builds, only non-secure renegotiation was supported on the back end.

The feature is supported on the following platforms:
• VPX
• MPX platforms containing N2 or N3 chips
• Intel Coleto SSL chip based platforms

The feature is not yet supported on the FIPS platform.

Secure renegotiation is denied by default on the back end of an ADC appliance. That is, the denySSLReneg parameter is set to ALL (default).

To allow secure renegotiation on the back end, select from one of the following settings for the denySSLReneg parameter:

- NO
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

Enable secure renegotiation by using the CLI

At the command prompt, type:

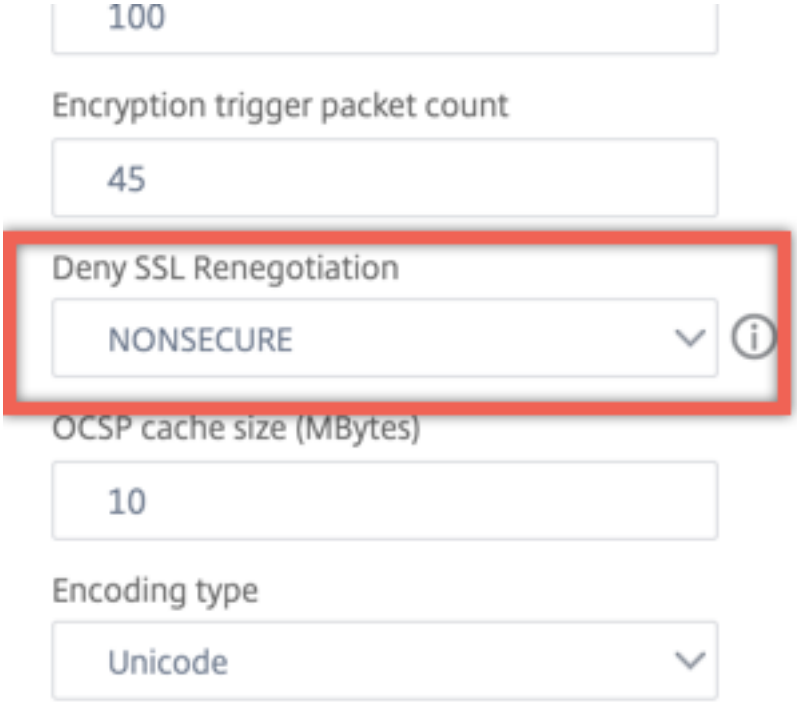
```
set ssl parameter -denySSLReneg <denySSLReneg>
```

Example:

```
set ssl parameter -denySSLReneg NONSECURE
 Done

sh ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                  : 8 KB
    Max CRL memory size                               : 256 MB
    Strict CA checks                                  : NO
    Encryption trigger timeout                        : 100 ms
    Send Close-Notify                                 : YES
    Encryption trigger packet count                   : 45
    Deny SSL Renegotiation                            : NONSECURE
    Subject/Issuer Name Insertion Format              : Unicode
    OCSP cache size                                   : 10 MB
    Push flag                                         : 0x0 (Auto)
    Strict Host Header check for SNI enabled SSL sessions : NO
    Match HTTP Host header with SNI                   : CERT
    PUSH encryption trigger timeout                   : 1 ms
    Crypto Device Disable Limit                       : 0
    Global undef action for control policies          : CLIENTAUTH
    Global undef action for data policies             : NOOP
    Default profile                                   : ENABLED
    SSL Insert Space in Certificate Header            : YES
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services      : NO
    Software Crypto acceleration CPU Threshold        : 0
    Hybrid FIPS Mode                                  : DISABLED
    Signature and Hash Algorithms supported by TLS1.2     : ALL
    SSL Interception Error Learning and Caching       : DISABLED
    SSL Interception Maximum Error Cache Memory       : 0 Bytes
    NDCPP Compliance Certificate Check                : NO
    Heterogeneous SSL HW (Cavium and Intel Based)         : DISABLED
    Crypto Operation Queue Limit                      : 150%
 Done
<!--NeedCopy-->
```


Enable secure renegotiation by using the GUI

1. Navigate to **Traffic Management > SSL > Change advanced SSL settings**.
2. Set **Deny SSL Renegotiation** to any value other than ALL.



Adaptive SSL traffic control

Note: This feature is supported in release 13.0 build 58.x and later.

When high traffic is received on the appliance and the crypto acceleration capacity is full, the appliance starts queuing connections to process later. Currently, the size of this queue is fixed at 64 K and the appliance starts dropping connections if this value is exceeded.

From release 13.0 build 58.x, the user can configure a value that is a percentage of the actual capacity. With this enhancement, the appliance drops new connections if the number of elements in the queue is greater than the limit that is adaptively and dynamically calculated. This approach controls incoming SSL connections and prevents excessive resource consumption and other failures, such as load balancing monitoring failure or slow response to secure applications, on the appliance.

If the queue is empty, the appliance can continue to accept connections. If the queue is not empty, the crypto system has reached its capacity and the appliance starts queuing connections.

The limit is calculated based on:

- The actual capacity of the appliance.
- Value configured by the user as a percentage of the actual capacity. Default value is set to 150%.

For example, if the actual capacity of an appliance is 1000 operations/second at a given time and the default percentage is configured, the limit after which the appliance drops connections is 1500 (150% of 1000).

To configure the operation queue limit by using the CLI

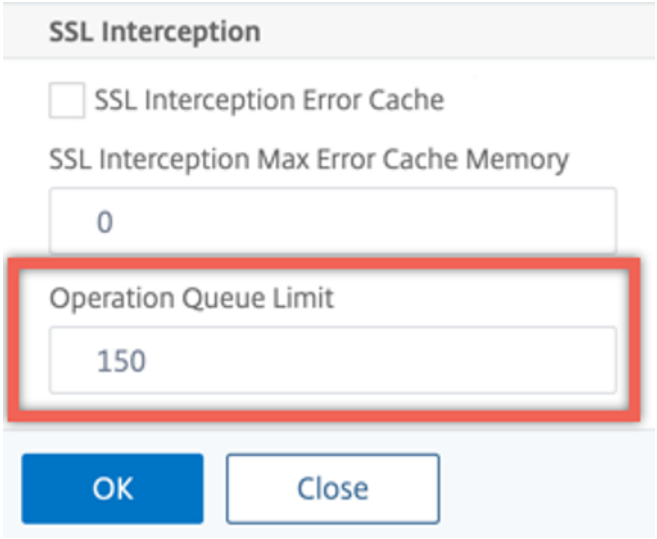
At the command prompt, type:

```
set ssl parameter -operationQueueLimit <positive_integer>
```

Operation Queue Limit - Limit in percentage of capacity of the crypto operations queue beyond which new SSL connections are not accepted until the queue is reduced. Default value: 150. Minimum value: 0. Maximum value: 10000.

To configure the operation queue limit by using the GUI

1. Navigate to **Traffic Management > SSL**.
2. In **Settings**, click **Change advanced SSL settings**.
3. Type a value in **Operation Queue Limit**. Default is 150.
4. Click **OK**.



Heterogeneous cluster deployments

From release 13.0 build 47.x, you can form a heterogeneous cluster deployment of Citrix ADC MPX appliances with a different number of packet engines by setting the SSL parameter “Heterogeneous SSL HW” to ENABLED. For example, to form a cluster of Cavium chip based appliances (MPX 14000 or similar) and Intel Coleto chip based appliances (MPX 15000 or similar), enable the SSL parameter

“Heterogeneous SSL HW.” To form a cluster of platforms using the same chip, keep the default value (DISABLED) for this parameter.

> **Notes:**
>
> The following features are not supported in a heterogeneous cluster:
>
> - VPX instances hosted on Citrix ADC SDX appliances.
> - SSLv3 protocol on SSL entities, such as virtual server, services, service group, and internal services.
> - Software crypto acceleration CPU threshold (using hardware and software to improve ECDSA and ECDHE cipher performance).

For more information about the platforms supported in a heterogeneous cluster, see https://docs.citrix.com/en-us/citrix-adc/13/clustering/support-for-heterogeneous-cluster.html.

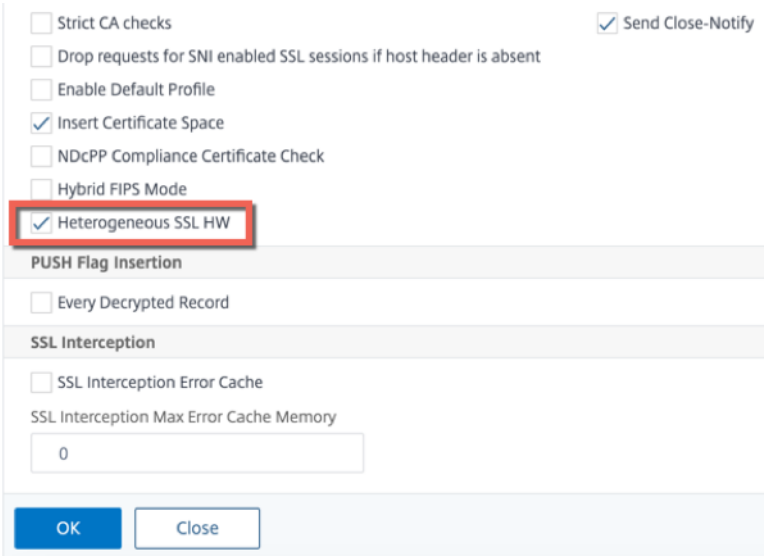
Enable a heterogeneous cluster using the CLI

At the command prompt, type:

```
set ssl parameter -heterogeneousSSLHW ENABLED
```

Enable a heterogeneous cluster using the GUI

1. Navigate to **Traffic Management > SSL** and, in the **Settings** group, select **Change advanced SSL settings**.
2. Select **Heterogeneous SSL HW**. Click **OK**.



PUSH flag based encryption trigger mechanism

The encryption trigger mechanism that is based on the PSH TCP flag now enables you to do the following:

- Merge consecutive packets in which the PSH flag is set into a single SSL record, or ignore the PSH flag.
- Perform timer-based encryption, in which the time-out value is set globally by using the `set ssl parameter -pushEncTriggerTimeout <positive_integer>` command.

Configure PUSH flag-based encryption by using the CLI

At the command prompt, type the following commands to configure PUSH flag-based encryption and verify the configuration:

```
set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]

show ssl vserver
<!--NeedCopy-->
```

Example:

```
set ssl vserver vserver1 -pushEncTrigger always

Done

sh ssl vserver vserver1

            Advanced SSL configuration for VServer vserver1:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED

                Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
```



```
21              HSTS Max-Age: 0
22              SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                    ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
23              Push Encryption Trigger: Always
24              Send Close-Notify: YES
25              Strict Sig-Digest Check: DISABLED
26              Zero RTT Early Data: DISABLED
27              DHE Key Exchange With PSK: NO
28              Tickets Per Authentication Context: 1
29              ECC Curve: P_256, P_384, P_224, P_521
30
31      1)      Cipher Name: DEFAULT
32              Description: Default cipher list with encryption strength
                   >= 128bit
33  Done
34  <!--NeedCopy-->
```

Configure PUSH flag-based encryption by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual servers** and open an SSL virtual server.
2. In the **SSL Parameters** section, from the **PUSH Encryption Trigger** list, select a value.

Support for TLS1.2 signature hash algorithm

The Citrix ADC appliance is completely TLS1.2 signature hash extension compliant.

In an SSL handshake, a client sends a list of supported signature hash algorithms. The client indicate to the server which signature hash algorithm pairs might be used in the SSL handshake messages (SKE and CCV) by using the “signature_algorithms” extension. The “extension_data” field of this extension contains a “supported_signature_algorithms” value in the Client Hello message. The SSL handshake proceeds if the server supports one of these signature hash algorithms. If the server does not support any of these algorithms, the connection is dropped.

Similarly, if the server requests a client certificate for client authentication, the Certificate Request message contains a “supported_signature_algorithms” value. The client certificate is selected based on this signature hash algorithm.

> **Note:**
>
> The Citrix ADC appliance acts as a server to a client and as a client to the back-end server.

The appliance supports only RSA-SHA1 and RSA-SHA256 on the front end, and RSA-MD5, RSA-SHA1, and RSA-SHA256 on the back end.

The MPX/SDX/VPX appliance supports the following signature hash combinations. On an SDX appliance, if an SSL chip is assigned to a VPX instance, the cipher support of an MPX appliance applies. Otherwise, the normal cipher support of a VPX instance applies.

- On a VPX instance and on an MPX/SDX appliance without N3 chips:
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
- On an MPX/SDX appliance with N3 chips:
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
  - ECDSA-SHA1
  - ECDSA-SHA224
  - ECDSA-SHA256
  - ECDSA-SHA384
  - ECDSA-SHA512

By default, all the signature hash algorithms are enabled. However, you can enable only a few signature hash algorithms by using the following command:

```
set ssl parameter -sigDigestType <sigDigestType>

Parameters

sigDigestType

Signature digest algorithms supported by the appliance. The platform
   determines the list of algorithms supported by default.

             On VPX: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384
                 RSA-

             SHA512

             On MPX with N3 cards: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-

```



```
15                SHA256 RSA-SHA384 RSA-SHA512 ECDSA-SHA1 ECDSA-SHA224
                      ECDSA-
16
17                SHA256 ECDSA-SHA384 ECDSA-SHA512
18
19                Other MPX Platforms: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-
                      SHA256 RSA-SHA384 RSA-
20
21                SHA512.
22
23      set ssl parameter -sigDigestType RSA-SHA224 RSA-SHA256 RSA-SHA384
            RSA-SHA512
24  <!--NeedCopy-->
```

Validate the peer certificate

According to RFC 5246, the peer certificate must be signed using one of the signature hash algorithms included in the Client Hello extension. You can use the `strictSigDigestCheck` parameter. Depending on the signature hash list sent by the client, if you enable `strictSigDigestCheck`, the appliance returns a certificate signed by one of the signature hash algorithms mentioned in the Client Hello extension. If the peer does not have a proper certificate, the connection is dropped. If this parameter is disabled, the signature hash is not checked in the peer certificate.

You can configure a strict signature digest check on an SSL virtual server and service. If you enable this parameter on an SSL virtual server, the server certificate sent by the server must be signed by one of the signature hash algorithms listed in the Client Hello extension. If client authentication is enabled, then the client certificate received by the server must be signed using one of the signature hash algorithms listed in the certificate request sent by the server.

If you enable this parameter on an SSL service, the server certificate received by the client must be signed by one of the signature hash algorithms listed in the Client Hello extension. The client certificate must be signed using one of the signature hash algorithms listed in the certificate request message.

If the default profile is enabled, you can use it to configure a strict signature digest check on an SSL virtual server, SSL service, and SSL profile.

Configure strict signature digest check on an SSL virtual server, service, or profile by using the CLI

At the command prompt, type:

```
1  set ssl vserver <vServerName> -strictSigDigestCheck ( ENABLED |
       DISABLED )
```



```
2
3 set ssl service <serviceName> -strictSigDigestCheck ( ENABLED |
     DISABLED )
4
5 set ssl profile <name>-strictSigDigestCheck ( ENABLED | DISABLED )
6
7 Parameters
8
9 strictSigDigestCheck
10
11              Check whether peer entity certificate is signed using one
                    of the signature-hash algorithms supported by the
                   Citrix ADC appliance.
12
13              Possible values: ENABLED, DISABLED
14
15              Default: DISABLED
16 <!--NeedCopy-->
```

Example:

```
1 set ssl vserver v1  -strictSigDigestCheck Enabled
2 set ssl service s1  -strictSigDigestCheck Enabled
3 set ssl profile p1  -strictSigDigestCheck Enabled
4 <!--NeedCopy-->
```

> **Important:**
>
> If DH, ECDHE, or ECDSA ciphers are configured on the appliance, the SKE message must be signed using one of the signature-hashes common to the client list and the list configured on the appliance. If there is no common signature hash, the connection is dropped.

TLSv1.3 protocol support as defined in RFC 8446

September 14, 2021

The Citrix ADC VPX and Citrix ADC MPX appliances now support the TLSv1.3 protocol, specified in RFC 8446.

> **Notes:**
>
> - From release 13.0 build 71.x and later, TLS1.3 hardware acceleration is supported on the following platforms:

  - MPX 5900
  - MPX/SDX 8900
  - MPX/SDX 15000
  - MPX/SDX 15000-50G
  - MPX/SDX 26000
  - MPX/SDX 26000-50S
  - MPX/SDX 26000-100G
  - Software-only support for the TLSv1.3 protocol is available on all other Citrix ADC MPX and SDX appliances except Citrix ADC FIPS appliances.
- TLSv1.3 is only supported with the enhanced profile. To enable the enhanced profile, see Enable the enhanced profile.
- To use TLS1.3, you must use a client that conforms to the RFC 8446 specification.

Supported Citrix ADC features

The following SSL features are supported:

1. TLSv1.3 cipher suites:
   - TLS1.3-AES256-GCM-SHA384 (0x1302)
   - TLS1.3_CHACHA20_POLY1305_SHA256 (0x1303)
   - TLS1.3-AES128_GCM-SHA256 (0x1301)
2. ECC curves for ephemeral Diffie-Hellman key exchange:
   - P_256
   - P_384
   - P_521
3. Abbreviated handshakes when ticket-based session resumption is enabled
4. 0-RTT early application data
5. Optional or mandatory certificate-based client authentication, with support for OCSP and CRL validation of client certificates
6. Server name extension: server certificate selection by using SNI
7. Application protocol negotiation (ALPN) by using the application_level_protocol_negotiation extension.
8. OCSP stapling
9. Log messages and AppFlow records are produced for TLSv1.3 handshakes.

10. Optional logging of TLS 1.3 traffic secrets by the `nstrace` packet capture utility.
11. Interoperability with TLS clients implementing RFC 8446. For example, Mozilla Firefox, Google Chrome, and OpenSSL.

Supported browsers

The following browser versions are supported and compatible with the Citrix ADC implementation oF TLS 1.3 protocol:

- Google Chrome - Version 72.0.3626.121 (Official Build) (64-bit)
- Mozilla Firefox - 65.0.2 (64-bit)
- Opera - Version:58.0.3135.79

Configuration

TLSv1.3 is disabled by default on an SSL profile.

Add an SSL profile by using the CLI

At the command prompt, type:

```
add ssl profile <tls13-profile-name>
<!--NeedCopy-->
```

Example:

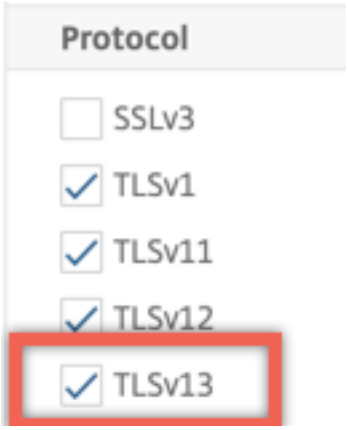
```
add ssl profile tls13profile

sh ssl profile tls13profile
1)  Name: tls13profile           (Front-End)
    SSLv3: DISABLED              TLSv1.0: ENABLED  TLSv1.1: ENABLED
       TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED                      Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED                            Refresh Count: 0
    Deny SSL Renegotiation                                ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
```



```
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:             1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag:            0x0 (Auto)
    SSL quantum size:                            8 kB
    Encryption trigger timeout           100 mS
    Encryption trigger packet count:               45
    Subject/Issuer Name Insertion Format: Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:  10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT Priority :1
    Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Add an SSL profile by using the GUI

1. Navigate to **System > Profiles**. Select **SSL Profiles**.
2. Click **Add** and specify a name for the profile.
3. In **Protocol**, select **TLSv13**.



4. Click **OK**.

Bind an SSL profile to an SSL virtual server by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName> -sslProfile <tls13-profile-name>
<!--NeedCopy-->
```

Example:

```
set ssl vserver ssl-vs -sslProfile tls13profile
```

Bind an SSL profile to an SSL virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and select an SSL virtual server.
2. In **Advanced Settings**, click **SSL Profile**.
3. Select the TLSv1.3 profile created earlier.
4. Click **OK**.
5. Click **Done**.

SSL profile parameters for TLSv1.3 protocol

1. Enable or disable TLS1.3 parameters in an SSL profile.

   tls13: State of TLSv1.3 protocol support for the SSL profile.

   Possible values: ENABLED, DISABLED

   Default value: DISABLED


```
set ssl profile tls13profile -tls13 enable
<!--NeedCopy-->
```

```
set ssl profile tls13profile -tls13 disable
<!--NeedCopy-->
```

2. Set number of session tickets issued.

   tls13SessionTicketsPerAuthContext: Number of tickets the SSL virtual server issues when TLS1.3 is negotiated, ticket-based resumption is enabled, and either (1) a handshake completes or (2) client authentication completes after the handshake.

   This value can be increased to enable clients to open multiple parallel connections using a fresh ticket for each connection.

   No tickets are sent if resumption is disabled.

   Default value: 1

   Minimum value: 1

   Maximum value: 10

```
set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 1

set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 10
<!--NeedCopy-->
```

3. Set DH key exchange

   dheKeyExchangeWithPsk: Specifies whether an SSL virtual server requires a DHE key exchange to occur when a preshared key is accepted during a TLS 1.3 session resumption handshake. A DHE key exchange ensures forward secrecy, even if ticket keys are compromised, at the expense of extra resources required to carry out the **DHE** key exchange.

   Available settings work as follows, if session ticket is enabled:

   YES: DHE key exchange is required when a pre-shared key is accepted, regardless of whether the client supports the key exchange. The handshake is aborted with a fatal alert, if the client does not support DHE key exchange when offering a pre-shared key.

   NO: DHE key exchange is performed when a pre-shared key is accepted, only if requested by the client.

   Possible values: YES, NO

   Default value: NO


```
set ssl profile tls13profile dheKeyExchangeWithPsk yes

set ssl profile tls13profile dheKeyExchangeWithPsk no
<!--NeedCopy-->
```

4. Enable or disable 0-RTT early data acceptance

   zeroRttEarlyData: State of TLS 1.3 early application data. Applicable settings work as follows:

   ENABLED: Early application data might be processed before the handshake is complete.
   DISABLED: Early application data is ignored.

   Possible values: ENABLED, DISABLED

   Default value: DISABLED

```
set ssl profile tls13profile -zeroRttEarlyData ENABLED

set ssl profile tls13profile -zeroRttEarlyData DISABLED
<!--NeedCopy-->
```

Default cipher group

The default cipher group includes TLS1.3 ciphers.

```
sh cipher DEFAULT
1) Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035

2)  Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
...
...
27) Cipher Name: TLS1.3-AES256-GCM-SHA384         Priority : 27
    Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(256) Mac=AEAD
         HexCode=0x1302

28) Cipher Name: TLS1.3_CHACHA20_POLY1305_SHA256     Priority : 28
    Description: TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256)
        Mac=AEAD   HexCode=0x1303

29) Cipher Name: TLS1.3-AES128_GCM-SHA256        Priority : 29
```



```
16     Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(128) Mac=AEAD
           HexCode=0x1301
17 Done
18 <!--NeedCopy-->
```

Limitations

- On the Citrix ADC MPX platform, TLSv1.3 processing is not offloaded to crypto hardware.
- TLSv1.3 is not supported on the back end.
- TLSv1.3 is not supported on a Citrix Secure Web Gateway appliance and on a Citrix ADC FIPS appliance.

Operational considerations

TLS 1.3 draft version compatibility note: A Citrix ADC appliance implements the RFC 8446 variant of the TLS 1.3 protocol (as opposed to earlier draft versions of the protocol). Use a TLS 1.3 client that supports RFC 8446 (or an interoperable draft # - 26, 27 or 28) to complete a TLS 1.3 handshake with a Citrix ADC appliance. Clients and servers that implement different draft versions of the TLS 1.3 protocol might not interoperate with each other.

Security restrictions

TLSv1.3 server operators must keep in mind the following security restrictions for backward compatibility outlined in RFC 8446. The default configuration on a NetScaler appliance is compliant with these restrictions. However, a NetScaler appliance does not enforce that these rules are adhered to.

- The security of RC4 cipher suites is considered insufficient as described in RFC7465. Implementations must not offer or negotiate RC4 cipher suites for any version of TLS.
- Old versions of TLS allowed the use of low strength ciphers. Ciphers with a strength less than 112 bits must not be offered or negotiated for any version of TLS.
- The security of SSL 3.0 [SSLv3] is considered insufficient as described in RFC7568, and must not be negotiated. Disable SSLv3 when TLSv1.3 is enabled (SSLv3 is disabled by default.)
- The security of SSL 2.0 [SSLv2] is considered insufficient as described in RFC6176, and must not be negotiated. Disable SSLv2 when TLS 1.3 is enabled (SSLv2 is disabled by default.)

> **Note:**
>
> For information about troubleshooting protocols that run over TLS1.3, see Decrypting TLS1.3 traffic from packet trace.

How-to articles

September 14, 2021

How-to articles are simple and easy to use articles with configuration steps for common deployments. Click a link to view the article.

Create a certificate signing request and use SSL certificates on a Citrix ADC appliance

Configure SSL action to forward client traffic

Configure SSL action to forward client traffic if a cipher is not supported on the ADC

Configure per directory client authentication

Configure support for Outlook web access

Configure SSL based header insertion

Configure SSL offloading with end-to-end encryption

Configure transparent SSL acceleration

Configure SSL acceleration with HTTP on the front end and SSL on the back end

Configure SSL offloading with other TCP protocols

Configure SSL bridging

Configure SSL monitoring when client authentication is enabled on the back-end service

Configure a secure content switching server

Configure an HTTPS virtual server to accept HTTP traffic

Configure graceful cleanup of SSL sessions

Configure support for HTTP strict transport security (HSTS)

Configure SSLv2 redirection

Configure synchronization of files in a high availability setup

Disable TLS 1.0 and TLS 1.1 on NSIP

Export certificates used on the Citrix ADC appliance as PFX file

SSL certificates

September 14, 2021

An SSL certificate, which is a part of any SSL transaction, is a digital data form (X509) that identifies a company (domain) or an individual. The certificate has a public key component that is visible to any client that wants to initiate a secure transaction with the server. The corresponding private key, which resides securely on the Citrix ADC appliance, is used to complete asymmetric key (or public key) encryption and decryption.

You can obtain an SSL certificate and key in either of the following ways:

- From an authorized certificate authority (CA), such as Verisign
- By generating a new SSL certificate and key on the Citrix ADC appliance

Alternately, you can use an existing SSL certificate on the appliance.

Certificates are categorized into four types by the Citrix ADC appliance:

- **Server certificates:** A server certificate authenticates the server's identity to the client. On the front-end, the ADC appliance acts as a server. You bind a server certificate and a private key to an SSL virtual server on the ADC appliance.
- **Client certificates:** A client certificate authenticates the client's identity to the server. On the back-end, the ADC appliance acts as a client. You bind a client certificate and private key to the SSL service or service group on the ADC appliance.
- **CA certificates:** CA certificates issue the end-user certificates (client and server certificates). A CA certificate can be a trusted root CA (self-signed by the certificate authority) or an intermediate CA (signed by a trusted root CA). Typically, CA certificates do not need private keys.
- **Unknown certificates:** All other certificates fall in this category.

Important: Citrix recommends that you use certificates obtained from authorized CAs, such as Verisign, for all your SSL transactions. Use certificates generated on the Citrix ADC appliance for testing purposes only, not in any live deployment.

- If while adding a certificate-key pair, you add a certificate file with the same name as an existing certificate file, the original certificate file is overwritten with no warning. This action might cause issues after the appliance is restarted because the original certificate file is no longer available in the `/nsconfig/ssl` directory.
- Removing any certificate or key files in a cluster environment restricts further configuration on the ADC appliance. Add the files back at the same location to make any configuration changes.

Note: You can use the ADM SSL dashboard for ease of SSL certificate management and set notifications for certificates that are unused or soon to expire. For more information, see SSL certificate management.

Create a certificate

September 14, 2021

A certificate authority (CA) is an entity that issues digital certificates for use in public key cryptography. Applications, such as web browsers, that conduct SSL transactions trust certificates issued or signed by a Certificate Authority. These applications maintain a list of the CAs that they trust. If any of the trusted CAs sign the certificate being used for the secure transaction, the application proceeds with the transaction.

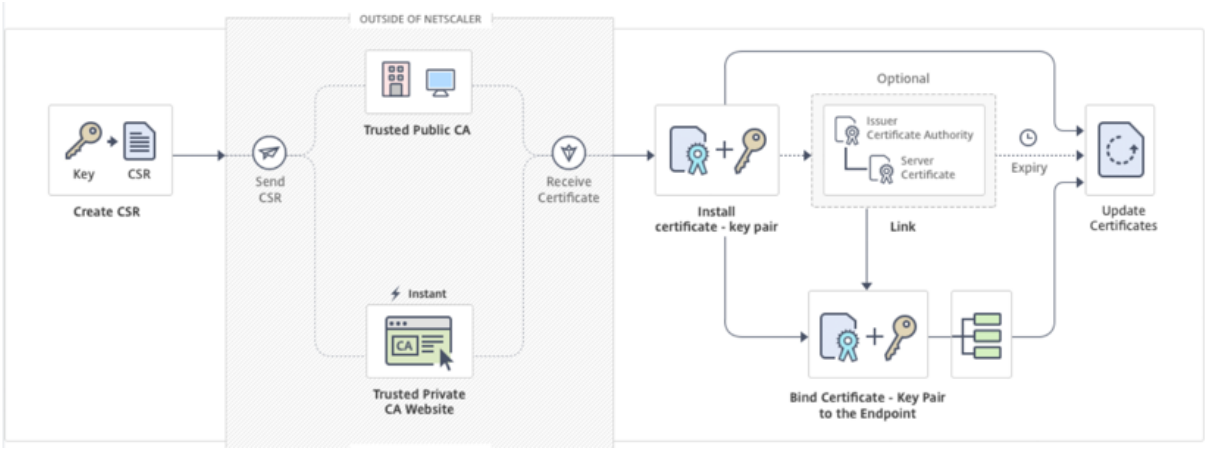
Caution: Citrix recommends that you use certificates obtained from authorized CAs, such as Verisign, for all your SSL transactions. Use certificates generated on the Citrix ADC appliance for testing purposes only, not in any live deployment.

To import an existing certificate and key, see Import a Certificate.

Perform the following steps to create a certificate and bind it to an SSL virtual server. The only special characters allowed in the file names are underscore and dot.

- Create a private key.
- Create a certificate signing request (CSR).
- Submit the CSR to a certificate authority.
- Create a certificate-key pair.
- Bind the certificate-key pair to an SSL virtual server

The following diagram illustrates the workflow.



Video link to How do I create and install a new certificate.

Create a private key

> **Notes:**
>
> - From release 12.1 build 49.x, you can use the AES256 algorithm with PEM key format to encrypt a private key on the appliance. AES with 256-bit key is more mathematically efficient and secure compared to the 56-bit key of the Data Encryption Standard (DES).
> - From release 12.1 build 50.x, you can create an RSA key in PKCS#8 format.

The private key is the most important part of a digital certificate. By definition, this key is not to be shared with anyone and must be kept securely on the Citrix ADC appliance. Any data encrypted with the public key can be decrypted only by using the private key.

The certificate that you receive from the CA is valid only with the private key that was used to create the CSR. The key is required for adding the certificate to the Citrix ADC appliance.

The appliance supports only the RSA encryption algorithms for creating private keys. You can submit either type of private key to the certificate authority (CA). The certificate that you receive from the CA is valid only with the private key that was used to create the CSR. The key is required for adding the certificate to the Citrix ADC appliance.

Important:

> - Be sure to limit access to your private key. Anyone who has access to your private key can decrypt your SSL data.
> - The length of the SSL key name allowed includes the length of the absolute path name if the path is included in the key name.

All SSL certificates and keys are stored in the `/nsconfig/ssl` folder on the appliance. For added security, you can use the DES or triple DES (3DES) algorithm to encrypt the private key stored on the appliance.

Create an RSA private key by using the CLI

At the command prompt, type:

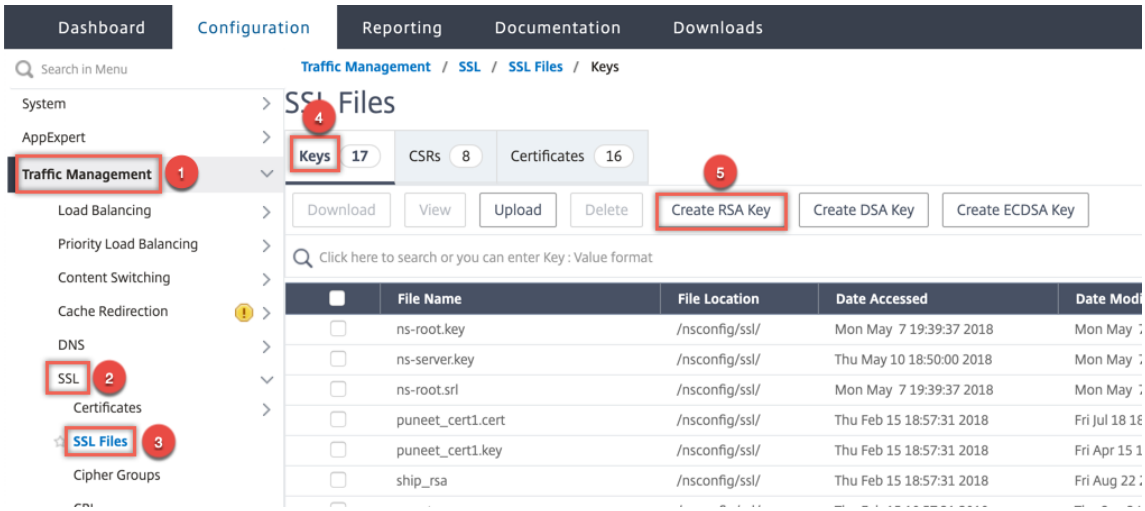
```
create ssl rsakey <keyFile> <bits> [-exponent ( 3 | F4 )] [-keyform (
    DER | PEM )] [-des | -des3 | -aes256] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

Example:

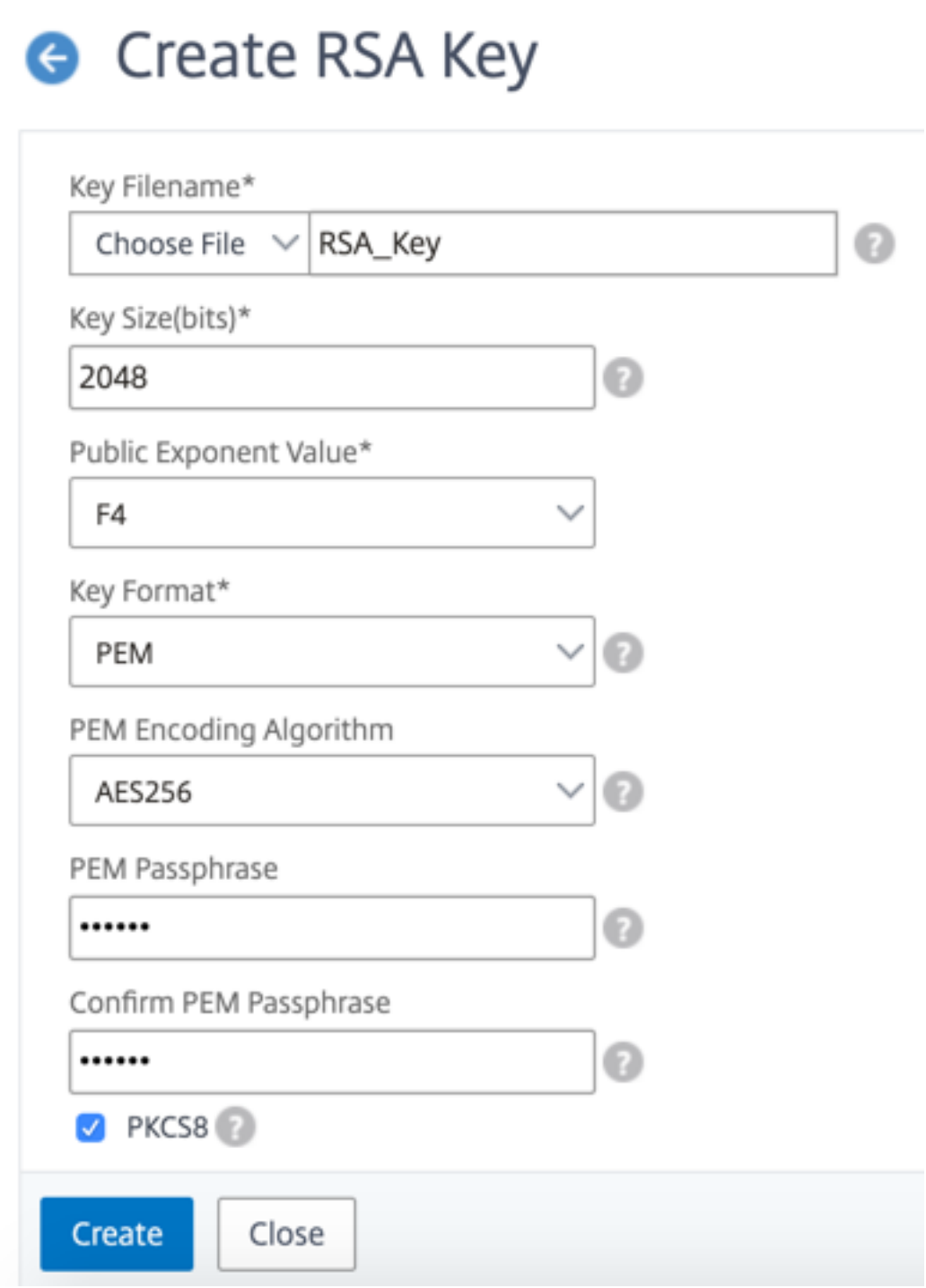
```
create rsakey testkey 2048 -aes256 -password 123456 -pkcs8
<!--NeedCopy-->
```


Create an RSA private key by using the GUI

1. Navigate to **Traffic Management > SSL > SSL Files**.
2. In the **Keys** tab, select **Create RSA Key**.



3. Enter values for the following parameters and click **Create**.
   - **Key Filename** - Name for and, optionally, path to the RSA key file. /nsconfig/ssl/ is the default path.
   - **Key Size** - Size, in bits, of the RSA key. Can range from 512 bits to 4096 bits.
   - **Public Exponent Value** - Public exponent for the RSA key. The exponent is part of the cipher algorithm and is required for creating the RSA key.
   - **Key Format** - The format in which the RSA key file is stored on the appliance.
   - **PEM Encoding Algorithm** - Encrypt the generated RSA key by using the AES 256, DES, or Triple-DES (DES3) algorithm. By default, private keys are unencrypted.
   - **PEM Passphrase** - If the private key is encrypted, enter a pass phrase for the key.



Select an AES256 encoding algorithm in an RSA key by using the GUI

1. Navigate to **Traffic Management > SSL > SSL Files > Create RSA Key**.
2. In **Key Format**, select **PEM**.

3. In **PEM Encoding Algorithm**, select **AES256**.
4. Select **PKCS8**.

Create a certificate signing request by using the CLI

At the command prompt, type:

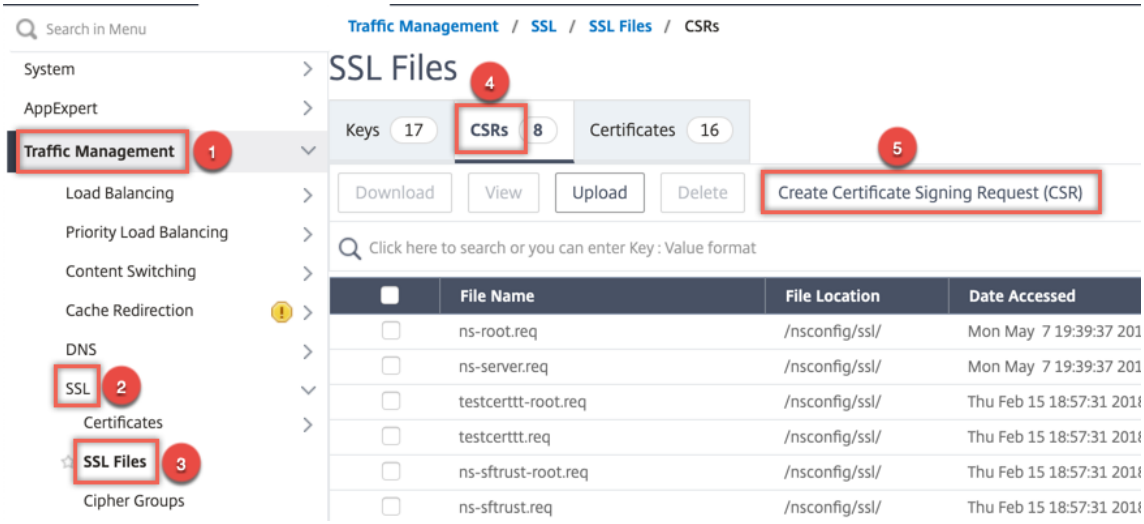
```
create ssl certreq <reqFile> -keyFile <input_filename> | -fipsKeyName <string>) [-keyForm (DER | PEM) {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string> -organizationUnitName <string> -localityName <string> -commonName <string> -emailAddress <string> {
 -challengePassword  }
  -companyName <string> -digestMethod ( SHA1 | SHA256 )
<!--NeedCopy-->
```

Example:

```
create ssl certreq priv_csr_sha256 -keyfile priv_2048_2 -keyform PEM -countryName IN -stateName Karnataka -localityName Bangalore -organizationName Citrix -organizationUnitName NS -digestMethod SHA256
<!--NeedCopy-->
```

Create a certificate signing request by using the GUI

1. Navigate to **Traffic Management > SSL**.
2. In **SSL** Certificate, click **Create Certificate Signing Request (CSR)**.



3. In **Digest** Method, select **SHA256**.

See Create a CSR for more information.

Support for subject alternative name in a certificate signing request

The subject alternative name (SAN) field in a certificate allows you to associate multiple values, such as domain names and IP addresses, with a single certificate. In other words, you can secure multiple domains, such as www.example.com, www.example1.com, www.example2.com, with a single certificate.

Some browsers, such as Google Chrome, no longer support a common name in a certificate signing request (CSR). They enforce SAN in all publicly trusted certificates.

The Citrix ADC appliance supports adding SAN values when creating a CSR. You can send a CSR with a SAN entry to a certificate authority to obtain a signed certificate with that SAN entry. When the appliance receives a request, it checks for a matching domain name in the SAN entries in the server certificate. If a match is found, it sends the certificate to the client and completes the SSL handshake. You can use the CLI or the GUI to create a CSR with SAN values.

> **Note:** The Citrix ADC appliance processes only DNS based SAN values.

Create a CSR with the subject alternative name by using the CLI

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-subjectAltName <string>] [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string>
     [-organizationUnitName <string>] [-localityName <string>] [-
    commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

Parameters:

subjectAltName: The subject alternative name (SAN) is an extension to X.509 that allows various values to be associated with a security certificate using a subjectAltName field. These values are called “Subject Alternative Names” (SAN). Names include:

1. IP addresses (Prefix with “IP:” Example: IP:198.51.10.5 IP:192.0.2.100)
2. DNS names (Prefix with “DNS:” Example: DNS:www.example.com DNS:www.example.org DNS:www.example.net)

On the command line, enter values within quotation marks. Separate two values with a space. Quotation marks are not required in the GUI.
Maximum Length: 127

Example:

```
create certReq test1.csr -keyFile test1.ky -countryName IN -stateName
    Kar -organizationName citrix -commonName ctx.com -subjectAltName "
    DNS:*.example.com DNS:www.example.org DNS:www.example.net"
<!--NeedCopy-->
```

> **Note:**
>
> On a FIPS appliance, you must replace the key file name with the FIPS key name if you create the FIPS key directly on the appliance.
>
> ```
> create certReq <csrname> -fipsKeyName fipskey.ky -countryName IN -
>     stateName Kar -organizationName citrix -commonName ctx.com -
>     subjectAltName "DNS:www.example.com DNS:www.example.org DNS:www.
>     example.net"
> <!--NeedCopy-->
> ```

Create a CSR by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **SSL Files**.
2. In the **CSR** tab, click **Create Certificate Signing Request (CSR)**.
3. Enter the values and click **Create**.

Limitations

To use SAN when creating an SSL certificate, you must explicitly specify the SAN values. The values are not read automatically from the CSR file.

Submit the CSR to the Certificate Authority

Most certificate authorities (CA) accept certificate submissions by email. The CA returns a valid certificate to the email address from which you submit the CSR.

The CSR is stored in the `/nsconfig/ssl` folder.

Generate a test certificate

> **Note:**
>
> To generate a server test certificate, see Generating a Server Test Certificate.

The Citrix ADC appliance has a built-in CA tools suite that you can use to create self-signed certificates for testing purposes.

Caution: Because the Citrix ADC appliance signs these certificates, and not an actual CA, you must not use them in a production environment. If you attempt to use a self-signed certificate in a production environment, users receive a "certificate invalid" warning each time the virtual server is accessed.

The appliance supports creation of the following types of certificates

- Root-CA certificates
- Intermediate-CA certificates
- End-user certificates
  - server certificates
  - client certificates

Before generating a certificate, create a private key and use that to create a certificate signing request (CSR) on the appliance. Then, instead of sending the CSR out to a CA, use the Citrix ADC CA Tools to generate a certificate.

Create a certificate by using a wizard

1. Navigate to **Traffic Management > SSL**.
2. In the details pane, under **Getting Started**, select the wizard for the type of certificate that you want to create.
3. Follow the instructions on the screen.

Create a root-CA certificate by using the CLI

At the command prompt, type the following command:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
   input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
<!--NeedCopy-->
```

In the following example, csreq1 is the CSR and rsa1 is the private key that was created earlier.

Example:

```
create ssl cert cert1 csreq1 ROOT_CERT -keyFile rsa1 -keyForm PEM -days
    365

  Done
```



```
4 <!--NeedCopy-->
```

Create an intermediate-CA certificate by using the CLI

```
1 create ssl cert <certFile> <reqFile> <certType> [-keyFile <
     input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
      [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
      DER | PEM )] [-CAkey <input_filename>] [-CAkeyForm ( DER | PEM )]
     [-CAserial <output_filename>]
2 <!--NeedCopy-->
```

In the following example, csr1 is the CSR created earlier. Cert1 and rsakey1 are the certificate and corresponding key of the self-signed (root-CA) certificate, and pvtkey1 is the private key of the intermediate-CA certificate.

Example:

```
1 create ssl cert certsy csr1 INTM_CERT -CAcert cert1 -CAkey rsakey1 -
     CAserial 23
2 Done
3
4 create ssl rsakey pvtkey1 2048 -exponent F4 -keyform PEM
5 Done
6 <!--NeedCopy-->
```

Create a root-CA certificate by using the GUI

Navigate to **Traffic Management > SSL** and, in the Getting Started group, select **Root-CA Certificate Wizard**, and configure a root CA certificate.

Create an intermediate-CA certificate by using the GUI

Navigate to **Traffic Management > SSL** and, in the Getting Started group, select **Intermediate-CA Certificate Wizard**, and configure an intermediate CA certificate.

Create an end-user certificate

An end-user certificate can be a client certificate or a server certificate. To create a test end-user certificate, specify the Intermediate CA certificate or the self-signed root-CA certificate.

> **Note:** To create an end-user certificate for production use, specify a trusted CA certificate and send the CSR to a certificate authority (CA).

Create a test end-user certificate by using the command line interface

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days<positive_integer>]
    [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
    DER | PEM )] [-CAkey<input_filename>] [-CAkeyForm ( DER | PEM )] [-
    CAserial <output_filename>]
<!--NeedCopy-->
```

If there is no intermediate certificate, use the certificate (cert1) and private key (rsakey1) values of the root-CA certificate in `CAcert` and `CAkey`.

Example:

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23

Done
<!--NeedCopy-->
```

If there is an intermediate certificate, use the certificate (`certsy`) and private key (pvtkey1) values of the intermediate certificate in `CAcert` and `CAkey`.

Example:

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert certsy -CAkey pvtkey1 -
    CAserial 23

Done
<!--NeedCopy-->
```

Create a self-signed SAN Certificate using OpenSSL

To create a self-signed SAN certificate with multiple subject alternate names, perform the following steps:

1. Create an OpenSSL configuration file on your local computer by editing the fields related as per the company requirements.

   Note: In the following example, the configuration file is “req.conf”.


```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = US
ST = VA
L = SomeCity
O = MyCompany
OU = MyDivision
CN = www.company.com
[v3_req]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[alt_names]
DNS.1 = www.company.net
DNS.2 = company.com
DNS.3 = company.net
<!--NeedCopy-->
```

2. Upload the file to the /nsconfig/ssl directory on the Citrix ADC appliance.
3. Log on to Citrix ADC CLI as `nsroot` user and switch to the shell prompt.
4. Run the following command to create the certificate:

```
cd /nsconfig/ssl
openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout cert.
   pem -out cert.pem -config req.conf -extensions 'v3_req'
<!--NeedCopy-->
```

5. Run the following command to verify the certificate:

```
openssl x509 -in cert.pem -noout  - text
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
ed:90:c5:f0:61:78:25:ab
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Validity
Not Before: Nov 6 22:21:38 2012 GMT
```



```
Not After : Nov 6 22:21:38 2014 GMT
Subject: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
    www.company.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus (2048 bit):
…
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Key Usage:
Key Encipherment, Data Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 Subject Alternative Name:
DNS:www.company.net, DNS:company.com, DNS:company.net
Signature Algorithm: md5WithRSAEncryption …
<!--NeedCopy-->
```

Install, link, and update certificates

September 14, 2021

To install a certificate, see Add or update a certificate-key pair.

Link certificates

Many server certificates are signed by multiple hierarchical Certificate Authorities (CA), which means that the certificates form a chain like the following:

```
Root CA
|
+----------- Intermediate CA
|
+----------------------- End-Entity (Server) Certificate
```

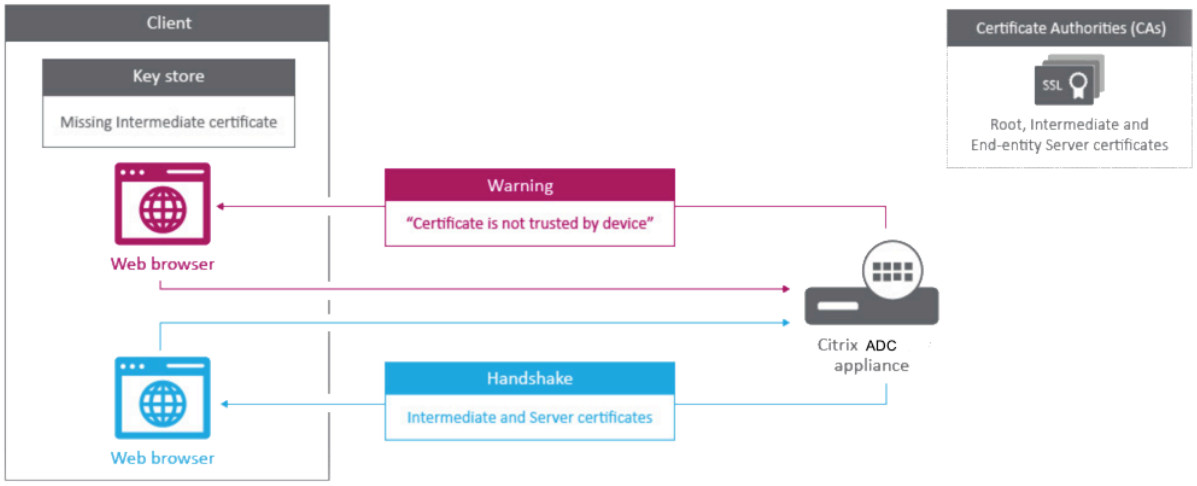
Sometimes, the Intermediate CA is split into a primary and secondary intermediate CA certificate. Then the certificates form a chain like the following:

```
Root CA
|
+------ Primary Intermediate CA
|+-------------- Secondary Intermediate CA
|
+----------------------- End-Entity (Server) Certificate
```


Client machines usually contain the root CA certificate in their local certificate store, but not one or more intermediate CA certificates. The ADC appliance must send one or more intermediate CA certificates to the clients.

Note: The appliance must not send the root CA certificate to the client. The Public Key Infrastructure (PKI) trust relationship model requires root CA certificates to be installed on clients through an out-of-band method. For example, the certificates are included with the operating system or web browser. The client ignores a root CA certificate sent by the appliance.

Sometimes, an intermediate CA that standard web browsers do not recognize as a trusted CA, issues the server certificate. In this case, one or more CA certificates must be sent to the client with the server's own certificate. Otherwise, the browser terminates the SSL session because it fails to authenticate the server certificate.



Video link to How do I link an intermediate authority certificate.

Refer to the following sections to add the server and intermediate certificates:

- Manual certificate linking
- Automated certificate linking
- Create a chain of certificates

Manual certificate linking

Note: This feature is not supported on the Citrix ADC FIPS platform and in a cluster setup.

Instead of adding and linking individual certificates, you can now group a server certificate and up to nine intermediate certificates in a single file. You can specify the file's name when adding a certificate-key pair. Before you do so, make sure that the following prerequisites are met.

- The certificates in the file are in the following order:
  - Server certificate (must be the first certificate in the file)

  - Optionally, a server key
  - Intermediate certificate 1 (ic1)
  - Intermediate certificate 2 (ic2)
  - Intermediate certificate 3 (ic3), and so on
    Note: Intermediate certificate files are created for each intermediate certificate with the name "<certificatebundlename>.pem_ic<
    n>" where n is between 1 and 9. For example, bundle.pem_ic1, where **bundle** is the name of the certificate set and ic1 is the first intermediate certificate in the set.
- Bundle option is selected.
- No more than nine intermediate certificates are present in the file.

The file is parsed and the server certificate, intermediate certificates, and server key (if present) are identified. First, the server certificate and key are added. Then, the intermediate certificates are added, in the order in which they were added to the file, and linked accordingly.

An error is reported if any of the following conditions exist:

- A certificate file for one of the intermediate certificates exists on the appliance.
- The key is placed before the server certificate in the file.
- An intermediate certificate is placed before the server certificate.
- Intermediate certificates are not in placed in the file in the same order as they are created.
- No certificates are present in the file.
- A certificate is not in the proper PEM format.
- The number of intermediate certificates in the file exceeds nine.

Add a certificate set by using the CLI

At the command prompt, type the following commands to create a certificate set and verify the configuration:

```
1 add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES
      | NO)
2
3 show ssl
4
5 show ssl certlink
6 <!--NeedCopy-->
```

In the following example, the certificate set (bundle.pem) contains the following files:

Server certificate (bundle) linked to bundle_ic1

First intermediate certificate (bundle_ic1) linked to bundle_ic2

Second intermediate certificate (bundle_ic2) linked to bundle_ic3

Third intermediate certificate (bundle_ic3)

```
add ssl certKey bundletest -cert bundle9.pem -key bundle9.pem -bundle
   yes

sh ssl certkey

1)      Name: ns-server-certificate
        Cert Path: ns-server.cert
        Key Path: ns-server.key
        Format: PEM
        Status: Valid,   Days to expiration:5733
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
           Internal,CN=default OULLFT
        Validity
                Not Before: Apr 21 15:56:16 2016 GMT
                Not After : Mar  3 06:30:56 2032 GMT
        Subject:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
           Internal,CN=default OULLFT
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

2)      Name: servercert
        Cert Path: complete/server/server_rsa_1024.pem
        Key Path: complete/server/server_rsa_1024.ky
        Format: PEM
        Status: Valid,   Days to expiration:7150
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 1F
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:  C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix
        Validity
                Not Before: Sep  2 09:54:07 2008 GMT
                Not After : Jan 19 09:54:07 2036 GMT
        Subject:  C=IN,ST=KAR,O=Citrix Pvt Ltd,CN=Citrix
        Public Key Algorithm: rsaEncryption
```



```
        Public Key size: 1024

3)      Name: bundletest
        Cert Path: bundle9.pem
        Key Path: bundle9.pem
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Validity
                Not Before: Nov 28 06:43:11 2014 GMT
                Not After : Nov 25 06:43:11 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=Server9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

4)      Name: bundletest_ic1
        Cert Path: bundle9.pem_ic1
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Validity
                Not Before: Nov 28 06:42:56 2014 GMT
                Not After : Nov 25 06:42:56 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

5)      Name: bundletest_ic2
        Cert Path: bundle9.pem_ic2
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
```



```
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Validity
                Not Before: Nov 28 06:42:55 2014 GMT
                Not After : Nov 25 06:42:55 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

6)      Name: bundletest_ic3
        Cert Path: bundle9.pem_ic3
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA6
        Validity
                Not Before: Nov 28 06:42:53 2014 GMT
                Not After : Nov 25 06:42:53 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

7)      Name: bundletest_ic4
        Cert Path: bundle9.pem_ic4
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA5
        Validity
                Not Before: Nov 28 06:42:51 2014 GMT
                Not After : Nov 25 06:42:51 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA6
```



```
131         Public Key Algorithm: rsaEncryption
132         Public Key size: 2048
133
134 8)      Name: bundletest_ic5
135         Cert Path: bundle9.pem_ic5
136         Format: PEM
137         Status: Valid,   Days to expiration:3078
138         Certificate Expiry Monitor: ENABLED
139         Expiry Notification period: 30 days
140         Certificate Type: Intermediate CA
141         Version: 3
142         Serial Number: 01
143         Signature Algorithm: sha256WithRSAEncryption
144         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA4
145         Validity
146                 Not Before: Nov 28 06:42:50 2014 GMT
147                 Not After : Nov 25 06:42:50 2024 GMT
148         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA5
149         Public Key Algorithm: rsaEncryption
150         Public Key size: 2048
151
152 9)      Name: bundletest_ic6
153         Cert Path: bundle9.pem_ic6
154         Format: PEM
155         Status: Valid,   Days to expiration:3078
156         Certificate Expiry Monitor: ENABLED
157         Expiry Notification period: 30 days
158         Certificate Type: Intermediate CA
159         Version: 3
160         Serial Number: 01
161         Signature Algorithm: sha256WithRSAEncryption
162         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA3
163         Validity
164                 Not Before: Nov 28 06:42:48 2014 GMT
165                 Not After : Nov 25 06:42:48 2024 GMT
166         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA4
167         Public Key Algorithm: rsaEncryption
168         Public Key size: 2048
169
170 10)     Name: bundletest_ic7
171         Cert Path: bundle9.pem_ic7
172         Format: PEM
173         Status: Valid,   Days to expiration:3078
174         Certificate Expiry Monitor: ENABLED
175         Expiry Notification period: 30 days
```



```
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA2
        Validity
                Not Before: Nov 28 06:42:46 2014 GMT
                Not After : Nov 25 06:42:46 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA3
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

11)     Name: bundletest_ic8
        Cert Path: bundle9.pem_ic8
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA1
        Validity
                Not Before: Nov 28 06:42:45 2014 GMT
                Not After : Nov 25 06:42:45 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA2
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

12)     Name: bundletest_ic9
        Cert Path: bundle9.pem_ic9
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=RootCA4096
        Validity
                Not Before: Nov 28 06:42:43 2014 GMT
                Not After : Nov 25 06:42:43 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA1
```



```
221         Public Key Algorithm: rsaEncryption
222         Public Key size: 2048
223 Done
224
225 sh ssl certlink
226
227 1)      Cert Name: bundletest    CA Cert Name: bundletest_ic1
228 2)      Cert Name: bundletest_ic1        CA Cert Name: bundletest_ic2
229 3)      Cert Name: bundletest_ic2        CA Cert Name: bundletest_ic3
230 4)      Cert Name: bundletest_ic3        CA Cert Name: bundletest_ic4
231 5)      Cert Name: bundletest_ic4        CA Cert Name: bundletest_ic5
232 6)      Cert Name: bundletest_ic5        CA Cert Name: bundletest_ic6
233 7)      Cert Name: bundletest_ic6        CA Cert Name: bundletest_ic7
234 8)      Cert Name: bundletest_ic7        CA Cert Name: bundletest_ic8
235 9)      Cert Name: bundletest_ic8        CA Cert Name: bundletest_ic9
236 Done
237 <!--NeedCopy-->
```

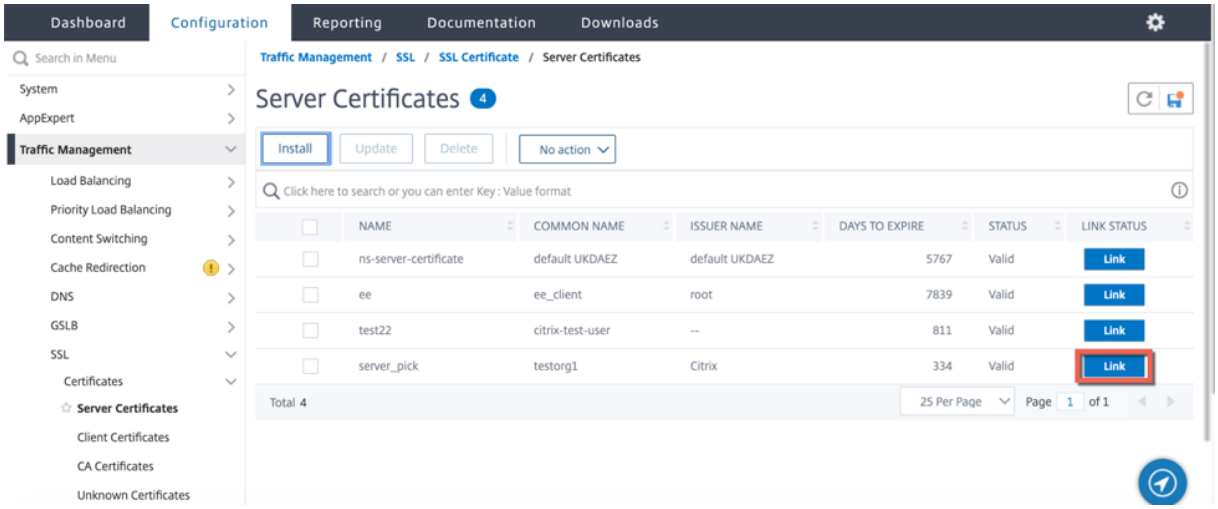
Add a certificate set by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Certificates** > **CA Certificates**.
2. In the details pane, click **Install**.
3. In the **Install Certificate** dialog box, type the details, such as the certificate and key file name, and then select **Certificate Bundle**.
4. Click **Install**, and then click **Close**.

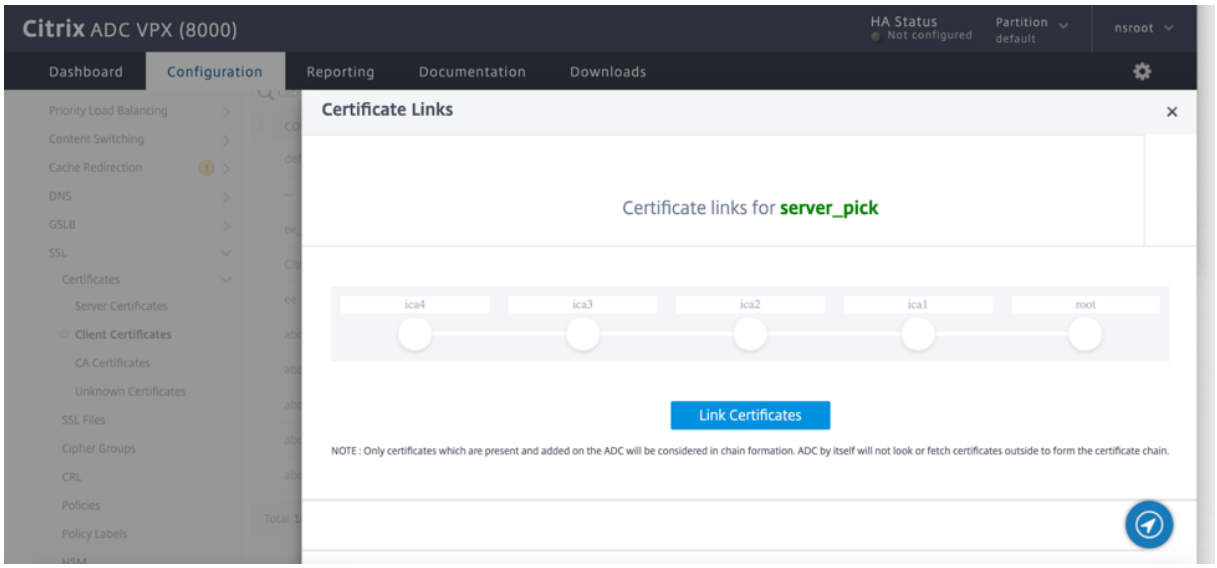
Automated certificate linking

Note: This feature is available from release 13.0 build 47.x.

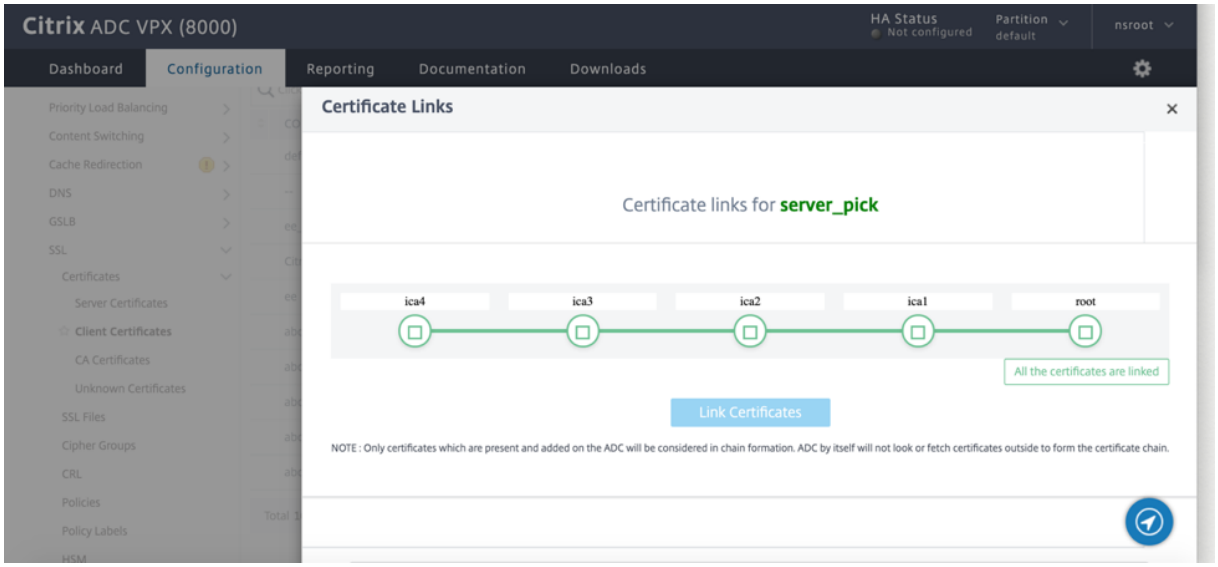
You no longer have to manually link a certificate to its issuer all the way to the root certificate. If the intermediate CA certificates and the root certificate are present on the appliance, you can click the **Link** button in the end user certificate.



The potential chain appears.



Click **Link Certificate** to link all the certificates.



Create a chain of certificates

Instead of using a set of certificates (a single file), you can create a chain of certificates. The chain links the server certificate to its issuer (the intermediate CA). This approach requires that the intermediate CA certificate file is installed on the ADC appliance, and the client application must trust one of the certificates in the chain. For example, link Cert-Intermediate-A to Cert-Intermediate-B, where Cert-Intermediate-B is linked to Cert-Intermediate-C, which is a certificate trusted by the client application.

Note: The appliance supports sending a maximum of 10 certificates in the chain of certificates sent to the client (one server certificate and nine CA certificates).

Create a certificate chain by using the CLI

At the command prompt, type the following commands to create a certificate chain and verify the configuration. (Repeat the first command for each new link in the chain.)

```
link ssl certkey <certKeyName> <linkCertKeyName>
show ssl certlink
<!--NeedCopy-->
```

Example:

```
link ssl certkey siteAcertkey CAcertkey
Done

show ssl certlink

linked certificate:
```



```
        1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
Done
<!--NeedCopy-->
```

Create a certificate chain by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Certificates**.
2. Select a server certificate, and in the **Action** list, select **Link**, and specify a CA certificate name.

Update an existing server certificate

To change an existing server certificate manually, you must perform the following steps:

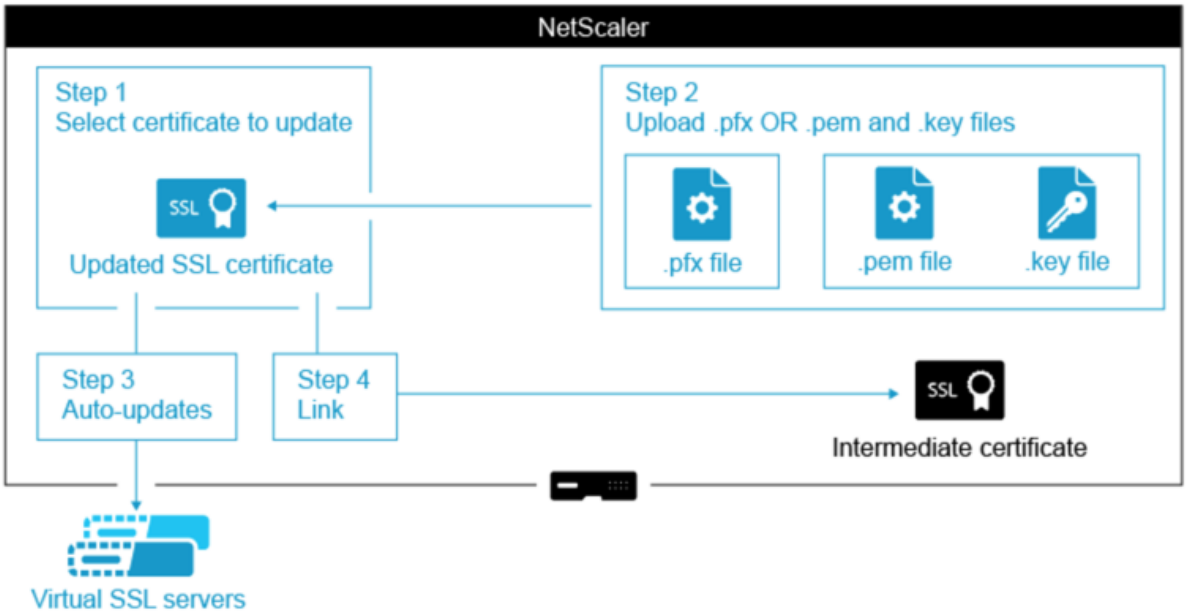
1. Unbind the old certificate from the virtual server.
2. Remove the certificate from the appliance.
3. Add the new certificate to the appliance.
4. Bind the new certificate to the virtual server.

To reduce downtime when replacing a certificate-key pair, you can update an existing certificate. If you want to replace a certificate with a certificate that was issued to a different domain, you must disable domain checks before updating the certificate.

To receive notifications about certificates due to expire, you can enable the expiry monitor.

When you remove or unbind a certificate from a configured SSL virtual server or service, the virtual server or service becomes inactive. They are active after a new valid certificate is bound to them. To reduce downtime, you can use the update feature to replace a certificate-key pair that is bound to an SSL virtual server or an SSL service.

Overview diagram of how to update an SSL certificate on the Citrix ADC appliance.



Video link to How do I update an existing certificate.

Update an existing certificate-key pair by using the CLI

At the command prompt, type the following commands to update an existing certificate-key pair and verify the configuration:

```
update ssl certkey <certkeyName> -cert <string> -key <string>

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

Example:

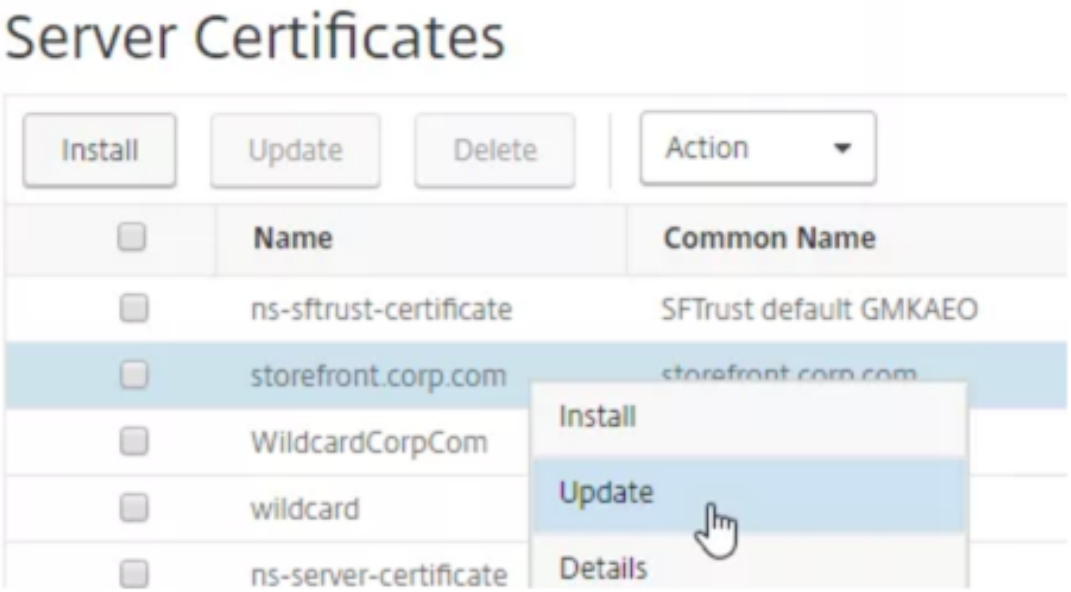
```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
   nsconfig/ssl/pkey.pem

Done

show ssl certkey siteAcertkey

Name: siteAcertkey       Status: Valid
          Version: 3
          Serial Number: 02
          Signature Algorithm: md5WithRSAEncryption
          Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech
          Validity
               Not Before: Nov 11 14:58:18 2001 GMT
```



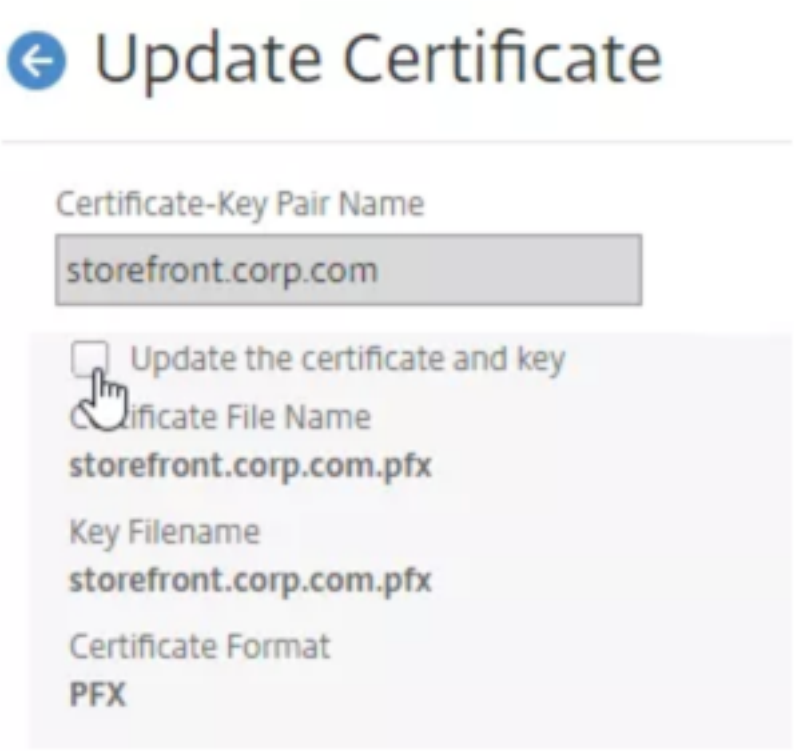
```
                Not After: Aug 7 14:58:18 2004 GMT
            Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security
            Public Key Algorithm: rsaEncryption
            Public Key size: 2048
Done
<!--NeedCopy-->
```

Update an existing certificate-key pair by using the GUI

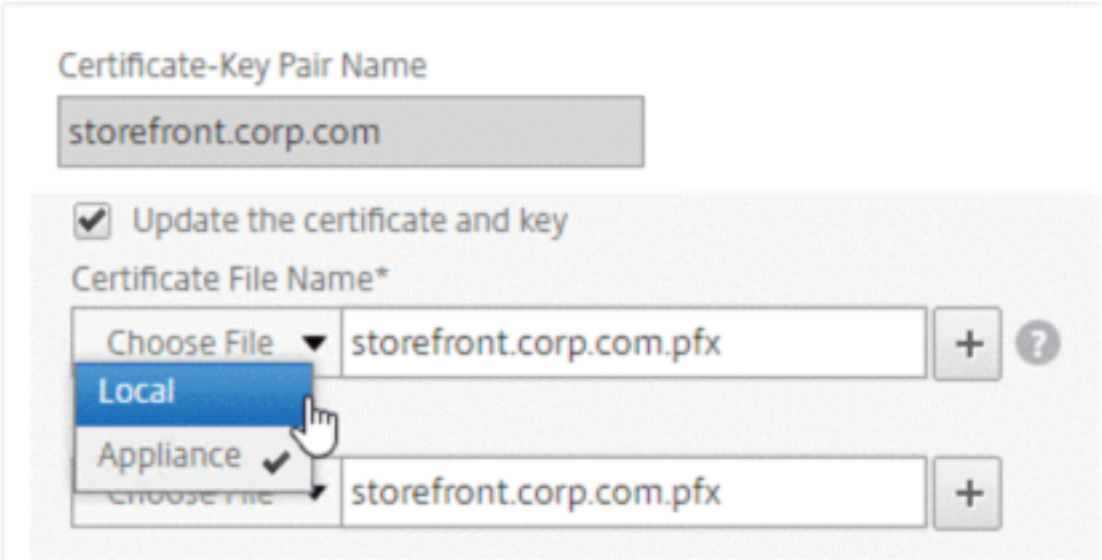
1. Navigate to **Traffic Management** > **SSL** > **Certificates** > **Server Certificates**.
2. Select the certificate that you want to update, and click **Update**.



3. Select **Update the certificate and key**.



4. In **Certificate File Name**, click **Choose File** > **Local**, and browse to the updated .pfx file or certificate PEM file.

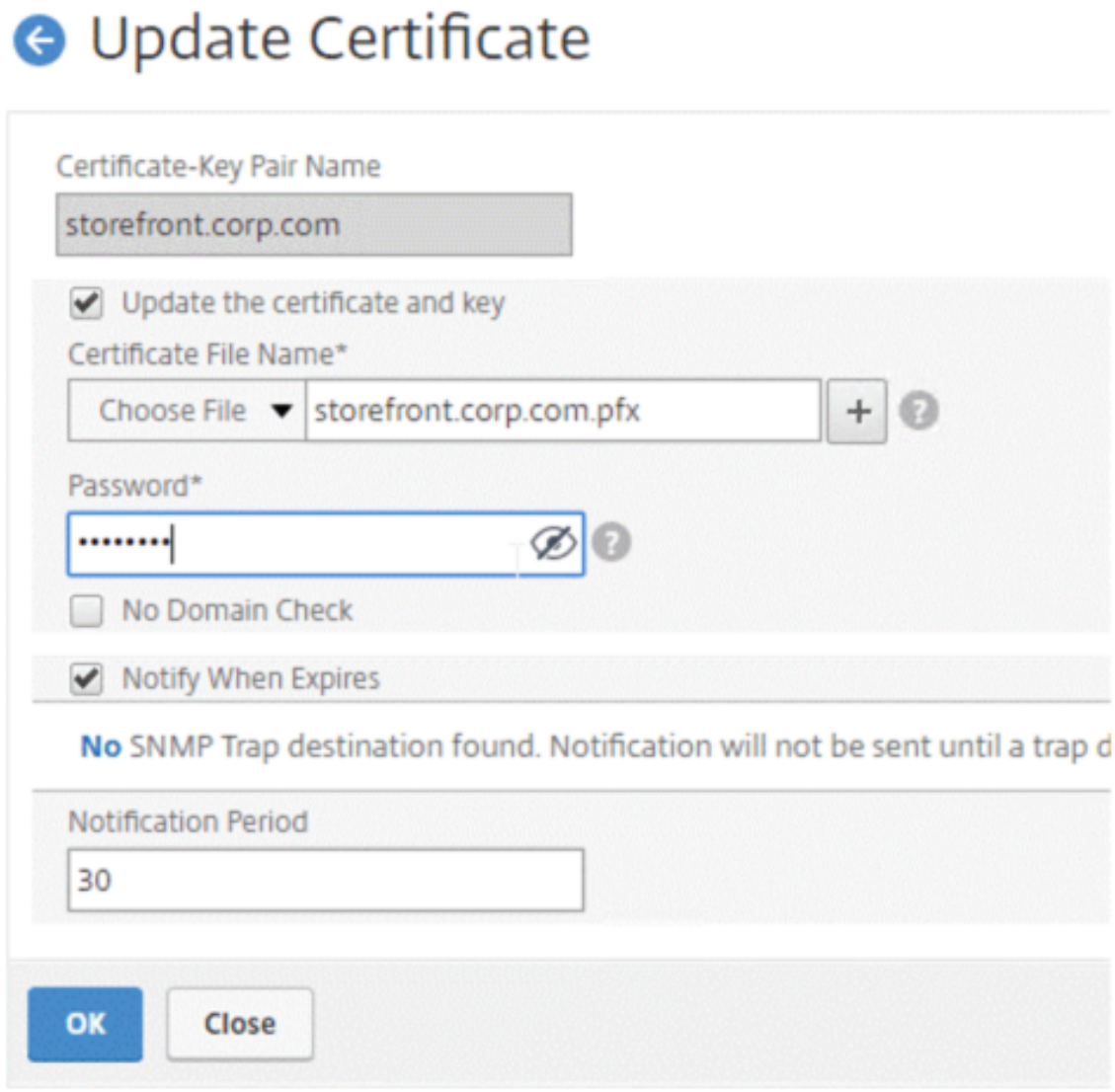


   - If you upload a .pfx file, you are prompted to specify the .pfx file password.
   - If you upload a certificate pem file, you must also upload a certificate key file. If the key is encrypted, you must specify the encryption password.

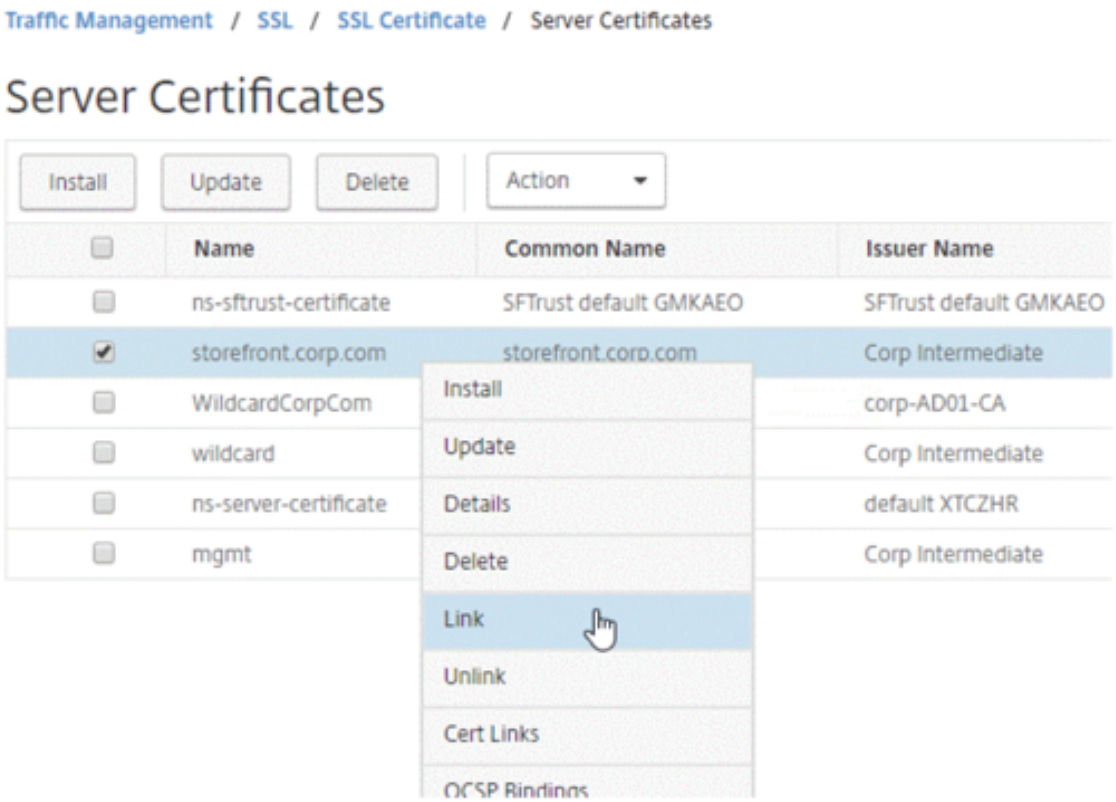
5. If the common name of the new certificate does not match the old certificate, then select **No**

Domain Check.

6. Click **OK**. All the SSL virtual servers to which this certificate is bound are automatically updated.

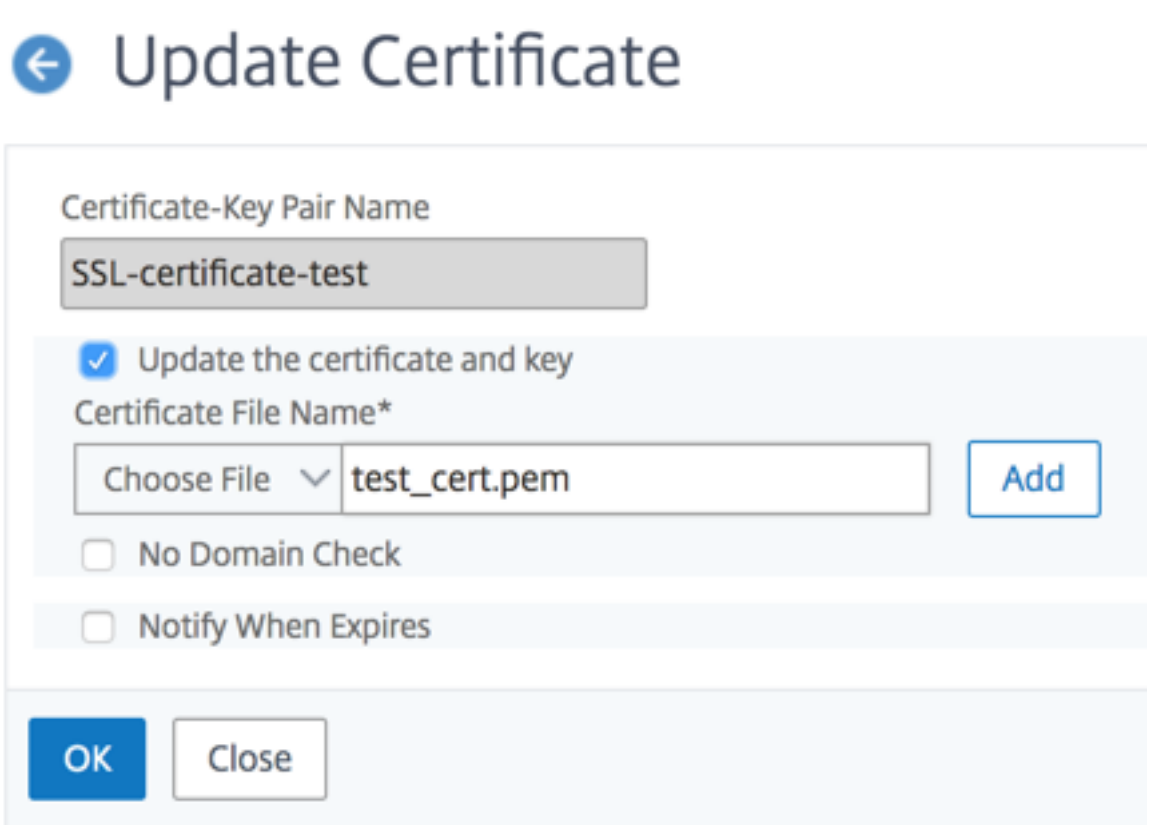


7. After replacing the certificate, you might have to update the certificate link to a new intermediate certificate. For more information about updating an intermediate certificate without breaking the links, see Update an intermediate certificate without breaking the links.
   - Right-click the updated certificate, and click **Cert Links**, to see if it is linked to an intermediate certificate.
   - If the certificate is not linked, then right-click the updated certificate, and click **Link** to link it to an intermediate certificate. If you don't see an option to link, then you must first have to install a new intermediate certificate on the appliance under the **CA Certificates** node.



Update an existing CA certificate

The steps to update an existing CA certificate are the same as updating an existing server certificate. The only difference is that you do not need a key in the case of CA certificates.



Disable domain checks

When an SSL certificate is replaced on the appliance, the domain name mentioned on the new certificate must match the domain name of the certificate being replaced. For example, if you have a certificate issued to abc.com, and you are updating it with a certificate issued to def.com, the certificate update fails.

However, if you want the server that has been hosting a particular domain to host a new domain, disable the domain check before updating its certificate.

Disable the domain check for a certificate by using the CLI

At the command prompt, type the following commands to disable the domain check and verify the configuration:

```
update ssl certKey <certkeyName> -noDomainCheck

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

Example:


```
update ssl certKey sv -noDomainCheck

Done

show ssl certkey sv

Name: sv
Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
Format: PEM
Status: Valid,   Days to expiration:9349
Certificate Expiry Monitor: DISABLED
Done
<!--NeedCopy-->
```

Disable the domain check for a certificate by using the GUI

1. Navigate to **Traffic Management > SSL > Certificates**, select a certificate, and click **Update**.
2. Select **No Domain Check**.

Replace the default certificate of an ADC appliance with a trusted CA certificate that matches the host name of the appliance

The following procedure assumes that the default certificate (`ns-server-certificate`) is bound to the internal services.

1. Navigate to **Traffic Management > SSL > SSL Certificates > Create Certificate Request**.
2. In common name, type `test.citrixadc.com`.
3. Submit the CSR to a trusted certificate authority.
4. After receiving the certificate from the trusted CA, copy the file to the `/nsconfig/ssl` directory.
5. Navigate to **Traffic Management > SSL > Certificates > Server Certificates**.
6. Select the default server certificate (`ns-server-certificate`) and click **Update**.
7. In the **Update Certificate** dialog box, in **Certificate File Name**, browse to the certificate received from the CA after signing.
8. In the **Key File Name** field, specify the default private key file name (`ns-server.key`).
9. Select **No Domain Check**.
10. Click **OK**.

Enable the expiry monitor

An SSL certificate is valid for a specific period. A typical deployment includes multiple virtual servers that process SSL transactions, and the certificates bound to them can expire at different times. An expiry monitor configured on the appliance creates entries in the appliance's syslog and ns audit logs when a certificate configured is due to expire.

If you want to create SNMP alerts for certificate expiration, you must configure them separately.

Enable an expiry monitor for a certificate by using the CLI

At the command prompt, type the following commands to enable an expiry monitor for a certificate and verify the configuration:

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED ) [-
    notificationPeriod <positive_integer>]]

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

Example:

```
set ssl certKey sv -expiryMonitor ENABLED  - notificationPeriod 60
Done
<!--NeedCopy-->
```

Enable an expiry monitor for a certificate by using the GUI

1. Navigate to **Traffic Management > SSL > Certificates**, select a certificate, and click **Update**.
2. Select **Notify When Expires**, and optionally specify a notification period.

Update an intermediate certificate without breaking the links

You can now update an intermediate certificate without breaking any existing links. The 'AuthorityKeyIdentifier' extension, in the linked certificate issued by the certificate to be replaced, must not contain an authority certificate serial number ('authorityCertSerialNumber') field. If the 'AuthorityKeyIdentifier' extension contains a serial number field, then the certificate serial numbers of the old and new certificate must be the same. You can update any number of certificates in the link, one at a time, if the preceding condition is met. Previously, the links broke if an intermediate certificate was updated.

For example, there are four certificates: `CertA`, `CertB`, `CertC`, and `CertD`. Certificate `CertA` is the issuer for `CertB`, `CertB` is the issuer for `CertC`, and so on. If you want to replace an intermediate

certificate CertB with CertB_new, without breaking the link, the following condition must be met:

The certificate serial number of CertB must match the certificate serial number of CertB_new if both of the following conditions are met:

- The AuthorityKeyIdentifier extension is present in CertC.
- This extension contains a serial number field.

If the common name in a certificate changes, while updating the certificate specify nodomaincheck.

In the preceding example, to change “www.example.com” in CertD to “*.example.com,” select the ‘No Domain Check” parameter.

Update the certificate by using the CLI

At the command prompt, type:

```
update ssl certKey <certkeyName> -cert <string> [-password] -key <
   string>  [-noDomainCheck]
<!--NeedCopy-->
```

Example:

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
   nsconfig/ssl/pkey.pem -noDomainCheck
<!--NeedCopy-->
```

Display a certificate chain

A certificate contains the name of the issuing authority and the subject to whom the certificate is issued. To validate a certificate, you must look at the issuer of that certificate and confirm if you trust the issuer. If you do not trust the issuer, you must see who issued the issuer certificate. Go up the chain until you reach the root CA certificate or an issuer that you trust.

As part of the SSL handshake, when a client requests a certificate, the appliance presents a certificate and the chain of issuer certificates present on the appliance. An administrator can view the certificate chain for the certificates present on the appliance and install any missing certificates.

View the certificate chain for the certificates present on the appliance by using the CLI

At the command prompt, type:

```
show ssl certchain <cert_name>
<!--NeedCopy-->
```


Examples

There are 3 certificates: c1, c2, and c3. Certificate c3 is the root CA certificate and signs c2, and c2 signs c1. The following examples illustrate the output of the `show ssl certchain c1` command in different scenarios.

Scenario 1:

Certificate c2 is linked to c1, and c3 is linked to c2.

Certificate c3 is a root CA certificate.

If you run the following command, the certificate links up to the root CA certificate are displayed.

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate name: c2              linked; not a root
       certificate
    2) Certificate name: c3              linked; root certificate
Done
<!--NeedCopy-->
```

Scenario 2:

Certificate c2 is linked to c1.

Certificate c2 is not a root CA certificate.

If you run the following command, the information that certificate c3 is a root CA certificate but is not linked to c2 is displayed.

```
show ssl certchain c1

Certificate chain  details of certificate name c1 are:
    1) Certificate Name: c2              linked; not a root
       certificate
    2) Certificate Name: c3              not linked; root certificate
    Done
<!--NeedCopy-->
```

Scenario 3:

Certificate c1, c2, and c3 are not linked but are present on the appliance.

If you run the following command, information about all the certificates starting with the issuer of certificate c1 is displayed. It is also specified that the certificates are not linked.

```
show ssl certchain c1
```



```
2
3 Certificate chain details of certificate name c1 are:
4     1) Certificate Name: c2               not linked; not a root
         certificate
5     2) Certificate Name: c3               not linked; root certificate
6     Done
7 <!--NeedCopy-->
```

Scenario 4:

Certificate c2 is linked to c1.

Certificate c3 is not present on the appliance.

If you run the following command, information about the certificate linked to c1 is displayed. You are prompted to add a certificate with the subject name specified in c2. In this case, the user is asked to add the root CA certificate c3.

```
1 show ssl certchain c1
2
3 Certificate chain details of certificate name c1 are:
4     1) Certificate Name: c2               linked; not a root
         certificate
5     2) Certificate Name: /C=IN/ST=ka/O=netscaler/CN=test
6        Action: Add a certificate with this subject name.
7 Done
8 <!--NeedCopy-->
```

Scenario 5:

A certificate is not linked to certificate c1 and the issuer certificate of c1 is not present on the appliance.

If you run the following command, you are prompted to add a certificate with the subject name in certificate c1.

```
1 sh ssl certchain c1
2
3 Certificate chain details of certificate name c1 are:
4     1) Certificate Name: /ST=KA/C=IN
5        Action: Add a certificate with this subject name.
6 <!--NeedCopy-->
```

Generate a server test certificate

September 14, 2021

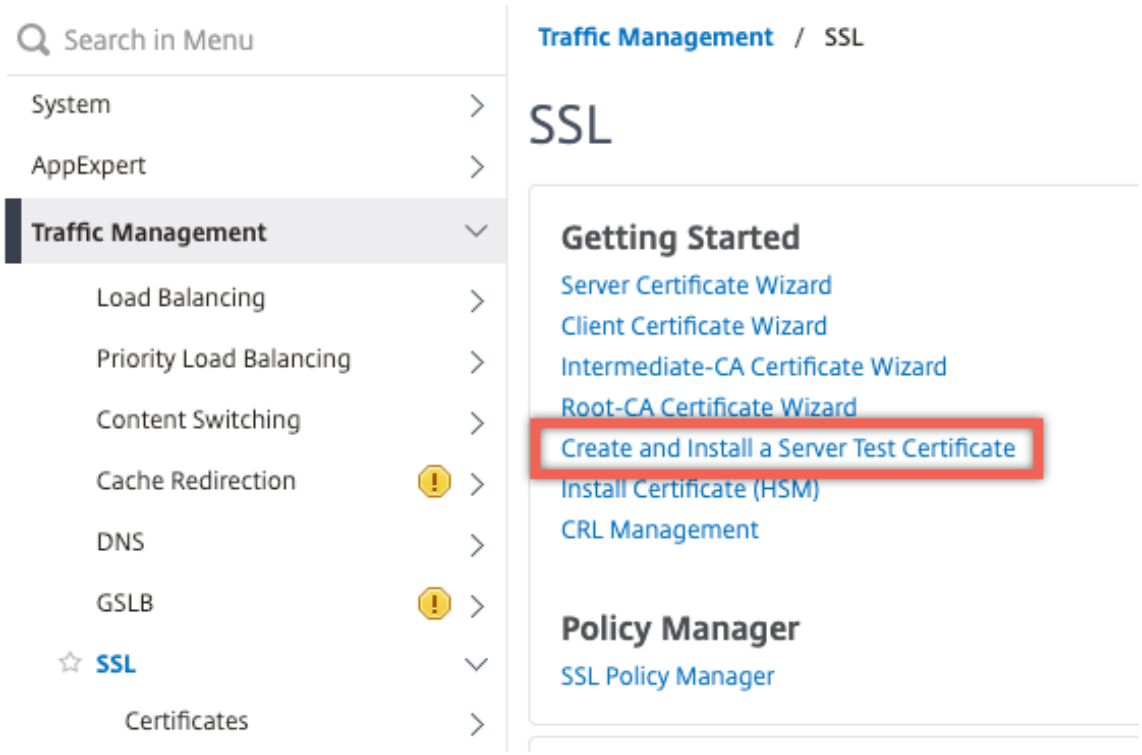
The Citrix ADC appliance allows you to create a test certificate for server authentication by using a GUI wizard in the configuration utility. A server certificate is used to authenticate and identify a server in an SSL handshake. Typically, a trusted CA issues a server certificate. The server sends the certificate to a client who uses it to authenticate the server.

For issuing a server test certificate, the appliance operates as a certificate authority. This certificate can be bound to an SSL virtual server for authentication in an SSL handshake with a client. This certificate is for testing purposes only. Do not use in a production environment.

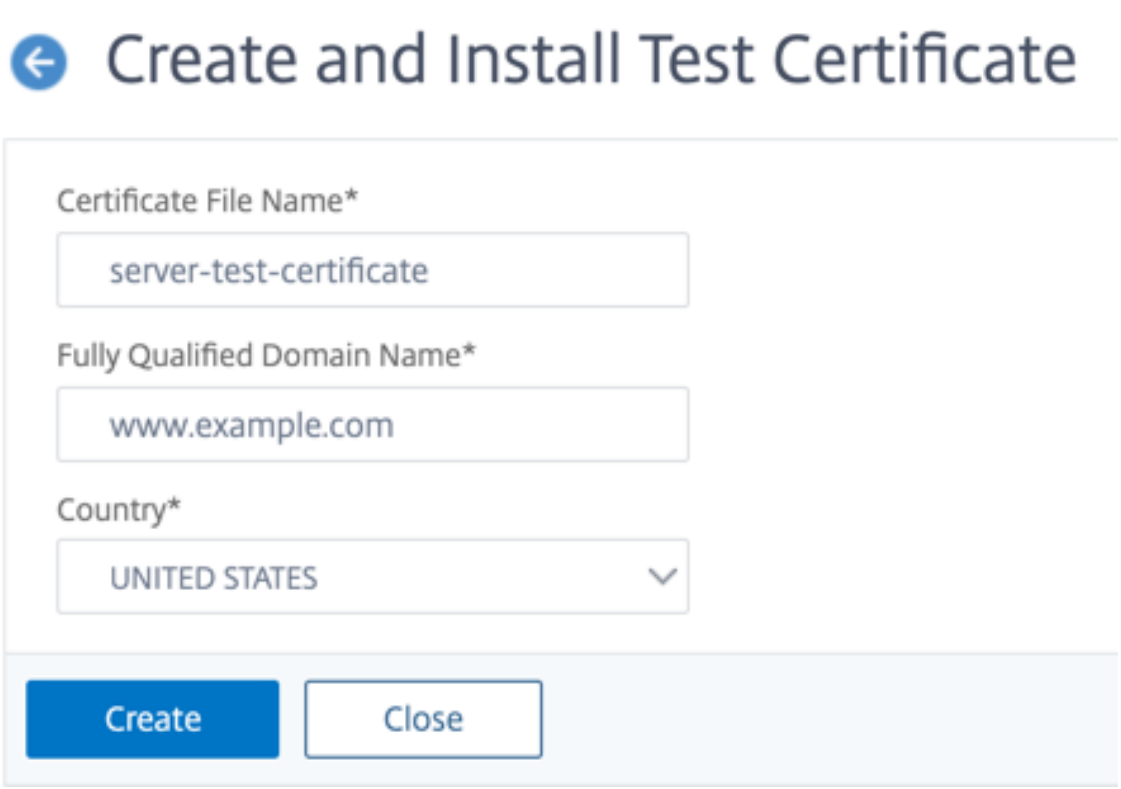
You can install the server test certificate on any virtual server that uses the SSL or the SSL_TCP protocol.

Generate a server test certificate by using the GUI

1. Navigate to **Traffic Management > SSL** and, in the **SSL Certificates group**, select **Create and Install a Server Test Certificate**.



2. Enter details for the parameters and click **Create**.



Import and convert SSL files

September 14, 2021

You can now import SSL resources, such as certificates, private keys, CRLs, and DH keys, from remote hosts even if FTP access to these hosts is not available. This feature is especially helpful in environments where shell access to the remote host is restricted. Default folders are created in /nsconfig/ssl as follows:

- For certificate files: /nsconfig/ssl/certfile
- For private keys: the /nsconfig/ssl/keyfile
- For CRLs: /var/netscaler/ssl/crlfile
- For DH keys: /nsconfig/ssl/dhfile

Imports from both HTTP and HTTPS servers are supported. However, the import fails if the file is on an HTTPS server that requires client certificate authentication for access.

> **Note:**
>
> The import command is not stored in the configuration (ns.conf) file, because reimporting the

> file after a restart might cause an error.

Import a certificate file

You can use the CLI and GUI to import a file (resource) from a remote host.

Import a certificate file from a remote host by using the CLI

At the command prompt, type:

```
import ssl certFile [<name>] [<src>]
<!--NeedCopy-->
```

Example:

```
import ssl certfile my-certfile http://www.example.com/file_1
<!--NeedCopy-->
```

```
show ssl certfile
     Name : my-certfile
     URL : http://www.example.com/file_1
<!--NeedCopy-->
```

To remove a certificate file, use the `rm ssl certFile` command, which accepts only the 'name' argument.

Import a key file from a remote host by using the CLI

At the command prompt, type:

```
import ssl keyFile [<name>] [<src>]
<!--NeedCopy-->
```

Example:

```
import ssl keyfile my-keyfile http://www.example.com/key_file
<!--NeedCopy-->
```

```
show ssl keyfile
     Name : my-keyfile
     URL : http://www.example.com/key_file
<!--NeedCopy-->
```

To remove a key file, use the `rm ssl keyFile` command, which accepts only the 'name' argument.

Import a CRL file from a remote host by using the CLI

At the command prompt, type:

```
import ssl crlFile [<name>] [<src>]
<!--NeedCopy-->
```

To remove a CRL file, use the `rm ssl crlFile` command, which accepts only the <name> argument.

Example:

```
import ssl crlfile my-crlfile http://www.example.com/crl_file

show ssl crlfile

    Name : my-crlfile
    URL : http://www.example.com/crl_file
<!--NeedCopy-->
```

Import a DH file from a remote host by using the CLI

At the command prompt, type:

```
import ssl dhFile [<name>] [<src>]
<!--NeedCopy-->
```

Example:

```
import ssl dhfile my-dhfile http://www.example.com/dh_file
show ssl dhfile
     Name : my-dhfile
     URL : http://www.example.com/dh_file
<!--NeedCopy-->
```

To remove a DH file, use the `rm ssl dhFile` command, which accepts only the <name> argument.

Import an SSL resource by using the GUI

Navigate to **Traffic Management > SSL > Imports**, and then select the appropriate tab.

Import PKCS#8 and PKCS#12 certificates

If you want to use certificates and keys that you already have on other secure servers or applications in your network, you can export them, and then import them to the Citrix ADC appliance. You might have to convert exported certificates and keys before you can import them to the Citrix ADC appliance.

For the details of how to export certificates from secure servers or applications in your network, see the documentation of the server or application from which you want to export.

> **Note:**
>
> For installation on the Citrix ADC appliance, key and certificate names cannot contain spaces or special characters other than those characters supported by the UNIX file system. Follow the appropriate naming convention when you save the exported key and certificate.

A certificate and private key pair is commonly sent in the PKCS#12 format. The appliance supports PEM and DER formats for certificates and keys. To convert PKCS#12 to PEM or DER, or PEM or DER to PKCS#12, see the "Convert SSL certificates for import or export" section later in this page.

The Citrix ADC appliance does not support PEM keys in PKCS#8 format. However, you can convert these keys to a supported format by using the OpenSSL interface, which you can access from the CLI or the configuration utility. Before you convert the key, you need to verify that the private key is in PKCS#8 format. Keys in PKCS#8 format typically start with the following text:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----



leuSSZQZKgrgUQ==



-----END ENCRYPTED PRIVATE KEY-----
<!--NeedCopy-->
```

Open the OpenSSL interface from the CLI

1. Open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance by using the administrator credentials.
3. At the command prompt, type shell.
4. At the shell prompt type `openssl`.

Open the OpenSSL interface from the GUI

Navigate to **Traffic Management > SSL** and, in the Tools group, select **OpenSSL interface**.

Convert a non-supported PKCS#8 key format to an encrypted supported key format by using the OpenSSL interface

At the OpenSSL prompt, type one of the following commands, depending on whether the non-supported key format is of type RSA or ECDSA:

```
1 OpenSSL>rsa- in <PKCS#8 Key Filename> -des3 -out <encrypted Key
     Filename>
2
3 OpenSSL>ec -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename
     >
4 <!--NeedCopy-->
```

Parameters for converting an unsupported key format to a supported key format

- **PKCS#8 Key Filename:** The input file name of the incompatible PKCS#8 private key.
- **encrypted Key Filename:** The output file name of the compatible encrypted private key in PEM format.
- **unencrypted Key Filename:** The output file name of the compatible unencrypted private key in PEM format.

Convert SSL certificates for import or export

A Citrix ADC appliance supports the PEM and DER formats for SSL certificates. Other applications, such as client browsers and some external secure servers, require various public key cryptography standard (PKCS) formats. The appliance can convert the PKCS#12 format to PEM or DER format for importing a certificate to the appliance, and can convert PEM or DER to PKCS#12 for exporting a certificate. For more security, conversion of a file for import can include encryption of the private key with the DES or DES3 algorithm.

> **Note:**
>
> If you use the GUI to import a PKCS#12 certificate, and the password contains a dollar sign ($), back quote (‘), or escape () character, the import might fail. If it does, the ERROR: Invalid password message appears. If you must use a special character in the password, be sure to prefix it with an escape character () unless all imports are performed by using the CLI.

Convert the format of a certificate by using the CLI

At the command prompt, type the following command:


```
convert ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-
    des | -des3] [-export [-certFile <inputFilename>] [-keyFile <
    inputFilename>]]
<!--NeedCopy-->
```

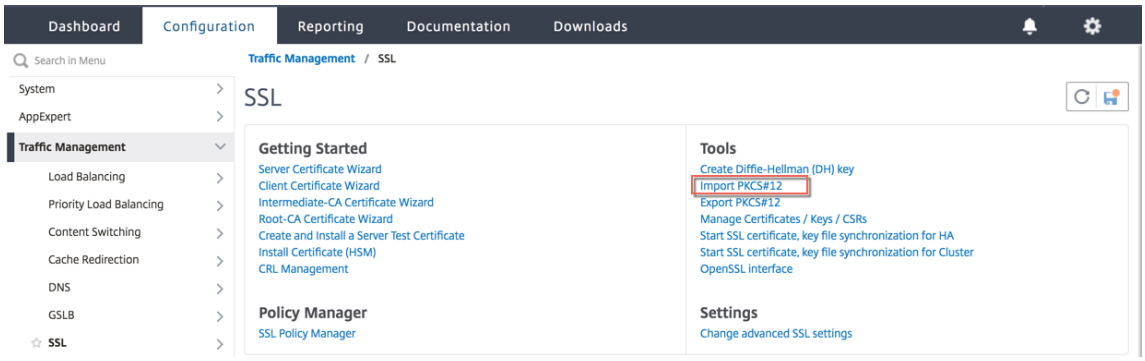
During the operation, you are prompted to enter an import password or an export password. For an encrypted file, you are also prompted to enter a passphrase.

Example:

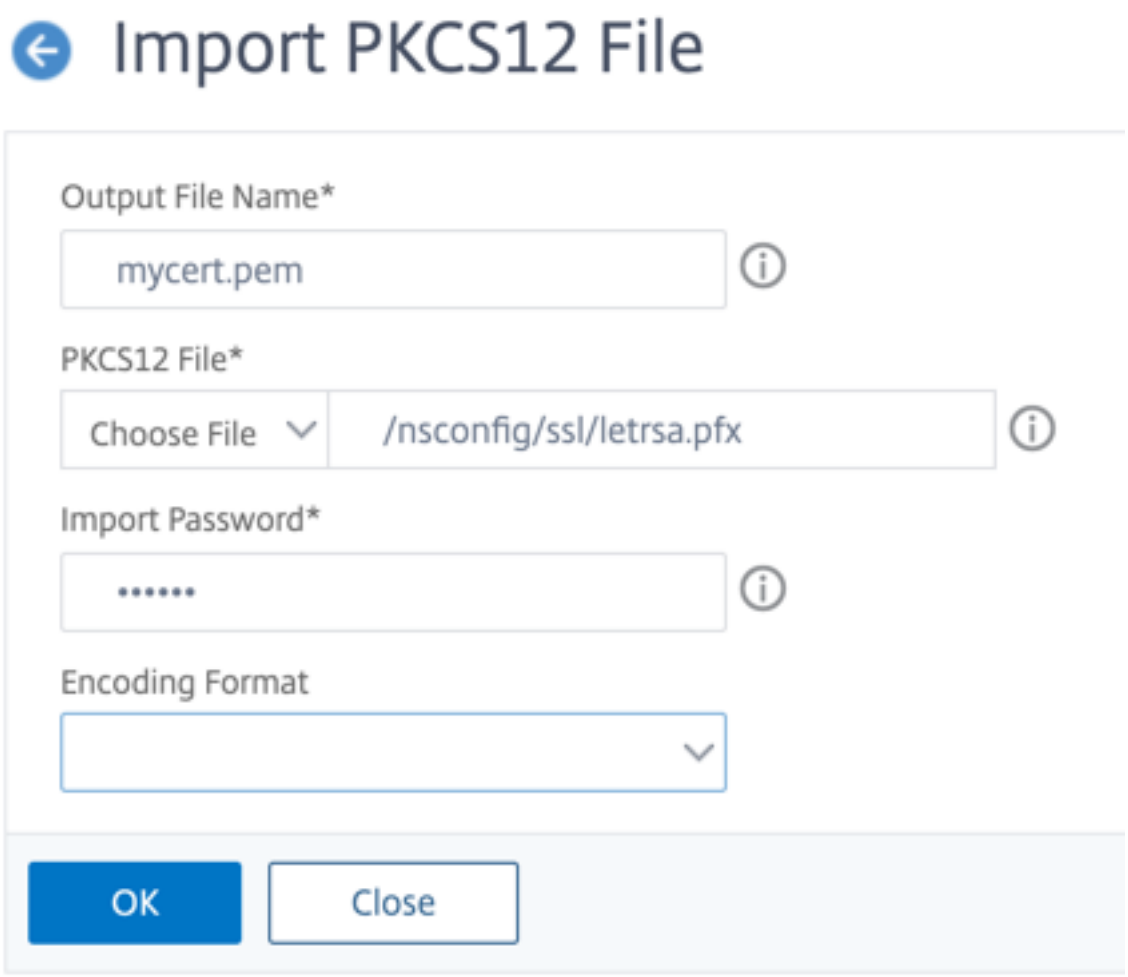
```
convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.
    pfx -des

convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -
    keyFile Key-Client-1
<!--NeedCopy-->
```

Convert the format of a certificate by using the GUI

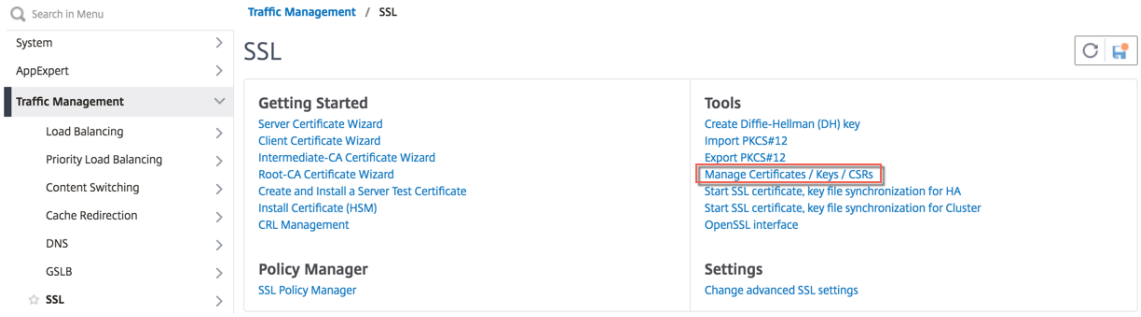
1. Navigate to **Traffic Management > SSL** and, in the **Tools** group, select **Import PKCS#12**.



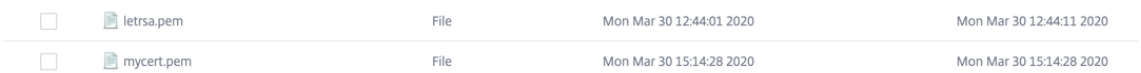
2. Specify the PEM certificate name in the **Output File Name** field.
3. Browse to the location of the PFX certificate on your local computer or the appliance.



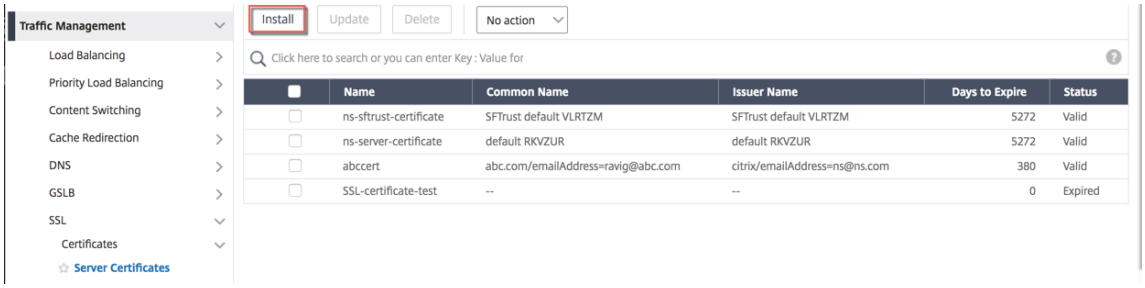
4. Click **OK**.
5. Click **Manage Certificates / Keys / CSRs** to view the converted PEM file.



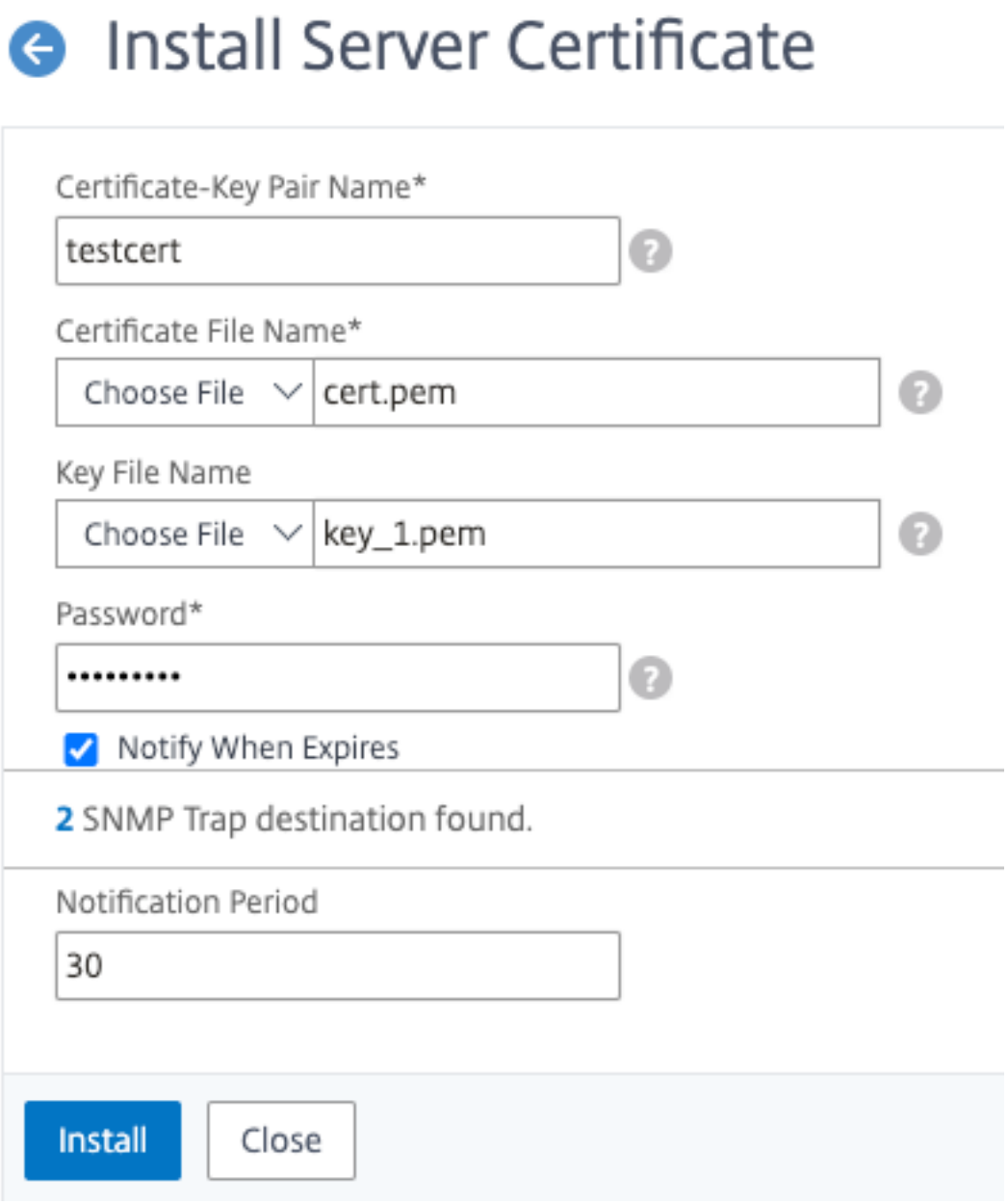
6. You can view the uploaded PFX file and the converted PEM file.



7. Navigate to **SSL** > **Certificates** > **Server Certificates** and click **Install**.



8. Specify a **Certificate-Key Pair Name**.
9. Browse to the location of the PEM file.
10. Specify the password when prompted.
11. Click **Install**.



12. Bind the certificate-key pair to an SSL virtual server.

Bind an SSL certificate to a virtual server on the Citrix ADC appliance

September 17, 2021

An SSL certificate is an essential part of SSL encryption and decryption processes. The certificate is used during an SSL handshake to establish the identity of the SSL server, which is the Citrix ADC appliance as it acts as the SSL termination point for the clients.

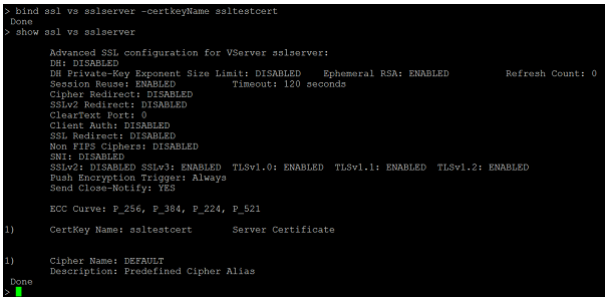
The certificate used for processing the SSL transactions must be bound to the virtual server (SSL) that receives the SSL data.

To bind an SSL certificate to an SSL virtual server using the command line interface

At the command prompt, type:

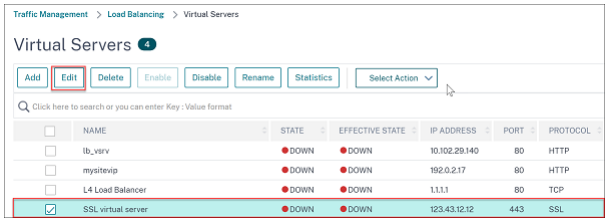
```
bind ssl vs <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vs <vServerName>
<!--NeedCopy-->
```

Example:



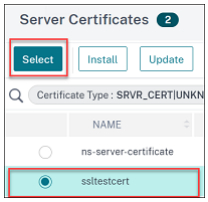
To bind an SSL certificate to an SSL virtual server using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Select a virtual server of type SSL and click **Edit**.



3. In the **Load Balancing Virtual Server** page, under the **Certificates** section, click **No Server Certificate**.

4. In the **Server Certificate Binding** page, click **Click to select**.
5. Select the SSL certificate and click **Select**.



6. Click **Bind** to bind the SSL certificate to the virtual server.
7. Click **Done**.

   You have completed binding the SSL certificate to the virtual server.

SSL profiles

September 14, 2021

An SSL profile is a collection of settings for SSL entities. It offers ease of configuration and flexibility. Instead of configuring the settings on each entity, you can configure them in a profile and bind the profile to all the entities that the settings apply to.

The SSL profile infrastructure has been enhanced to use the latest ciphers and protocols. Differences between the legacy profile (old profile) and the enhanced SSL profile (new profile) are highlighted.

Differences between the old and the new SSL profile infrastructure

Differences	Old Profile	New Profile
Ciphers and ECC Curves included in the profile	No	Yes
Inserting a cipher or cipher group in the middle of an existing list	Unbind all the ciphers and bind again in the order of the required priority.	Add a cipher and assign it a priority. If a priority is not specified, the cipher is assigned the lowest priority in the list.

Differences	Old Profile	New Profile
Unbinding all the ciphers	`unbind ssl vserver \< name\> ciphername -ALL`	`unbind ssl profile -cipherName FlushAllCiphers` (Release 11.0 build 64.x or later includes the FlushAllCiphers parameter for unbinding all the ciphers or cipher groups from a profile, because ALL is treated like a cipher group.)
State of SSLv3	n/a	Disabled on the default front-end profile (ns_default_ssl_profile_frontend). Note: Before you enable this profile, SSLv3 is enabled globally. After enabling the profile, SSLv3 is disabled on the front-end default profile.

SSL profile infrastructure

September 14, 2021

Vulnerabilities in SSLv3 and RC4 implementation have emphasized the need to use the latest ciphers and protocols to negotiate the security settings for a network connection. Implementing any changes to the configuration, such as disabling SSLv3 across thousands of SSL end points, is a cumbersome process. Therefore, settings that were part of the SSL end points configuration have been moved to the SSL profiles, along with the default ciphers. To implement changes in the configuration, including cipher support, you need only modify the profile that is bound to the entities.

The default front-end and default back-end SSL profiles contain all the default ciphers and ECC curves, in addition to the settings that were part of the old profiles. Sample outputs for the default profiles are provided in the appendix. The Enable Default Profile operation automatically binds the default front-end profile to all front-end entities, and the default back-end profile to all back-end entities. You can modify a default profile to suit your deployment. You can also create custom profiles and bind them to SSL entities.

The front-end profile contains parameters applicable to a front-end entity. That is, they apply to the entity that receives requests from a client. Typically, this entity is an SSL virtual server or transpar-

ent SSL service on the Citrix ADC appliance. The back-end profile contains parameters applicable to a back-end entity. That is, they apply to the entity on the ADC appliance that sends client requests to a back-end server. Typically, this entity is an SSL service on the Citrix ADC appliance. If you try to configure an unsupported parameter, the error `ERROR: Specified parameters are not applicable for this type of SSL profile` appears.

> **Important:**
>
> - An SSL profile takes precedence over SSL parameters. That is, if you configure SSL parameters using the `set ssl parameter` command, and later bind a profile to an SSL entity, the settings in the profile take precedence.
> - After the upgrade, if you enable the default profiles, you cannot undo the changes. That is, the profiles cannot be disabled. Save the configuration and create a copy of the configuration file (ns.conf) before enabling the profiles. However, if you do not want to use the features in the default profile, you can continue to use the old SSL profiles. For more information about these profiles, see Legacy SSL profile.
> - From release 11.1 51.x, in the GUI and CLI, a confirmation prompt is added when you enable the default profile to prevent enabling it by mistake.

Command:

```
1 set ssl parameter -defaultProfile ENABLED
2     Save your configuration before enabling the Default profile. You
         cannot undo the changes. Are you sure you want to enable the
         Default profile? [Y/N]Y
3 Done
4 <!--NeedCopy-->
```

By default, some SSL parameters, called *global parameters*, apply to all the SSL end points. However, if a profile is bound to an SSL end point, the global parameters do not apply. The settings specified in the profile apply instead.

Points to note

1. A profile can be bound to multiple virtual servers, but a virtual server can have only one profile bound to it.
2. To delete a profile that is bound to a virtual server, first unbind the profile.
3. A cipher or cipher group can be bound to multiple profiles at different priorities.
4. A profile can have multiple ciphers and cipher groups bound at different priorities.
5. Changes to a cipher group are immediately reflected in all the profiles and in all the virtual servers that one of the profiles is bound to.

6. If a cipher suite is part of a cipher group, edit the cipher group to remove that cipher suite before removing the cipher suite from the profile.
7. If you do not assign a priority to a cipher suite or cipher group attached to a profile, it is assigned the lowest priority within the profile.
8. You can create a custom cipher group (also called a user-defined cipher group) from existing cipher groups and cipher suites. If you create cipher group A and add existing cipher groups X and Y to it, in that order, Y is assigned at a lower priority than X. That is, the group that is added first has a higher priority.
9. If a cipher suite is part of two cipher groups attached to the same profile, the cipher suite is not added as part of the second cipher group. The cipher suite at the higher priority is in effect when traffic is processed.
10. Cipher groups are not expanded in the profile. As a result, the number of lines in the configuration file (ns.conf) is greatly reduced. For example, if two cipher groups containing 15 ciphers each are bound to a thousand SSL virtual servers, expansion adds 30*1000 cipher-related entries in the configuration file. With the new profile, it would have only two entries: one for each cipher group that is bound to a profile.
11. Creating a user defined cipher group from existing ciphers and cipher groups is a copy-paste operation. Any changes in the original group are not reflected in the new group.
12. A user-defined cipher group lists all the profiles that it is a part of.
13. A profile lists all the SSL virtual server, services, and service groups that it is bound to.
14. If the default SSL profile feature is enabled, use the profile to set or change any of the attributes of an SSL entity. For example, virtual server, service, service group, or an internal service.

Save the configuration by using the CLI

At the command prompt, type:

```
save config

shell

root@ns# cd /nsconfig

root@ns# cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber
   >
<!--NeedCopy-->
```

Example:

```
save config
shell
root@ns# cd /nsconfig
```



```
root@ns# cp ns.conf ns.conf.NS.11.0.jun.16
<!--NeedCopy-->
```

Enable the default profile

> **Important:**
>
> Save your configuration before you upgrade the software and enable the default profiles.
>
> From release 11.1 build 51.x, in the GUI and CLI, a confirmation prompt appears when you enable the default profile to avoid enabling it by mistake.

Command: The following command enables the default profile and binds this profile to the SSL entities to which a profile is already bound. That is, if a profile (for example P1) is already bound to an SSL entity, the default front-end profile or the default back-end profile replaces P1. The older profile (P1) is not deleted. It is now an enhanced SSL profile and contains the earlier settings, and the ciphers and ECC curves. If you do not want the default profile, you can explicitly bind P1 to the SSL entity.

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

Upgrade the software to a build that supports the enhanced profile infrastructure, and then enable the default profiles.

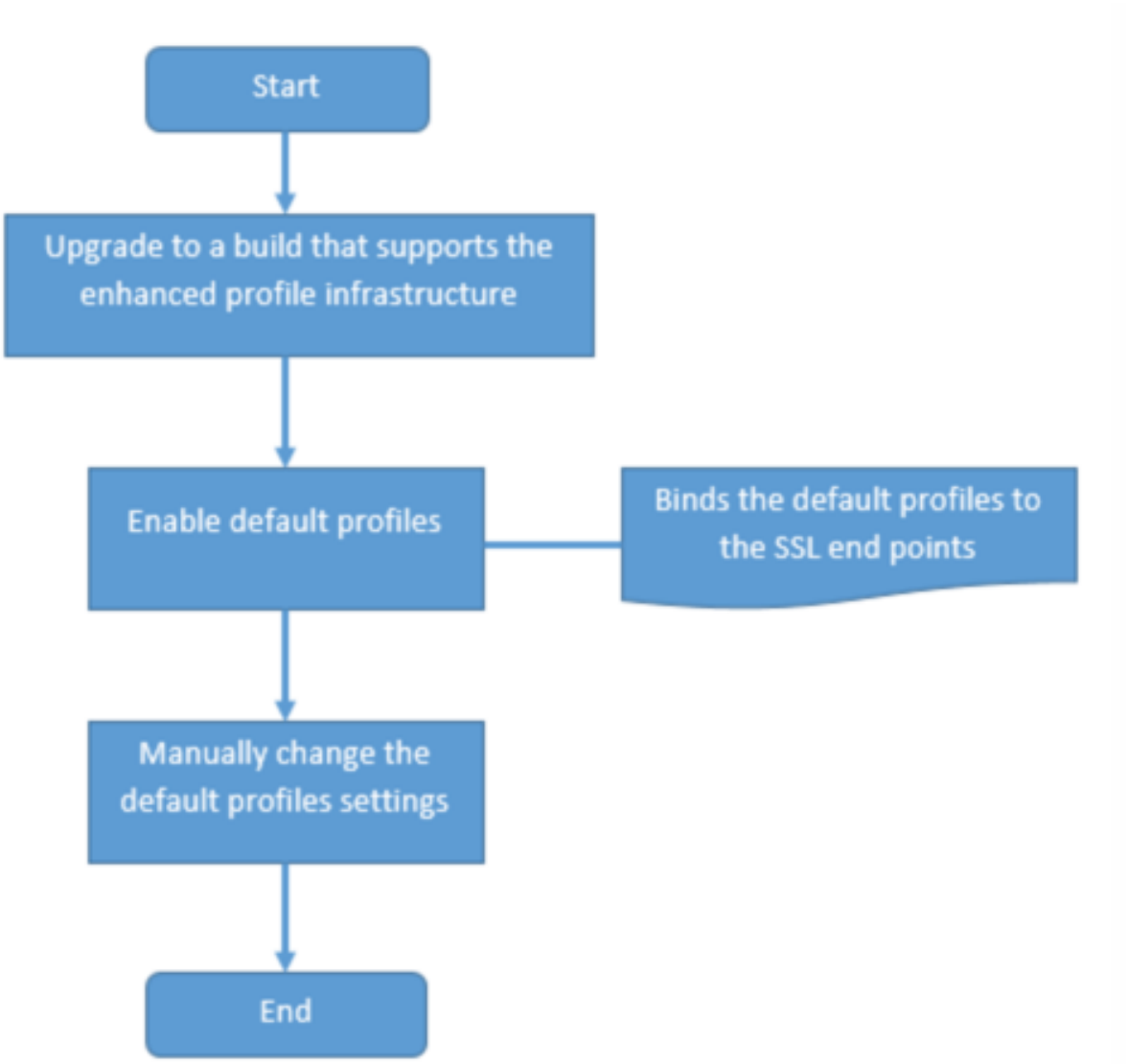
> **Notes:**
>
> - If a legacy profile (P1) is already bound to an SSL entity, and you enable the default profile, the default profile overrides the earlier binding. That is, the default profile is bound to the SSL entities. If you do not want the default profile to be bound, you must bind P1 to the SSL entity again.
> - A single operation (Enable Default Profile or `set ssl parameter -defaultProfile ENABLED`) enables (binds) both the default front-end profile and the default back-end profile.

Use case

After you enable the default profiles, they are bound to all the SSL end points. The default profiles are editable. If your deployment uses most of the default settings and changes only a few parameters, you can edit the default profiles. The changes are immediately reflected across all the end points.

You can also create custom SSL profiles with some custom and some default parameters and bind it to the SSL entities.

The following flowchart explains the steps that you must perform:



1. For information about upgrading the software, see Upgrading the System Software.
2. Enable the default profiles by using the CLI or GUI.

- At the command line, type: `set ssl parameter -defaultProfile ENABLED`
- If you prefer to use the GUI, navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**, scroll down, and select **Enable Default Profile**.

If a profile was not bound to an end point before the upgrade, a default profile is bound to the SSL end point. If a profile was bound to an end point before the upgrade, the same profile is bound after the upgrade, and default ciphers are added to the profile.

1. (Optional) Manually change any settings in the default profile.

- At the command line, type: `set ssl profile <name>` followed by the parameters to modify.
- If you prefer to use the GUI, navigate to **System** > **Profiles**. In **SSL Profiles**, select a profile and click **Edit**.

SSL profile parameters

You can set the following SSL parameters in an SSL profile. You can set some of these parameters in an SSL virtual server. For more information about SSL virtual server parameters, see SSL virtual server parameters.

Support for secure renegotiation at the back end of a Citrix ADC appliance

Note: This parameter is introduced in release 13.0 build 58.x and later. In earlier releases and builds, only non-secure renegotiation was supported on the back end.

The feature is supported on the following platforms:

- VPX
- MPX platforms containing N2 or N3 chips
- Intel Coleto SSL chip based platforms

The feature is not yet supported on the FIPS platform.

Secure renegotiation is denied by default on the back end of an ADC appliance. That is, the `denySSLReneg` parameter is set to ALL (default).

To allow secure renegotiation on the back end, select from one of the following settings for the `denySSLReneg` parameter:

- NO
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

Enable secure renegotiation by using the CLI

At the command prompt, type:

```
set ssl profile <name> -denySSLReneg <denySSLReneg>
```

Example:

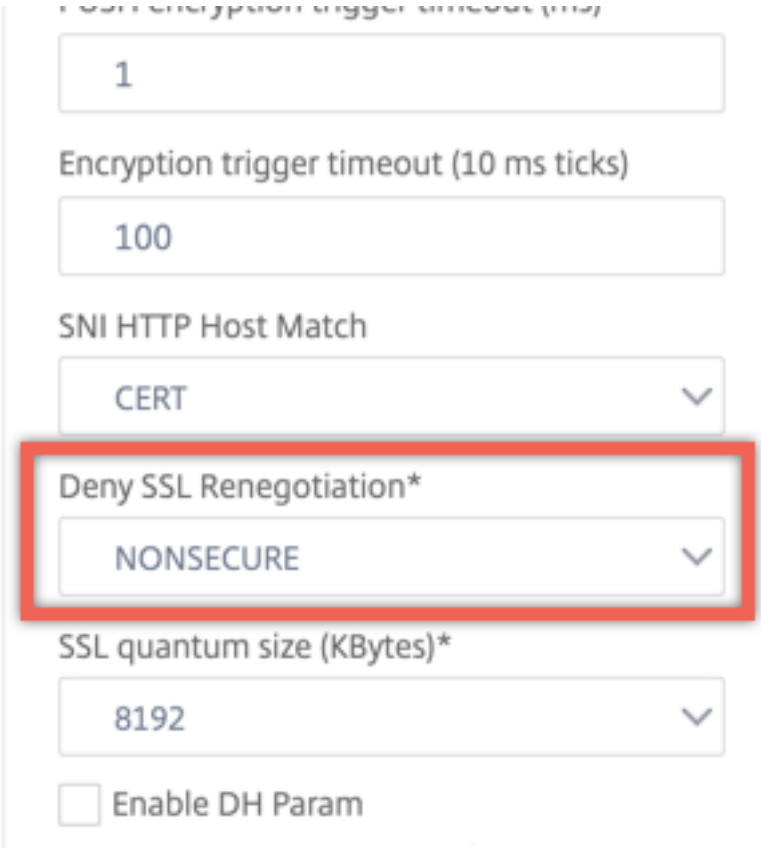
```
1 set ssl profile ns_default_ssl_profile_backend -denySSLReneg NONSECURE
2  Done
```



```
3
4  sh ssl profile ns_default_ssl_profile_backend
5  1)  Name: ns_default_ssl_profile_backend    (Back-End)
6      SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
          ENABLED  TLSv1.3: DISABLED
7      Server Auth: DISABLED
8      Use only bound CA certificates: DISABLED
9      Strict CA checks:       NO
10     Session Reuse: ENABLED      Timeout: 300 seconds
11     DH: DISABLED
12     Ephemeral RSA: DISABLED
13     Deny SSL Renegotiation      NONSECURE
14     Non FIPS Ciphers: DISABLED
15     Cipher Redirect: DISABLED
16     SSL Redirect: DISABLED
17     Send Close-Notify: YES
18     Strict Sig-Digest Check: DISABLED
19     Push Encryption Trigger: Always
20     PUSH encryption trigger timeout:    1 ms
21     SNI: DISABLED
22     OCSP Stapling: DISABLED
23     Strict Host Header check for SNI enabled SSL sessions:      NO
24     Push flag:  0x0 (Auto)
25     SSL quantum size:       8 kB
26     Encryption trigger timeout  100 mS
27     Encryption trigger packet count:    45
28
29     ECC Curve: P_256, P_384, P_224, P_521
30
31 1)  Cipher Name: DEFAULT_BACKEND        Priority :2
32     Description: Predefined Cipher Alias
33
34 1)  Service Name: s187
35  Done
36 <!--NeedCopy-->
```

Enable secure renegotiation by using the GUI

1. Navigate to **System > Profiles > SSL Profile**.
2. Add or edit a profile.
3. Set **Deny SSL Renegotiation** to any value other than ALL.



Host header validation

Note: This parameter is introduced in release 13.0 build 52.x.

With HTTP/1.1, clients had to use multiple connections to process multiple requests. With HTTP/2, clients can reuse connections across domains that are covered by the same certificate. For an SNI enabled session, the ADC appliance must be able to control how the HTTP host header is validated to accommodate this change. In earlier builds, the request was dropped if the parameter was enabled (set to “Yes”) and the request did not contain the host header for an SNI enabled session. If the parameter was disabled (set to “No”), the appliance did not perform the validation. A new parameter `SNIHTTPHostMatch` is added to an SSL profile and SSL global parameters to have better control on this validation. This parameter can take three values; CERT, STRICT, and NONE. These values work as follows for SNI enabled sessions only. SNI must be enabled on the SSL virtual server or the profile bound to the virtual server, and the HTTP request must contain the host header.

- CERT - Connection is forwarded if the host header value in the request is covered by the certificate used to establish this SSL session.
- STRICT - Connection is forwarded only if the host header value in the request matches the server name value passed in the Client Hello message of the SSL connection.
- NO - The host header value is not validated.

Possible values: NO, CERT, STRICT
Default value: CERT

With the introduction of the new parameter `SNIHTTPHostMatch` there is a change in the behavior of the `dropReqWithNoHostHeader` parameter. The setting of the `dropReqWithNoHostHeader` parameter no longer affects how the host header is validated against the SNI certificate.

Set SSL profile parameters by using the CLI

At the command prompt, type:

```
set ssl profile <name> [-ssllogProfile <string>] [-dh ( ENABLED |
    DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-
    dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED |
    DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED |
    DISABLED )
[-sessTimeout <positive_integer>]] [-cipherRedirect ( ENABLED |
    DISABLED ) [-cipherURL <URL>]] [-clientAuth ( ENABLED | DISABLED )[-
    clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
DISABLED )] [-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl3 (
    ENABLED | DISABLED )] [-tls1 ( ENABLED | DISABLED )] [-tls11 (
    ENABLED| DISABLED )] [-tls12 ( ENABLED | DISABLED )] [-tls13 (
    ENABLED |DISABLED )] [-SNIEnable ( ENABLED | DISABLED )] [-
    ocspStapling (ENABLED | DISABLED )] [-serverAuth ( ENABLED |
    DISABLED )] [-commonName <string>] [-pushEncTrigger <pushEncTrigger
    >] [-sendCloseNotify ( YES |
NO )] [-clearTextPort <port|*>] [-insertionEncoding ( Unicode | UTF-8)]
     [-denySSLReneg <denySSLReneg>] [-quantumSize <quantumSize>]
[-strictCAChecks ( YES | NO )] [-encryptTriggerPktCount <
    positive_integer>] [-pushFlag <positive_integer>][-
    dropReqWithNoHostHeader ( YES | NO )] [-SNIHTTPHostMatch <
    SNIHTTPHostMatch>] [-pushEncTriggerTimeout <positive_integer>]
[-sslTriggerTimeout <positive_integer>] [-clientAuthUseBoundCAChain (
    ENABLED | DISABLED )] [-sslInterception ( ENABLED | DISABLED )][-
    ssliReneg ( ENABLED | DISABLED )] [-ssliOCSPCheck ( ENABLED |
    DISABLED )] [-ssliMaxSessPerServer <positive_integer>] [-HSTS (
    ENABLED| DISABLED )] [-maxage <positive_integer>] [-
    IncludeSubdomains ( YES | NO )] [-preload ( YES | NO )] [-
    sessionTicket ( ENABLED | DISABLED )][-sessionTicketLifeTime <
    positive_integer>] [-sessionTicketKeyRefresh (ENABLED | DISABLED )]
    {
 -sessionTicketKeyData  }
  [-sessionKeyLifeTime <positive_integer>] [-prevSessionKeyLifeTime <
     positive_integer>]
```

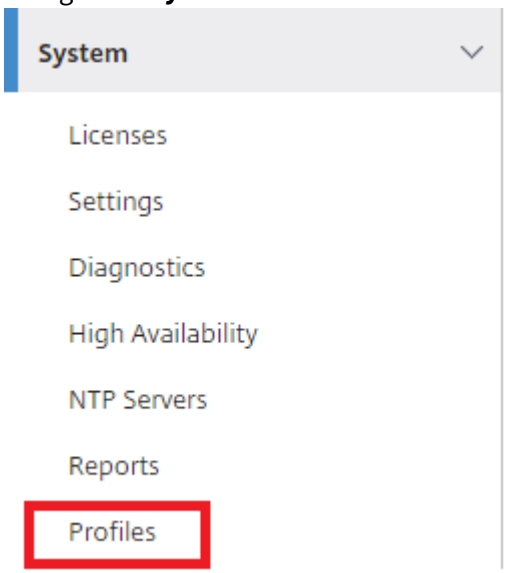


```
[-cipherName <string> -cipherPriority <positive_integer>][-
   strictSigDigestCheck ( ENABLED | DISABLED )]
[-skipClientCertPolicyCheck ( ENABLED | DISABLED )] [-zeroRttEarlyData
   ( ENABLED | DISABLED )] [-tls13SessionTicketsPerAuthContext
<positive_integer>] [-dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

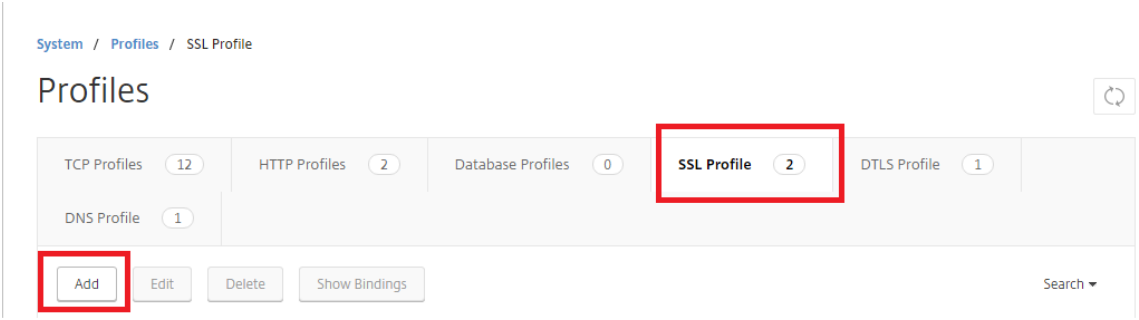
Set SSL profile parameters by using the GUI

To add a profile:

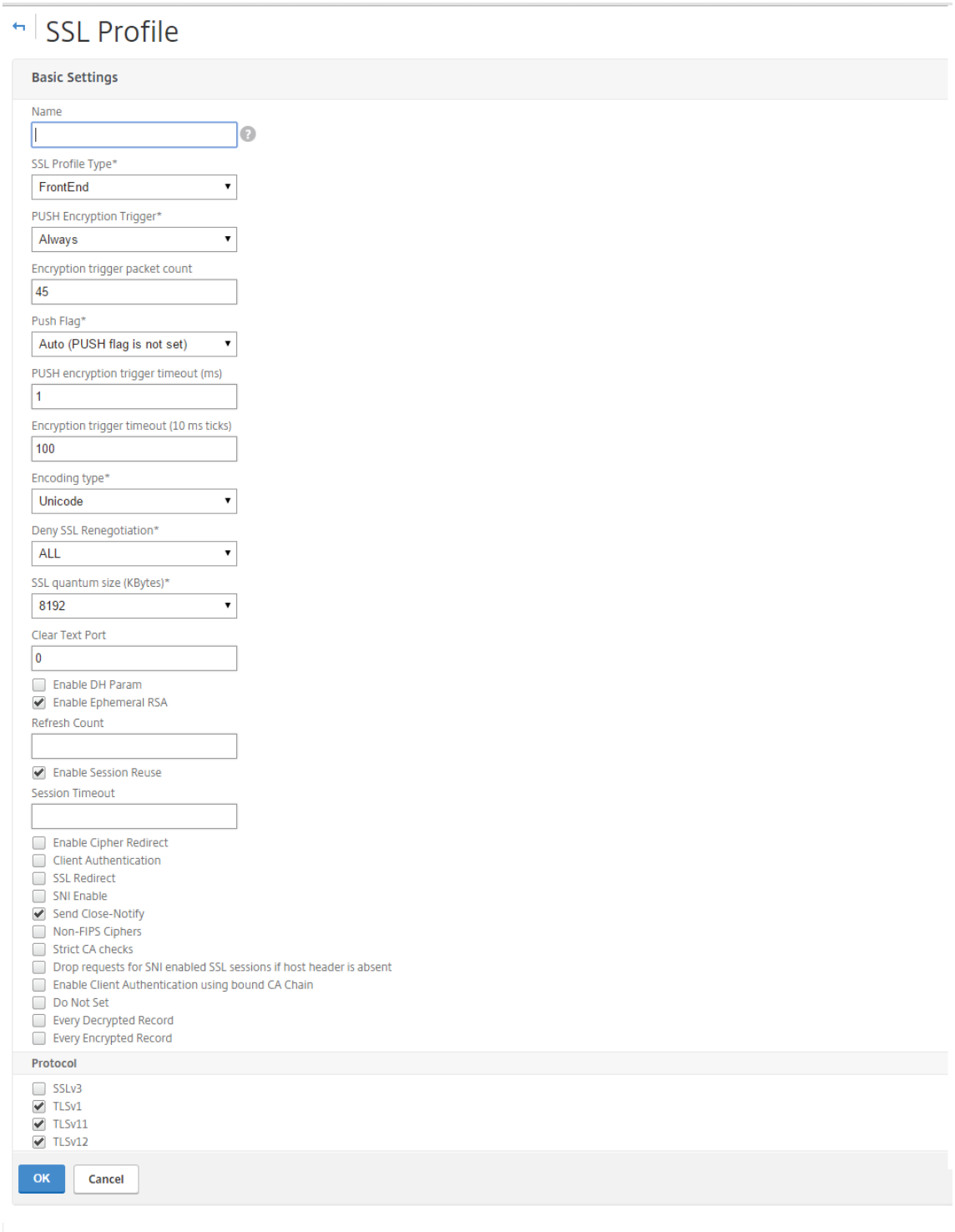
1. Navigate to **System > Profiles**.



2. Select **SSL Profiles**. Click **Add**.



3. Specify values for the different parameters.



4. Click **OK**.
5. Click **Done**.

To reuse an existing SSL profile:

1. Navigate to **System > Profiles**.
2. Select an existing profile and click **Add**.

3. Specify a different name, change any parameters, and click **OK**.
4. Click **Done**.

TLS session ticket extension

An SSL handshake is a CPU-intensive operation. If session reuse is enabled, the server/client key exchange operation is skipped for existing clients. They are allowed to resume their sessions. This action improves the response time and increases the number of SSL transactions per second that a server can support. However, the server must store details of each session state, which consumes memory and is difficult to share among multiple servers if requests are load balanced across servers.

Citrix ADC appliances support the SessionTicket TLS extension. Use of this extension indicates that the session details are stored on the client instead of on the server. The client must indicate that it supports this mechanism by including the session ticket TLS extension in the client Hello message. For new clients, this extension is empty. The server sends a new session ticket in the NewSessionTicket handshake message. The session ticket is encrypted by using a key-pair known only to the server. If a server cannot issue a new ticket now, it completes a regular handshake.

This feature is available only in front-end SSL profiles, and only at the front end of communication in which the appliance acts as a server and generates session tickets.

Limitations

- This feature is not supported on a FIPS platform.
- This feature is supported only with TLS versions 1.1 and 1.2.
- SSL session ID persistency is not supported with session tickets.

Enable TLS session ticket extension by using the CLI

At the command prompt, type:

```
set ssl profile <name> -sessionTicket (ENABLED | DISABLED ) [-
    sessionTicketLifeTime <positive_integer>
<!--NeedCopy-->
```

Arguments:

sessionTicket: State of TLS session ticket extension. Use of this extension indicates that the session details are stored on the client instead of on the server, as defined in RFC 5077.

Possible values: ENABLED, DISABLED

Default value: DISABLED

sessionTicketLifeTime: Specify a time, in seconds, after which the session ticket expires and a new SSL handshake must be initiated.

Default value: 300

Minimum value: 0

Maximum value: 172800

Example:

```
add ssl profile profile1 -sessionTicket ENABLED -sessionTicketlifeTime
    300
Done
<!--NeedCopy-->
```

Enable TLS session ticket extension by using the GUI

1. Navigate to **System** > **Profiles**. Select **SSL Profiles**.
2. Click **Add** and specify a name for the profile.
3. Select **Session ticket**.
4. Optionally, specify **Session Ticket Lifetime (secs)**.

Secure implementation of session tickets

By using TLS session tickets, clients can use abbreviated handshakes for faster reconnection to servers. However, if session tickets aren't encrypted or changed for long periods of time, they can pose a security risk. You can secure session tickets by encrypting them with a symmetric key. To achieve forward secrecy, you can specify a time interval at which the session-ticket key is refreshed.

The appliance generates the session ticket keys by default. However, if multiple appliances in a deployment need to decrypt each other's session tickets, they must all use the same session-ticket key. Therefore, you must set (add or load) the same session-ticket key data manually on all the appliances. Session-ticket key data includes the following information:

- Session ticket name.
- Session AES key used to encrypt or decrypt the ticket.
- Session HMAC key used to compute the digest of the ticket.

You can now configure session ticket key data of length 64 bytes to support 256-bit HMAC keys as recommended in RFC 5077. Key lengths of 48 bytes are also supported for backward compatibility.

> **Note:**
>
> While typing the session-ticket key data manually, ensure that the configuration across all the

> Citrix ADC appliances in an HA setup or in a cluster setup is the same.

The `sessionTicketKeyLifeTime` parameter specifies how often a session-ticket key is refreshed. You can set the `prevSessionTicketKeyLifeTime` parameter to specify how long the previous session-ticket key will be maintained for decrypting tickets using that key, after a new key is generated. The `prevSessionTicketKeyLifeTime` setting extends the time for which a client can use an abbreviated handshake to reconnect. For example, if `sessionTicketKeyLifeTime` is set to 10 minutes and `prevSessionTicketKeyLifeTime` to 5 minutes, a new key is generated after 10 minutes and used for all new sessions. However, previously connected clients have another 5 minutes for which previously issued tickets are honored for an abbreviated handshake.

Configure SSL session-ticket data by using the CLI

At the command prompt, type:

```
set ssl profile <name> -sessionTicket ENABLED -sessionTicketLifeTime <
    positive_integer> -sessionTicketKeyRefresh ( ENABLED | DISABLED )] -
    sessionTicketKeyLifeTime <positive_integer> [-
    prevSessionTicketKeyLifeTime <positive_integer>]
<!--NeedCopy-->
```

Arguments:

sessionTicket: Use session tickets as described by RFC 5077. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the **ENABLED** setting, a server issues a session ticket to a client, which the client can use to perform an abbreviated handshake.

Possible values: ENABLED, DISABLED. Default: DISABLED

sessionTicketLifeTime: Lifetime, in seconds, of the session ticket. After this time expires, clients cannot use this ticket to resume their sessions.

Maximum value: 172800. Minimum value: 0. Default: 300.

sessionTicketKeyRefresh: When the time specified by the session-ticket key lifetime parameter expires, regenerate the session-ticket key used to encrypt or decrypt the session tickets. Automatically enabled if sessionTicket is enabled. Disabled if an administrator enters the session-ticket data.

Possible values: ENABLED, DISABLED. Default: ENABLED

sessionKeyLifeTime: Lifetime, in seconds, of a symmetric key used to encrypt the session tickets issued by a Citrix ADC appliance.

Maximum value: 86400. Minimum value: 600. Default: 3000

prevSessionKeyLifeTime: Time, in seconds, for which the previous symmetric key used to encrypt session tickets remains valid for existing clients after the session-ticket key lifetime expires. Within

this time, existing clients can resume their sessions by using the previous session ticket key. Session tickets for new clients are encrypted by using the new key.

Maximum value: 172800. Minimum value: 0. Default: 0

Example:

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED
    -sessionTicketlifeTime 120 -sessionTicketKeyRefresh ENABLED -
    sessionTicketKeyLifeTime 100 -prevSessionTicketKeyLifeTime 60

Done

show ssl profile ns_default_ssl_profile_frontend

    Session Ticket: ENABLED
    Session Ticket Lifetime: 120 (secs)
    Session Key Auto Refresh: ENABLED
    Session Key Lifetime: 100 (secs)
    Previous Session Key Lifetime: 60 (secs)
<!--NeedCopy-->
```

Configure SSL session-ticket data by using the GUI

1. Navigate to **System** > **Profiles**, and select **SSL Profile**.
2. Select **ns_default_ssl_profile_frontend** and click **Edit**.
3. In the **Basic Settings** section, click the pencil icon and set the following parameters:
   - Session Ticket
   - Session Ticket Lifetime (secs)
   - Session Ticket Key Auto Refresh
   - Session Ticket Key Lifetime (secs)
   - Previous Session Ticket Key Lifetime (secs)
4. Click **OK**.

Type SSL session ticket data manually by using the CLI

At the command prompt, type:

```
set ssl profile <name> -sessionTicket ENABLED

set ssl profile <name> -sessionTicketKeyData

```



```
show ssl profile ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

Arguments:

sessionTicket: Use of session tickets as described by RFC 5077. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the **ENABLED** setting, a server issues a session ticket to a client, which the client can use to perform an abbreviated handshake.

Possible values: ENABLED, DISABLED. Default: DISABLED

sessionTicketKeyData: Contains the session ticket name (0-15 bytes), the session AES key used to encrypt or decrypt the session ticket (16-31 bytes), and the session HMAC key used to compute the digest of the ticket (32-63 bytes). Externally generated by an administrator and added to a Citrix ADC appliance.

Maximum length: 64 bytes

Example:

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED

Done

set ssl profile ns_default_ssl_profile_frontend -sessionTicketKeyData
    1111111111111111111111111111111111111111111111111

Done

show ssl profile ns_default_ssl_profile_frontend

    1) Name: ns_default_ssl_profile_frontend (Front-End)
    SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1.2: ENABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks: NO
    Session Reuse: ENABLED Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA: ENABLED
        Refresh Count: 0
    Deny SSL Renegotiation ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
```



```
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout: 1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag: 0x0 (Auto)
    SSL quantum size: 8 kB
    Encryption trigger timeout 100 mS
    Encryption trigger packet count: 45
    Subject/Issuer Name Insertion Format: Unicode
    Session Ticket: ENABLED
    Session Ticket Lifetime: 300 (secs)
    Session Key Auto Refresh: DISABLED
    Session Key Lifetime: 3000 (secs)
    Previous Session Key Lifetime: 0 (secs)
    Session Key Data: 84
       dad1afc6d56b0deeb0a7fd7f299a207e8d8c15cdd087a5684a11a329fd732e87a0535d9088

    47
       e8c181ba266f5c8838ae472cb3ab9255b683bf922fad32cee816c329989ef7cdeb278e93ac


    ECC Curve: P_256, P_384, P_224, P_521

    1) Cipher Name: DEFAULT Priority :4
    Description: Predefined Cipher Alias

    1) Internal Service Name (Front-End): nsrnatsip-127.0.0.1-5061
    2) Internal Service Name (Front-End): nskrpcs-127.0.0.1-3009
    3) Internal Service Name (Front-End): nshttps-::1l-443
    4) Internal Service Name (Front-End): nsrpcs-::1l-3008
    5) Internal Service Name (Front-End): nshttps-127.0.0.1-443
    6) Internal Service Name (Front-End): nsrpcs-127.0.0.1-3008
    7) Vserver Name: v1

Done
<!--NeedCopy-->
```

Type SSL session ticket data manually by using the GUI

1. Navigate to **System > Profiles**, and select **SSL Profile**.
2. Select **ns_default_ssl_profile_frontend** and click **Edit**.
3. In the **Basic Settings** section, click the pencil icon and set the following parameters:

- Session Ticket
- Session Ticket Key Data
- Confirm Session Ticket Key Data

4. Click **OK**.

Support for Extended Master Secret in SSL handshake on Citrix ADC non-FIPS platforms

Note: This parameter is introduced in release 13.0 build 61.x.

Extended Master Secret (EMS) is an optional extension to the Transport Layer Security (TLS) protocol. A new parameter is added that applies to both front-end and back-end SSL profiles to support EMS on the Citrix ADC appliance. If the parameter is enabled and the peer supports EMS, the ADC appliance uses the EMS calculation. If the peer does not support EMS, then the EMS calculation is not used for the connection even though the parameter is enabled on the appliance. For more information about EMS, see RFC 7627.

Note: EMS is only applicable for handshakes that use TLS protocol version 1.0, 1.1, or 1.2.

Platform support for EMS

- MPX and SDX platforms containing either Cavium N3 chips or Intel Coleto Creek crypto cards. The following platforms ship with Intel Coleto chips:
- MPX 5900
- MPX/SDX 8900
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPS/SDX 26000-100G
- MPX/SDX 15000-50G

You can also use the “show hardware’ command to identify whether your appliance has Coleto (COL) or N3 chips.

- MPX and SDX platforms without crypto cards (software-only).
- Software-only platforms: VPX, CPX, and BLX.

EMS cannot be enabled on the following platforms:

- MPX 9700 FIPS and MPX 14000 FIPS platforms.
- MPX and SDX platforms containing Cavium N2 crypto chips.

If the parameter is enabled, the ADC appliance attempts to use EMS in TLS 1.2, TLS 1.1, and TLS 1.0 connections. The setting does not affect TLS 1.3 or SSLv3 connections.

To allow EMS to be negotiated with the peer, enable the setting on the SSL profile bound to the virtual server (front end) or service (back end).

Enable EMS using the CLI

At the command prompt, type:

```
set ssl profile <profile name> [-allowExtendedMasterSecret (YES | NO)]
```

Examples

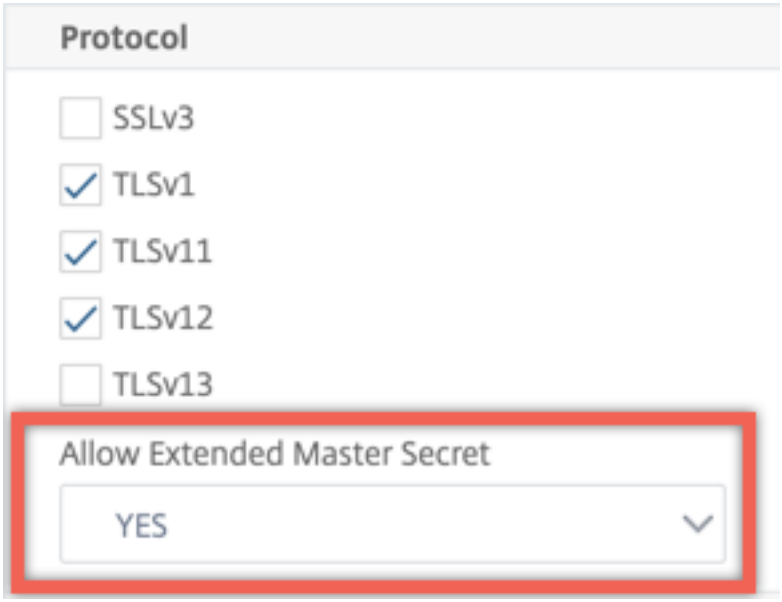
```
1 set ssl profile ns_default_ssl_profile_frontend  -
     allowExtendedMasterSecret YES
2
3 set ssl profile ns_default_ssl_profile_backend  -
     allowExtendedMasterSecret YES
4 <!--NeedCopy-->
```

The following table shows the default value of the `allowExtendedMasterSecret` parameter on different default and user defined profiles.

Profile	Default setting
Default front-end profile	NO
Default front-end secure profile	YES
Default back-end profile	NO
User-defined profile	NO

Enable EMS using the GUI

1. Navigate to **System > Profiles > SSL Profile**.
2. Add a profile or edit a profile.
3. Set **Allow Extended Master Secret** to YES.



Support for processing of ALPN extension in the client hello message

Note: This feature is supported in release 13.0 build 61.x and later.

A parameter `alpnProtocol` is added to the front-end SSL profiles to negotiate the application protocol in the ALPN extension for the connections handled by the SSL_TCP virtual server. Only the protocol specified in the SSL profile is negotiated, if the same protocol is received in the ALPN extension of the client hello message.

Note: The `alpnProtocol` parameter is supported only on front end SSL profiles and is applicable to the SSL connections handled by SSL_TCP type virtual servers.

Set the protocol in the front-end SSL profile using the CLI

At the command prompt, type:

```
set ssl profile ns_default_ssl_profile_frontend -alpnProtocol <protocol_name
>
```

The `alpnProtocol` parameter can take three values. Maximum length: 4096 bytes.

- **NONE:** Application protocol negotiation does not take place. This setting is the default.
- **HTTP1:** HTTP1 can be negotiated as the application protocol.
- **HTTP2:** HTTP2 can be negotiated as the application protocol.

Example:

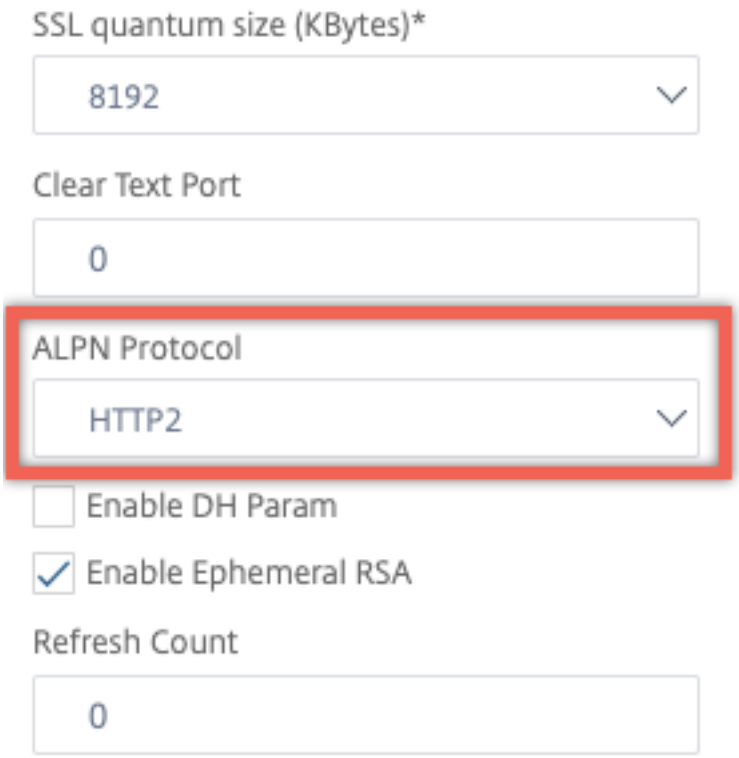
```
1 set ssl profile ns_default_ssl_profile_frontend -ALPNProtocol HTTP2
2 > sh ssl profile ns_default_ssl_profile_frontend
```



```
1)  Name: ns_default_ssl_profile_frontend   (Front-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED  Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED  Refresh Count: 0
    Deny SSL Renegotiation  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:  NO
    Match HTTP Host header with SNI:    CERT
    Push flag:  0x0 (Auto)
    SSL quantum size:   8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45
    Subject/Issuer Name Insertion Format:   Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server: 10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    Allow Extended Master Secret: NO
    Send ALPN Protocol: HTTP2

 Done
 <!--NeedCopy-->
```


Set the protocol in the front-end SSL profile using the GUI

1. Navigate to **System > Profiles**, and select **SSL Profile**.
2. Select **ns_default_ssl_profile_frontend** and click **Edit**.
3. In the **ALPN Protocol** list, select **HTTP2**.



Load an old configuration

Enabling the default profiles is not reversible. However, if you decide that your deployment does not require the default profiles, you can load an older configuration that you saved before you enabled the default profiles. The changes are effective after you restart the appliance.

Load an old configuration by using the CLI

At the command prompt, type:

```
shell

root@ns# clear config
```



```
4
5  root@ns# cd /nsconfig
6
7  root@ns# cp ns.conf.NS.11.0.jun.16 ns.conf
8
9  root@ns# reboot
10 <!--NeedCopy-->
```

Secure front-end profile

September 14, 2021

In addition to a default front-end and a default back-end profile, a new default secure front-end profile is available from release 12.1. The settings required for an A+ rating (as of May 2018) from Qualys SSL Labs are preloaded into this profile. Earlier, you had to explicitly set each of the parameters required for an A+ rating on an SSL front-end profile or an SSL virtual server. Now you can bind the ns_default_ssl_profile_secure_frontend profile to your SSL virtual server and the required parameters are automatically set on your SSL virtual server.

> **Note:**
>
> The secure front-end profile is not editable.

When you enable the default profile, the default front-end profile is automatically bound to all the SSL virtual servers. To get an A+ rating, you must explicitly bind the ns_default_ssl_profile_secure_frontend profile and also bind an SHA2/SHA256 server certificate to your SSL virtual server.

Secure front-end profile parameters

The parameters with their default settings are listed here:

```
1  SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2: ENABLED
       TLSv1.3: DISABLED
2
3  Deny SSL Renegotiation: NONSECURE
4
5  HSTS: ENABLED
6
7  HSTS IncludeSubDomains: YES
8
9  HSTS Max-Age: 15552000
10
```



```
Cipher Name: SECURE    Priority :1
<!--NeedCopy-->
```

Secure cipher alias

A new secure cipher alias is added and bound to the secure front-end profile. To list the ciphers that are part of this alias, at the command prompt type: show cipher SECURE

```
show cipher SECURE

    1) Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 1
           Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256)
              Mac=AEAD   HexCode=0xc030
    2) Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 2
           Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128)
              Mac=AEAD   HexCode=0xc02f
    3) Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
       Priority : 3
           Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256)
              Mac=AEAD   HexCode=0xc02c
    4) Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
       Priority : 4
           Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128)
              Mac=AEAD   HexCode=0xc02b
Done
<!--NeedCopy-->
```

Configuration

Perform the following steps:

1. Add a load balancing virtual server of type SSL.
2. Bind a SHA2/SHA256 certificate.
3. Enable the default profile.
4. Bind the secure front-end profile to the SSL virtual server.

Get an A+ rating for an SSL virtual server by using the CLI

At the command prompt, type:

```
add lb vserver <name> <serviceType> <IPAddress> <port>
bind ssl vserver <vServerName> -certkeyName <string>
set ssl parameter -defaultProfile ENABLED
```



```
set ssl vserver <vServerName> -sslProfile
   ns_default_ssl_profile_secure_frontend
show ssl vserver [<vServerName>]
<!--NeedCopy-->
```

Example:

```
add lb vserver ssl-vsvr SSL 192.0.2.240 443

bind ssl vserver ssl-vsvr -certkeyName letrsa

set ssl parameter -defaultProfile ENABLED

Save your configuration before enabling the Default profile. You cannot
    undo the changes. Are you sure you want to enable the Default
   profile? [Y/N]y

set ssl vserver ssl-vsvr -sslProfile
   ns_default_ssl_profile_secure_frontend
<!--NeedCopy-->
```

```
sh ssl vserver ssl-vsvr

    Advanced SSL configuration for VServer ssl-vsvr:
    Profile Name :ns_default_ssl_profile_secure_frontend
    1) CertKey Name: letrsa     Server Certificate
Done
<!--NeedCopy-->
```

```
sh ssl profile ns_default_ssl_profile_secure_frontend

    1) Name: ns_default_ssl_profile_secure_frontend (Front-End)
    SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:                   NO
    Session Reuse: ENABLED   Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED     Refresh Count: 0
    Deny SSL Renegotiation                NONSECURE
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
```



```
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:  1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:
                            NO
    Push flag:              0x0 (Auto)
    SSL quantum size:                8 kB
    Encryption trigger timeout       100 mS
    Encryption trigger packet count: 45
    Subject/Issuer Name Insertion Format:  Unicode
    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:   10
    Session Ticket: DISABLED
    HSTS: ENABLED
    HSTS IncludeSubDomains: YES
    HSTS Max-Age: 15552000
    ECC Curve: P_256, P_384, P_224, P_521
    1) Cipher Name: SECURE    Priority :1
    Description: Predefined Cipher Alias
    1) Vserver Name: v2
Done
<!--NeedCopy-->
```

Get an A+ rating for an SSL virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and select an SSL virtual server.
2. In Advanced Settings, click SSL Profile.
3. Select ns_default_ssl_profile_secure_frontend.
4. Click OK.
5. Click Done.

Appendix A: Sample migration of the SSL configuration after upgrade

September 14, 2021

Note: This content has been removed because the SSL migration script for the new default profile is no longer supported.

Appendix B: Default front-end and back-end SSL profile settings

September 14, 2021

A default front-end profile has the following settings:

```
sh ssl profile ns_default_ssl_profile_frontend

1)Name: ns_default_ssl_profile_frontend

     Configuration for Front-End SSL profile
     DH: DISABLED
     Ephemeral RSA: ENABLED          Refresh Count: 0
     Session Reuse: ENABLED          Timeout: 120 seconds
     Non FIPS Ciphers: DISABLED
     Cipher Redirect: ENABLED   Redirect URL: http://10.102.28.212/
        redirect.html
     Client Auth: DISABLED
     SSL Redirect: DISABLED
     SNI: DISABLED
     SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
        ENABLED
     Push Encryption Trigger: Always
     PUSH encryption trigger timeout:     1 ms
     Send Close-Notify: YES
     Push flag: 0x0 (Auto)
     Deny SSL Renegotiation          NO
     SSL quantum size:          8 kB
     Strict CA checks:          NO
     Encryption trigger timeout 100 mS
     Encryption trigger packet count:     45
     Use only bound CA certificates: DISABLED
     Subject/Issuer Name Insertion Format: Unicode
     Strict Host Header check for SNI enabled SSL sessions:          NO

```



```
28       ECC Curve: P_256, P_384, P_521
29
30  1)   Cipher Name: AES     Priority :2
31       Description: Predefined Cipher Alias
32
33  1)   Vserver Name: v1
34  2)   Vserver Name: nshttps-::1l-443
35  3)   Vserver Name: nsrpcs-::1l-3008
36  4)   Vserver Name: nskrpcs-127.0.0.1-3009
37  5)   Vserver Name: nshttps-127.0.0.1-443
38  6)   Vserver Name: nsrpcs-127.0.0.1-3008
39  Done
40  <!--NeedCopy-->
```

A default back-end profile has the following settings:

```
1   sh ssl profile ns_default_ssl_profile_backend
2
3   1)Name: ns_default_ssl_profile_backend
4
5        Configuration for Back-End SSL profile
6        Session Reuse: ENABLED          Timeout: 300 seconds
7        Non FIPS Ciphers: DISABLED
8        Server Auth: DISABLED
9        SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2:
             DISABLED
10       Push Encryption Trigger: Always
11       PUSH encryption trigger timeout:     1 ms
12       Send Close-Notify: YES
13       Push flag: 0x0 (Auto)
14       Deny SSL Renegotiation          ALL
15       SSL quantum size:          8 kB
16       Strict CA checks:          NO
17       Encryption trigger timeout 100 mS
18       Encryption trigger packet count:     45
19       Use only bound CA certificates: DISABLED
20
21       ECC Curve: P_256, P_224, P_521
22
23  1)   Cipher Name: AES     Priority :1
24       Description: Predefined Cipher Alias
25
26  2)   Cipher Name: RC4     Priority :2
27       Description: Predefined Cipher Alias
28
```



```
1)   Service Name: s2
2)   Service Name: s1
Done
<!--NeedCopy-->
```

Legacy SSL profile

September 14, 2021

> **Note:**
>
> Citrix recommends using the enhanced profiles instead of legacy profiles. For information about the enhanced profile infrastructure, see SSL profile infrastructure.

Important:

> Bind an SSL profile to an SSL virtual server. Do not bind a DTLS profile to an SSL virtual server. For information about DTLS profiles, see DTLS Profiles.

You can use an SSL profile to specify how a Citrix ADC processes SSL traffic. The profile is a collection of SSL parameter settings for SSL entities, such as virtual servers, services, and service groups, and offers ease of configuration and flexibility. You are not limited to configuring only one set of global parameters. You can create multiple sets (profiles) of global parameters and assign different sets to different SSL entities. SSL profiles are classified into two categories:

- Front end profiles, containing parameters applicable to the front-end entity. That is, they apply to the entity that receives requests from a client.
- Back-end profiles, containing parameters applicable to the back-end entity. That is, they apply to the entity that sends client requests to a server.

Unlike a TCP or HTTP profile, an SSL profile is optional. Therefore, there is no default SSL profile. The same profile can be reused across multiples entities. If an entity does not have a profile attached, the values set at the global level apply. For dynamically learned services, current global values apply.

The following table lists the parameters that are part of each profile.

Front end profile	Back-end profile
cipherRedirect, cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth, clientCert	nonFipsCiphers
denySSLReneg	pushEncTrigger

Front end profile	Back-end profile
dh, dhFile, dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA, eRSACount	serverAuth
insertionEncoding	commonName
nonFipsCiphers	sessReuse, sessTimeout
pushEncTrigger	SNIEnable
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	tls1
sendCloseNotify	-
sessReuse, sessTimeout	-
SNIEnable	-
ssl3	-
sslRedirect	-
sslTriggerTimeout	-
strictCAChecks	-
tls1, tls11, tls12	-

* The clearTextPort parameter applies only to an SSL virtual server.

An error message appears if you try to set a parameter that is not part of the profile. For example, if you try to set the clientAuth parameter in a back-end profile.

Some SSL parameters, such as CRL memory size, OCSP cache size, UndefAction Control, and UndefAction Data, are not part of any of the preceding profiles, because these parameters are independent of entities.

An SSL profile supports the following operations:

- Add—Creates an SSL profile on the Citrix ADC. Specify whether the profile is front end or back end. Front end is the default.
- Set—Modifies the settings of an existing profile.

- Unset—Sets the specified parameters to their default values. If you do not specify any parameters, an error message appears. If you unset a profile on an entity, the profile is unbound from the entity.
- Remove—Deletes a profile. A profile that is being used by any entity cannot be deleted. Clearing the configuration deletes all the entities. As a result, the profiles are also deleted.
- Show—Displays all the profiles that are available on the Citrix ADC. If a profile name is specified, the details of that profile are displayed. If an entity is specified, the profiles associated with that entity are displayed.

Create an SSL profile by using the CLI

- To add an SSL profile, type:

```
add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )]
<!--NeedCopy-->
```

- To modify an existing profile, type:

```
set ssl profile <name>
<!--NeedCopy-->
```

- To unset an existing profile, type:

```
unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA] ...
<!--NeedCopy-->
```

- To unset an existing profile from an entity, type:

```
unset ssl vserver <vServerName>  -sslProfile
<!--NeedCopy-->
```

- To remove an existing profile, type:

```
rm ssl profile <name>
<!--NeedCopy-->
```

- To display an existing profile, type:

```
sh ssl profile <name>
<!--NeedCopy-->
```

Create an SSL profile by using the GUI

Navigate to **System** > **Profiles**, select the SSL Profiles tab, and create an SSL profile.

Enable stricter control on client certificate validation

The Citrix ADC appliance accepts valid Intermediate-CA certificates if a single Root-CA has issued them. That is, if only the Root-CA certificate is bound to the virtual server, and that Root-CA validates any of the intermediate certificates sent with the client certificate, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, the intermediate certificates can be validated by using a CRL or OCSP responder only if that certificate is bound to the SSL virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that one of the CA certificates bound to that virtual server has signed. To do so, you must enable the `ClientAuthUseBoundCAChain` setting in the SSL profile bound to the virtual server. The handshake fails if one of the CA certificates bound to the virtual server has not signed the client certificate.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, the Citrix ADC appliance only accepts clientcert1.

Enable stricter control on client certificate validation by using the CLI

At the command prompt, type: `set ssl profile <name> -ClientAuthUseBoundCAChain Enabled`

Enable stricter control on client certificate validation by using the GUI

1. Navigate to **System** > **Profiles**, select the **SSL Profiles** tab, and create an SSL profile, or select an existing profile.
2. Select **Enable Client Authentication using bound CA Chain**.

Certificate revocation lists

September 14, 2021

A certificate issued by a CA typically remains valid until its expiration date. However, in some circumstances, the CA might revoke the issued certificate before the expiration date. For example, when an owner's private key is compromised, a company's or individual's name changes, or the association between the subject and the CA changes.

A Certificate Revocation List (CRL) identifies invalid certificates by serial number and issuer.

Certificate authorities issue CRLs regularly. You can configure the Citrix ADC appliance to use a CRL to block client requests that present invalid certificates.

If you already have a CRL file from a CA, add that to the Citrix ADC appliance. You can configure refresh options. You can also configure the Citrix ADC to sync the CRL file automatically at a specified interval, from either a web location or an LDAP location. The appliance supports CRLs in either the PEM or the DER file format. Be sure to specify the file format of the CRL file being added to the Citrix ADC appliance.

If you have used the ADC as a CA to create certificates that are used in SSL deployments, you can also create a CRL to revoke a particular certificate. This feature can be used, for example, to ensure that self-signed certificates that are created on the Citrix ADC are not used either in a production environment or beyond a particular date.

Note:

By default, CRLs are stored in the /var/netscaler/ssl directory on the Citrix ADC appliance.

Create a CRL on the ADC appliance

Since you can use the ADC appliance to act as a CA and create self-signed certificates, you can also revoke the following certificates:

- Certificates that you have created.
- Certificates whose CA certificate you own.

The appliance must revoke invalid certificates before creating a CRL for those certificates. The appliance stores the serial numbers of revoked certificates in an index file and updates the file each time it revokes a certificate. The index file is automatically created the first time a certificate is revoked.

Revoke a certificate or create a CRL by using the CLI

At the command prompt, type the following command:

```
1 create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <
     input_filename> | -genCRL <output_filename>)
2 <!--NeedCopy-->
```

Example:

```
1 create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1
2
3 create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1
4 <!--NeedCopy-->
```


Revoke a certificate or create a CRL by using the GUI

1. Navigate to **Traffic Management > SSL** and, in the Getting Started group, select CRL Management.
2. Enter the certificate details and, in the **Choose Operation** list, select **Revoke Certificate**, or **Generate CRL**.

Add an existing CRL to the ADC

Before you configure the CRL on the Citrix ADC appliance, make sure that the CRL file is stored locally on the Citrix ADC appliance. In an HA setup, the CRL file must be present on both ADC appliances, and the directory path to the file must be the same on both appliances.

Add a CRL on the Citrix ADC by using the CLI

At the command prompt, type the following commands to add a CRL on the Citrix ADC and verify the configuration:

```
add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

Example:

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM

Done

> show ssl crl crl-one

            Name: crl-one   Status: Valid,  Days to expiration: 29
            CRL Path: /var/netscaler/ssl/CRL-one
            Format: PEM     CAcert: samplecertkey
            Refresh: DISABLED
            Version: 1
            Signature Algorithm: sha1WithRSAEncryption
            Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,
               OU=SSL Acceleration,CN=www.ns.com/emailAddress=
               support@Citrix ADC appliance.com
            Last_update:Jun 15 10:53:53 2010 GMT
            Next_update:Jul 15 10:53:53 2010 GMT

    1)      Serial Number: 00
```



```
            Revocation Date:Jun 15 10:51:16 2010 GMT
      Done
<!--NeedCopy-->
```

Add a CRL on the Citrix ADC by using the GUI

Navigate to **Traffic Management** > **SSL** > **CRL**, and add a CRL.

Configure CRL refresh parameters

A CRL is generated and published by a Certificate Authority periodically or, sometimes, immediately after a particular certificate is revoked. Citrix recommends that you update CRLs on the Citrix ADC appliance regularly, for protection against clients trying to connect with certificates that are not valid.

The Citrix ADC appliance can refresh CRLs from a web location or an LDAP directory. When you specify refresh parameters and a web location or an LDAP server, the CRL does not have to be present on the local hard disk drive at the time you run the command. The first refresh stores a copy on the local hard disk drive, in the path specified by the CRL File parameter. The default path for storing the CRL is /var/netscaler/ssl.

Note: In release 10.0 and later, the method for refreshing a CRL is not included by default. Specify an HTTP or LDAP method. If you are upgrading from an earlier release to release 10.0 or later, you must add a method and run the command again.

Configure CRL autorefresh by using the CLI

At the command prompt, type the following commands to configure CRL auto refresh and verify the configuration:

```
set ssl crl <crlName> [-refresh ( ENABLED | DISABLED )] [-CAcert <
    string>] [-server <ip_addr|ipv6_addr|*> | -url <URL>] [-method (
    HTTP | LDAP )] [-port <port>] [-baseDN <string>] [-scope ( Base |
    One )] [-interval <interval>] [-day <positive_integer>] [-time <HH:
    MM>][-bindDN <string>] {
 -password  }
  [-binary ( YES | NO )]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

Example:


```
1     set CRL crl1  -refresh enabled -method ldap -inform DER -CAcert ca1
          -server 10.102.192.192 -port 389 -scope base -baseDN "cn=
         clnt_rsa4_multicert_der,ou=eng,o=ns,c=in" -time 00:01
2
3     set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80
          -time 00:10 -url http://10.102.192.192/crl/ca1.crl
4
5
6     > sh crl
7
8     1)         Name: crl1        Status: Valid,     Days to expiration:
          355
9                  CRL Path: /var/netscaler/ssl/crl1
10                 Format: PEM      CAcert: ca1
11                 Refresh: ENABLED          Method: HTTP
12                 URL: http://10.102.192.192/crl/ca1.crl
                      Port:80
13                 Refresh Time: 00:10
14                 Last Update: Successful, Date:Tue Jul  6 14:38:13 2010
15    Done
16 <!--NeedCopy-->
```

Configure CRL autorefresh using LDAP or HTTP by using the GUI

1. Navigate to **Traffic Management > SSL > CRL**.
2. Open a CRL, and select **Enable CRL Auto Refresh**.

> **Note**
>
> If the new CRL has been refreshed in the external repository before its actual update time as specified by the **Last Update time** field of the CRL, you must do the following:
>
> Immediately refresh the CRL on the Citrix ADC appliance.

To view the last update time, select the CRL, and click **Details**.

Synchronize CRLs

The Citrix ADC appliance uses the most recently distributed CRL to prevent clients with revoked certificates from accessing secure resources.

If CRLs are updated often, the Citrix ADC appliance needs an automated mechanism to fetch the latest CRLs from the repository. You can configure the appliance to update CRLs automatically at a specified refresh interval.

The appliance maintains an internal list of CRLs that need to be updated at regular intervals. At these specified intervals, the appliance scans the list for CRLs that need to be updated. It then connects to the remote LDAP server or HTTP server, retrieves the latest CRLs, and then updates the local CRL list with the new CRLs.

> **Note:**
>
> If the CRL check is set to mandatory when the CA certificate is bound to the virtual server, and the initial CRL refresh fails, the following action is taken for connections:
>
> All client-authentication connections with the same issuer as the CRL are rejected as REVOKED until the CRL is successfully refreshed.

You can specify the interval at which the CRL refresh must be carried out. You can also specify the exact time.

Synchronize CRL autorefresh by using the CLI

At the command prompt, type the following command:

```
set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <
   HH:MM>]
<!--NeedCopy-->
```

Example:

```
set ssl crl CRL-1 -refresh ENABLE -interval MONTHLY -days 10 -time
   12:00
<!--NeedCopy-->
```

Synchronize CRL refresh by using the GUI

1. Navigate to **Traffic Management > SSL > CRL**.
2. Open a CRL, select **enable CRL Auto Refresh**, and specify the interval.

Perform client authentication by using a certificate revocation list

If a certificate revocation list (CRL) is present on a Citrix ADC appliance, a CRL check is performed regardless of whether performing the CRL check is set to mandatory or optional.

The success or failure of a handshake depends on a combination of the following factors:

- Rule for CRL check
- Rule for client certificate check
- State of the CRL configured for the CA certificate

The following table lists the results of the possible combinations for a handshake involving a revoked certificate.

Table 1. Result of a Handshake with a Client Using a Revoked Certificate

Rule for CRL Check	Rule for Client Certificate Check	State of the CRL Configured for the CA certificate	Result of a Handshake with a Revoked Certificate
Optional	Optional	Missing	Success
Optional	Mandatory	Missing	Success
Optional	Mandatory	Present	Failure
Mandatory	Optional	Missing	Success
Mandatory	Mandatory	Missing	Failure
Mandatory	Optional	Present	Success
Mandatory	Mandatory	Present	Failure
Optional/Mandatory	Optional	Expired	Success
Optional/Mandatory	Mandatory	Expired	Failure

Note:

- The CRL check is optional by default. To change from optional to mandatory or conversely, you must first unbind the certificate from the SSL virtual server, and then bind it again after changing the option.
- In the output of the `sh ssl vserver` command, OCSP check: optional implies that a CRL check is also optional. The CRL check settings are displayed in the output of the `sh ssl vserver` command only if the CRL check is set to mandatory. If the CRL check is set to optional, the CRL check details do not appear.

To configure CRL check by using the CLI

At the command prompt, type the following command:

```
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck (
    Mandatory | Optional ))]
sh ssl vserver
<!--NeedCopy-->
```

Example:


```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
> sh ssl vs v1

Advanced SSL configuration for VServer v1:

DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1
    .2: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Configure CRL check by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open an SSL virtual server.
2. Click in the **Certificates** section.
3. Select a certificate and, in the **OCSP and CRL Check** list, select **CRL Mandatory**.

Result of a handshake with a revoked or valid certificate

Rule for CRL check	Rule for client certificate check	State of the CRL configured for the CA certificate	Result of a handshake with a revoked certificate	Result of a handshake with a valid certificate
Mandatory	Mandatory	Present	Failure	Success
Mandatory	Mandatory	Expired	Failure	Failure
Mandatory	Mandatory	Missing	Failure	Failure
Mandatory	Mandatory	Undefined	Failure	Failure
Optional	Mandatory	Present	Failure	Success
Optional	Mandatory	Expired	Success	Success
Optional	Mandatory	Missing	Success	Success
Optional	Mandatory	Undefined	Success	Success
Mandatory	Optional	Present	Success	Success
Mandatory	Optional	Expired	Success	Success
Mandatory	Optional	Missing	Success	Success
Mandatory	Optional	Undefined	Success	Success
Optional	Optional	Present	Success	Success
Optional	Optional	Expired	Success	Success
Optional	Optional	Missing	Success	Success
Optional	Optional	Undefined	Success	Success

Monitor certificate status with OCSP

September 14, 2021

Online Certificate Status Protocol (OCSP) is an Internet protocol that is used to determine the status of a client SSL certificate. Citrix ADC appliances support OCSP as defined in RFC 2560. OCSP offers significant advantages over certificate revocation lists (CRLs) in terms of timely information. Up-to-date revocation status of a client certificate is especially useful in transactions involving large sums of money and high-value stock trades. It also uses fewer system and network resources. Citrix ADC implementation of OCSP includes request batching and response caching.

OCSP implementation

OCSP validation on a Citrix ADC appliance begins when the appliance receives a client certificate during an SSL handshake. To validate the certificate, the appliance creates an OCSP request and forwards it to the OCSP responder. To do so, the appliance uses a locally configured URL. The transaction is in a suspended state until the appliance evaluates the response from the server and determines whether to allow the transaction or reject it. If the response from the server is delayed beyond the configured time and no other responders are configured, the appliance allows the transaction or display an error, depending on whether the OCSP check was set to optional or mandatory, respectively.

The appliance supports batching of OCSP requests and caching of OCSP responses to reduce the load on the OCSP responder and provide faster responses.

OCSP request batching

Each time the appliance receives a client certificate, it sends a request to the OCSP responder. To help avoid overloading the OCSP responder, the appliance can query the status of more than one client certificate in the same request. For this feature to work efficiently, a timeout needs to be defined so that processing of a single certificate is not inordinately delayed while waiting to form a batch.

OCSP response caching

Caching of responses received from the OCSP responder enables faster responses to the clients and reduces the load on the OCSP responder. Upon receiving the revocation status of a client certificate from the OCSP responder, the appliance caches the response locally for a predefined length of time. When a client certificate is received during an SSL handshake, the appliance first checks its local cache for an entry for this certificate. If an entry is found that is still valid (within the cache timeout limit), it is evaluated and the client certificate is accepted or rejected. If a certificate is not found, the appliance sends a request to the OCSP responder and stores the response in its local cache for a configured length of time.

> **Note**: From release 12.1 build 49.x, the cache timeout limit is now increased to a maximum of 43200 minutes (30 days). Earlier the limit was 1440 minutes (one day). The increased limit helps reduce the lookups on the OCSP server and avoid any SSL/TLS connection failures in case the OCSP server is not reachable due to network or other problems.

OCSP responder configuration

Configuring OCSP involves adding an OCSP responder, binding the OCSP responder to a certification authority (CA) certificate, and binding the certificate to an SSL virtual server. If you need to bind a different certificate to an OCSP responder that has already been configured, you need to first unbind the responder and then bind the responder to a different certificate.

Add an OCSP responder by using the CLI

At the command prompt, type the following commands to configure OCSP and verify the configuration:

```
add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [ -batchingDepth <
    positive_integer>][-batchingDelay <positive_integer>] [-resptimeout
    <positive_integer>] [-responderCert <string> | -trustResponder] [-
    producedAtTimeSkew <positive_integer>][-signingCert <string>][-
    useNonce ( YES | NO )][ -insertClientCert( YES | NO )]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]
<!--NeedCopy-->
```

```
bind ssl vserver <vServerName>@ (-certkeyName <string> ( CA [-ocspCheck
     ( Mandatory | Optional )]))
<!--NeedCopy-->
```

```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Example:

```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/
    ocsp/" -cache ENABLED -cacheTimeout 30 -batchingDepth 8 -
    batchingDelay 100 -resptimeout 100 -responderCert responder_cert -
    producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
<!--NeedCopy-->
```

```
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
<!--NeedCopy-->
```

```
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory
<!--NeedCopy-->
```

```
sh ocspResponder ocsp_responder1

    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
```



```
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done
<!--NeedCopy-->
```

```
show certkey ca_cert

    Name: ca_cert     Status: Valid,   Days to expiration:8907
    Version: 3
    …

    1)  VServer name: vs1      CA Certificate
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
Done
<!--NeedCopy-->
```

```
sh ssl vs vs1

    Advanced SSL configuration for VServer vs1:
    DH: DISABLED
    …

    1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
    1) Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
    Done
<!--NeedCopy-->
```

Modify an OCSP responder by using the CLI

You cannot modify the responder name. All other parameters can be changed using the `set ssl ocspResponder` command.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>] [-batchingDepth <
    positive_integer>] [-batchingDelay <positive_integer>] [-resptimeout
```



```
      <positive_integer>] [ -responderCert <string> | -trustResponder][-
      producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
      useNonce ( YES | NO )]
2
3 unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]
4
5 bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
     positive_integer>]
6
7 show ssl ocspResponder [<name>]
8 <!--NeedCopy-->
```

Configure an OCSP responder by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **OCSP Responder**, and configure an OCSP responder.
2. Navigate to **Traffic Management** > **SSL** > **Certificates**, select a certificate, and in the **Action** list, select **OCSP Bindings**. Bind an OCSP responder.
3. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, open a virtual server, and click in the Certificates section to bind a CA certificate.
4. Optionally, select **select OCSP Mandatory**.

OCSP stapling

September 14, 2021

The Citrix ADC implementation of CRL and OCSP reports the revocation status of client certificates only. To check the revocation status of a server certificate received during an SSL handshake, a client must send a request to a certificate authority.

For websites with heavy traffic, many clients receive the same server certificate. If each client sent a query for the revocation status of the server certificate, the certificate authority would be inundated with OCSP requests to check the validity of the certificate.

OCSP stapling solution

To avoid unnecessary congestion, the Citrix ADC appliance now supports OCSP stapling. That is, the appliance can now send the revocation status of a server certificate to a client, at the time of the SSL handshake, after validating the certificate status from an OCSP responder. The revocation status of a server certificate is “stapled” to the response the appliance sends to the client as part of the SSL

handshake. To use the OCSP stapling feature, you must enable it on an SSL virtual server and add an OCSP responder on the appliance.

Note:

- Citrix ADC appliances support OCSP stapling as defined in RFC 6066.
- OCSP stapling is supported only on the front-end of Citrix ADC appliances.

> **Important:**
>
> Citrix ADC support for OCSP stapling is limited to handshakes using TLS protocol version 1.0 or higher.

OCSP response caching of server certificates

During the SSL handshake, when a client requests the revocation status of the server certificate, the appliance first checks its local cache for an entry for this certificate. If a valid entry is found, it is evaluated and the server certificate and its status are presented to the client. If a revocation status entry is not found, the appliance sends a request for the revocation status of the server certificate to the OCSP responder. If it receives a response, it sends the certificate and the revocation status to the client. If the next update field is present in the OCSP response, the response is cached for the configured length of time (value specified in the timeout field.)

> **Note**: From release 12.1 build 49.x, you can clear the cached response, of the server certificate, from the OCSP responder even before the timeout expires. Earlier, it was not possible to discard the cached status in the certificate-key pair until the configured timeout was over.

To clear the cached status by using the CLI, at the command prompt, type:

```
clear ssl certKey <certkey name> -ocspstaplingCache
<!--NeedCopy-->
```

Example:

```
clear ssl certKey s1 -ocspstaplingCache
<!--NeedCopy-->
```

To clear the cached status by using the GUI

1. In the GUI, navigate to **Traffic Management** > **SSL** > **Certificates** > **CA Certificates**.
2. In the details pane, select a certificate.
3. In the **Select Action** list, select **Clear**. When prompted to confirm, click **Yes**.

OCSP stapling configuration

Configuring OCSP stapling involves enabling the feature and configuring OCSP. To configure OCSP, you must add an OCSP responder, bind the OCSP responder to a CA certificate, and bind the certificate to an SSL virtual server.

> **Note:**
>
> OCSP responders with only HTTP based URL are supported.

Enable OCSP stapling by using the CLI

At the command prompt, type:

```
set ssl vserver <name> -ocspstapling [ENABLED | DISABLED]
<!--NeedCopy-->
```

Example:

```
set ssl vserver vip1 -ocspStapling ENABLED
Done

sh ssl vserver vip1

     Advanced SSL configuration for VServer vip1:
     DH: DISABLED
     DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA:
        ENABLED Refresh Count: 0
     Session Reuse: ENABLED Timeout: 120 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     ClearText Port: 0
     Client Auth: DISABLED
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: ENABLED
     OCSP Stapling: ENABLED
     SSLv2: DISABLED SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED
          TLSv1.2: ENABLED
     Push Encryption Trigger: Always
     Send Close-Notify: YES

     ECC Curve: P_256, P_384, P_224, P_521

    1) CertKey Name: server_certificate1 Server Certificate

```



```
26      1) Cipher Name: DEFAULT
27       Description: Default cipher list with encryption strength >= 128
            bit
28  Done
29  <!--NeedCopy-->
```

Note: If the default (enhanced) profile is enabled, use the `set ssl profile <profile name> -ocspStapling [ENABLED | DISABLED]` command to enable or disable OCSP.

Enable OCSP stapling by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Virtual Server**.
2. Open a virtual server and, in **SSL Parameters**, select **OCSP Stapling**.

OCSP configuration

An OCSP responder is added dynamically or manually to send OCSP stapling requests. An internal responder is dynamically added when you add a server certificate and its issuer certificate based on the OCSP URL in the server certificate. A manual OCSP responder is added from the CLI or GUI. To send an OCSP request for a server certificate, the Citrix ADC appliance selects an OCSP responder based on the priority assigned to it when binding it to an issuer certificate. If a responder fails to send an OCSP stapling request, the responder with the next highest priority is selected for sending the request. For example, if only one responder is manually configured and it fails and a dynamically bound responder exists, it is selected for sending the OCSP request.

If the OCSP URL is other than HTTP, an internal OCSP responder is not created.

> **Note**
>
> A manually added OCSP responder takes precedence over a dynamically added responder.

Difference between a manually created OCSP responder and an internally created OCSP responder

Manually created OCSP responder	Internally (dynamically) created OCSP responder
Created manually and explicitly bound to the issuer certificate with a priority.	Created and bound by default, while adding a server certificate and its issuer certificate (CA certificate). Name starts with “ns_internal_”.

Priority between 1 and 127 is reserved for a configured responder.	Priority is automatically assigned from 128 onwards.
URL and batching depth can be changed.	URL and batching depth cannot be changed.
Deleted directly.	Deleted only when you delete the server certificate or the CA certificate.
Can be bound to any CA certificate.	Bound by default to one CA certificate. Cannot be bound to any other CA certificate.
Saved in the configuration (ns.conf).	Add commands are not saved in the configuration. Only set commands are saved.
If you bind three OCSP responders to the same issuer certificate with priorities 1, 2, and 3 respectively, and later unbind priority 2, the other priorities are not affected.	Three OCSP responders are automatically bound to an issuer certificate with priorities 128, 129, and 130 respectively. If you remove the server certificate that was used to create a responder bound with priority 129, then that responder is deleted. Also, the priority for the next responder (priority 130) is automatically changed to 129.

> **Example of request handling**:

1. Add a virtual server (VIP1).
2. Add issuer certificate (CA1) and bind it to VIP1.
3. Add three certificates S1, S2, and S3. Internal responders resp1, resp2, and resp3 respectively are created by default.
4. Bind S3 to VIP1.
5. A request comes to VIP1. Responder resp3 is selected.

To create an internal OCSP responder dynamically, the appliance needs the following:

- Certificate of the issuer of the server certificate (usually the CA certificate).
- Certificate-key pair of the server certificate. This certificate must contain the OCSP URL provided by the certificate authority. The URL is used as the name of the dynamically added internal responder.

An internal OCSP responder has the same default values as a manually configured responder.

> **Note:**
>
> Caching is disabled by default on an internal responder. Use the `set ssl ocspResponder`

> command to enable caching.

Configure OCSP by using the CLI

At the command prompt, type the following commands to configure OCSP and verify the configuration:

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]

add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [-resptimeout <positive_integer
    >] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew
    <positive_integer>][-signingCert <string>][-useNonce ( YES | NO )][
    -insertClientCert ( YES | NO )]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Parameters:

httpMethod:

HTTP method used to send OCSP requests. For requests, less than 255 bytes long, you can configure the HTTP GET method for queries to an OCSP server. If you specify the GET method but the length is greater than 255 bytes, the appliance uses the default method (POST).

Possible values: GET, POST

Default value: POST

ocspUrlResolveTimeout:

Time, in milliseconds, to wait for an OCSP URL resolution. After this time elapses, the responder with the next higher priority is selected. If all the responders fail, an error message appears or the connection is dropped, depending on the settings on the virtual server.

Minimum value: 100

Maximum value: 2000

Example:


```
add ssl certkey root_ca1  - cert root_cacert.pem
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/
    ocsp/" -cache ENABLED -cacheTimeout 30  -resptimeout 100 -
    responderCert responder_cert -producedAtTimeSkew 300 -signingCert
    sign_cert  -insertClientCert YES
bind ssl certKey root_ca1 -ocspResponder ocsp_responder1 -priority 1
sh ocspResponder ocsp_responder1
    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done

show certkey root_ca1
    Name: root_ca1     Status: Valid,   Days to expiration:8907
    Version: 3
    …
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
    Done
<!--NeedCopy-->
```

Modify OCSP by using the CLI

You cannot modify the name of an OCSP responder, but you can use the `set ssl ocspResponder` command to change any of the other parameters.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>]  [-resptimeout <
    positive_integer>] [ -responderCert <string> | -trustResponder][-
    producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
    useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

```



```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Configure OCSP by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **OCSP Responder**, and configure an OCSP responder.
2. Navigate to **Traffic Management** > **SSL** > **Certificates**, select a certificate, and in the **Action** list, select **OCSP Bindings**. **Bind an OCSP responder**.
3. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, open a virtual server, and click in the Certificates section to bind a CA certificate.
4. Optionally, select **OCSP Mandatory**.

> **Note:**
>
> The insert client certificate parameter in the `add ssl ocspResponder` and the `set ssl ocspResponder` commands is no longer valid. That is, the parameter is ignored during configuration.

Ciphers available on the Citrix ADC appliances

September 14, 2021

Your Citrix ADC appliance ships with a predefined set of cipher groups. To use ciphers that are not part of the DEFAULT cipher group, you have to explicitly bind them to an SSL virtual server. You can also create a user-defined cipher group to bind to the SSL virtual server. For more information about creating a user-defined cipher group, see Configure user-defined cipher groups on the ADC appliance.

> **Notes**
>
> RC4 cipher is not included in the default cipher group on the Citrix ADC appliance. However, it is supported in the software on the N3-based appliances. RC4 encryption, including the handshake, is done in software.
>
> Citrix recommends that you do not use this cipher because it is considered insecure and deprecated by RFC 7465.
>
> Use the 'show hardware' command to identify whether your appliance has N3 chips.

```
sh hardware

```



```
Platform: NSMPX-22000 16*CPU+24*IX+12*E1K+2*E1K+4*CVM N3 2200100

Manufactured on: 8/19/2013

CPU: 2900MHZ

Host Id: 1006665862

Serial no: ENUK6298FT

Encoded serial no: ENUK6298FT
<!--NeedCopy-->
```

- To display information about the cipher suites bound by default at the front end (to a virtual server), type: `sh cipher DEFAULT`
- To display information about the cipher suites bound by default at the back end (to a service), type: `sh cipher DEFAULT_BACKEND`
- To display information about all the cipher groups (aliases) defined on the appliance, type: `sh cipher`
- To display information about all the cipher suites that are part of a specific cipher group, type: `sh cipher <alias name>`. For example, sh cipher ECDHE.

The following links list the cipher suites supported on different Citrix ADC platforms and on external hardware security modules (HSMs):

- **Citrix ADC MPX/SDX (N3) appliance**: Cipher support on a Citrix ADC MPX/SDX (N3) appliance
- **Citrix ADC MPX/SDX Intel Coleto appliance**: Cipher support on a Citrix ADC MPX/SDX Intel Coleto SSL chip based appliance
- **Citrix ADC VPX appliance**: Cipher support on a Citrix ADC VPX appliance
- **Citrix ADC MPX/SDX 14000 FIPS appliance**: Cipher support on a Citrix ADC MPX/SDX 14000 FIPS appliance
- **External HSM (Thales/Safenet)**: Cipher supported on an External HSM (Thales/Safenet)
- **Citrix ADC MPX/SDX (N2) appliance**: Cipher support on a Citrix ADC MPX/SDX (N2) appliance
- **Citrix ADC MPX 9700 FIPS appliance**: Cipher support on a Citrix ADC MPX 9700 FIPS with firmware 2.2
- **Citrix ADC VPX FIPS and MPX FIPS appliances**: Cipher support on Citrix ADC VPX FIPS and MPX FIPS certified appliances

> **Note:**
>
> For DTLS cipher support, see DTLS cipher support on Citrix ADC VPX, MPX, and SDX appliances.

Table1 - Support on virtual server/frontend service/internal service:

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
TLS 1.3	13.0 all builds	13.0 all builds	13.0 all builds	Not supported	Not supported	13.0 all builds
	12.1–50.x	12.1–50.x	12.1–50.x	Not supported	Not supported	12.1–50.x
TLS 1.1/1.2	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 all builds	11.1 all builds	11.1 all builds	11.1 all builds	11.1 all builds	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
	11.0 all builds	11.0 all builds	11.0 all builds	11.0 all builds	11.0 all builds	11.0–70.x (only on MPX 5900/8900)
	10.5 all builds	10.5 all builds	10.5–57.x	10.5 58.1108.e	10.5–59.1359.e	10.5–67.x, 10.5-63.47 (only on MPX 5900/8900)
ECDHE/DHE (Example TLS1-ECDHE-RSA-AES128-SHA)	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G
	11.1 all builds	11.1 all builds	11.1 all builds	11.1 all builds	11.1–51.x	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
	11.0 all builds	11.0 all builds	11.0 all builds			11.0–70.114 (only on MPX 5900/8900)

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	10.5–53.x	10.5–53.x	10.5 all builds	10.5–59.1306.e		10.5–67.x, 10.5-63.47 (only on MPX 5900/8900)
AES-GCM (Example TLS1.2-AES128-GCM-SHA256)	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 all builds	11.1 all builds	11.1 all builds	11.1–51.x (See note)	11.1–51.x (See note)	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
	11.0 all builds	11.0 all builds	11.0–66.x			11.0–70.114 (only on MPX 5900/8900)
	10.5–53.x	10.5–53.x				10.5–67.x, 10.5-63.47 (only on MPX 5900/8900)
SHA-2 Ciphers (Example TLS1.2-AES-128-SHA256)	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G
	11.1 all builds	11.1 all builds	11.1 all builds	11.1–52.x	11.1–52.x	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
	11.0 all builds	11.0 all builds	11.0–66.x			11.0–72.x, 11.0-70.114 (only on MPX 5900/8900)
	10.5–53.x	10.5–53.x				10.5–67.x, 10.5-63.47 (only on MPX 5900/8900)

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
ECDSA (Example TLS1-ECDHE-ECDSA-AES256-SHA)	Not supported	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	Not supported	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	Not supported	12.0 all builds	12.0—57.x	Not applicable	Not supported	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
		11.1 all builds				11.1–56.x, 11.1-54.126 (Only ECC curves P_256 and P_384 are supported.)
CHACHA20	Not supported	13.0 all builds	13.0 all builds	Not supported	Not supported	13.0 all builds
	Not supported	Not supported	12.1 all builds	Not supported	Not supported	12.1–49.x (only on MPX 5900/8900)
	Not supported	Not supported	12.0–56.x	Not supported	Not supported	Not supported

Table 2 - Support on backend services:

TLS 1.3 is not supported on the back end.

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
TLS 1.1/1.2	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G
	11.1 all builds	11.1 all builds	11.1 all builds	11.1 all builds	11.1 all builds	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
	11.0–50.x	11.0–50.x	11.0–66.x	11.0 all builds		11.0–70.119 (only on MPX 5900/8900)

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	10.5–59.x	10.5–59.x		10.5–58.1108.e	10.5–59.1359.e	10.5–67.x, 10.5-63.47 (only on MPX 5900/8900)
ECDHE/DHE (Example TLS1-ECDHE-RSA-AES128-SHA)	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	12.0 all builds	12.0 all builds	12.0–56.x	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 all builds	11.1 all builds		11.1 all builds	11.1–51.x	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
	11.0–50.x	11.0–50.x				11.0–70.119 (only on MPX 5900/8900)
	10.5–58.x	10.5–58.x		10.5–59.1306.e		10.5–67.x, 10.5-63.47 (only on MPX 5900/8900)
AES-GCM (Example TLS1.2-AES128-GCM-SHA256)	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 all builds	12.0 all builds	Not supported	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G
	11.1 all builds	11.1 all builds		11.1–51.x	11.1–51.x	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G
SHA-2 Ciphers (Example TLS1.2-AES-128-SHA256)	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	12.0 all builds	12.0 all builds	Not supported	12.0 all builds	12.0 all builds	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G
	11.1 all builds	11.1 all builds		11.1–52.x	11.1–52.x	11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
ECDSA (Example TLS1-ECDHE-ECDSA-AES256-SHA)	Not supported	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds	13.0 all builds
	Not supported	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds	12.1 all builds for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	Not supported	12.0 all builds	12.0–57.x	Not applicable	Not supported	12.0 all builds for MPX 5900/8900, 12.0-57.x for MPX 15000-50G, 12.0-60.x for MPX 26000-100G

Protocol/Platform	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX 9700* FIPS with firmware 2.2	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
		11.1–51.x		Not applicable		11.1–56.x for MPX 5900/8900 and MPX 15000-50G, 11.1-60.x for MPX 26000-100G (Only ECC curves P_256 and P_384 are supported.)
CHACHA20	Not supported	13.0 all builds	13.0 all builds	Not supported	Not supported	13.0 all builds
	Not supported	Not supported	12.1 all builds	Not supported	Not supported	12.1–49.x for MPX 5900/8900, 12.1-50.x for MPX 15000-50G and MPX 26000-100G
	Not supported	Not supported	12.0–56.x	Not supported	Not supported	Not supported

For the detailed list of ECDSA ciphers supported, see ECDSA Cipher Suites support.

> Note
>
> - TLS-Fallback_SCSV cipher suite is supported on all appliances from release 10.5 build 57.x
> - HTTP Strict Transport Security (HSTS) support is policy-based.

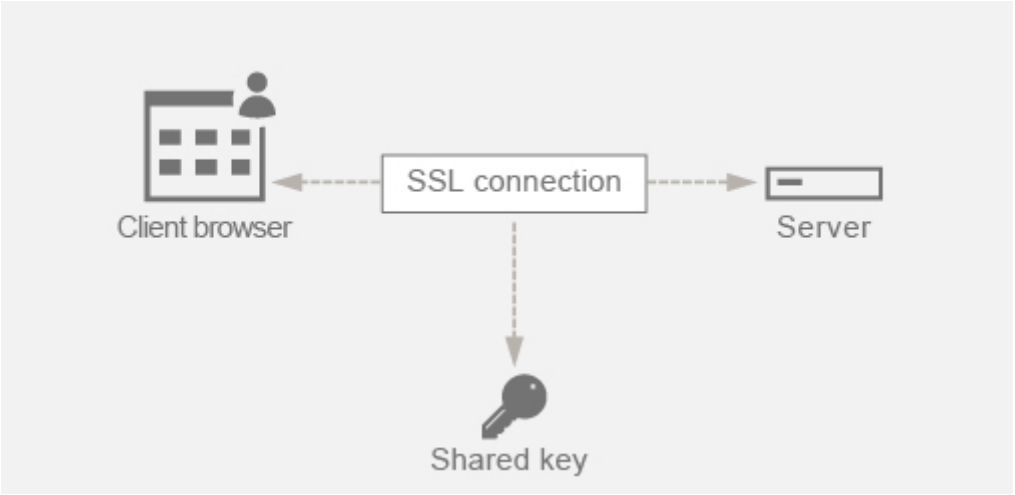
- All SHA-2 signed-certificates (SHA256, SHA384, SHA512) are supported on the front end of all appliances. In release 11.1 build 54.x and later, these certificates are also supported on the back-end of all appliances. In release 11.0 and earlier, only SHA256 signed-certificates are supported on the back end of all appliances.
- In release 11.1 build 52.x and earlier, the following ciphers are supported only on the front end of the MPX 9700 and MPX/SDX 14000 FIPS appliances:
  - TLS1.2-ECDHE-RSA-AES-256-SHA384
  - TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 From release 11.1 build 53.x, and in release 12.0, these ciphers are also supported on the back end.
- All ChaCha20-Poly1035 ciphers use a TLS pseudo random function (PSF) with the SHA-256 hash function.

Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy ensures protection of current SSL communications even if the session key of a web server is compromised at a later point in time.

Why do you need Perfect Forward Secrecy (PFS)?

An SSL connection is used to secure the data being passed between a client and a server. This connection begins with the SSL handshake that takes place between a client's browser and the contacted web server. It is during this handshake that the browser and the server exchange certain information to arrive upon a session key which serves as a means to encrypt the data throughout the rest of the communication.



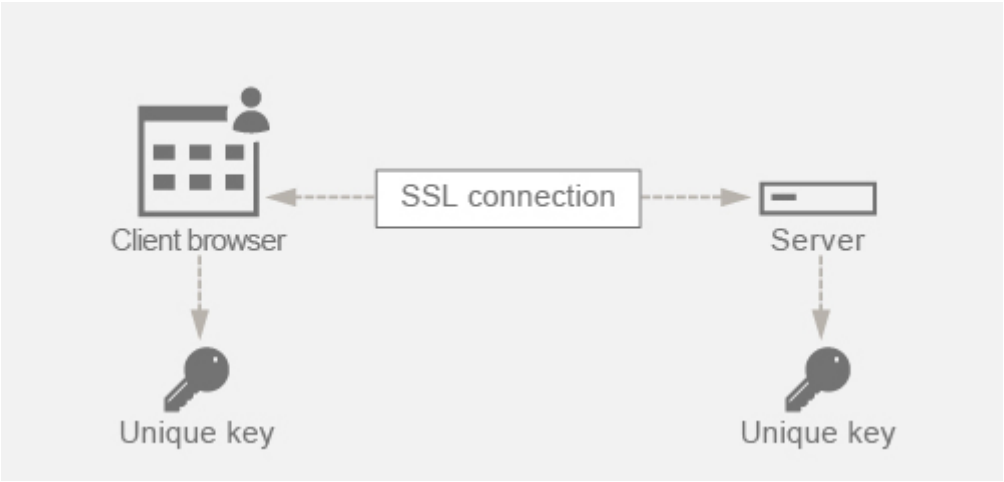
RSA is the most commonly used algorithm for key exchange. The browser uses the server's public key to encrypt and send across the pre-master secret to a server. This pre-master secret is used to arrive at the session key. The problem in the RSA key exchange approach is that if an attacker manages to

get hold of the server's private key at any point in time in the future, then the attacker gets hold of the pre-master secret using which the session key can be obtained. This session key can now be used by the attacker to decrypt all the SSL conversations. This means that your historical SSL communication was secure earlier but they are no longer secure as the server's stolen private key can be used to arrive at the session key and thus decrypt any saved historical conversation as well.

The need is to be able to protect the past SSL communication even if the server's private key has been compromised. This is where configuring Perfect Forward Secrecy (PFS) comes to the rescue.

How does PFS help?

Perfect Forward Secrecy (PFS) protects the past SSL communication by having the client and server agree upon a new key for each session and keeping the computation of this session key a secret. It works on the basis that compromise of a server key must not result in compromise of the session key. Session key is derived separately at both ends and is never transferred over the wire. The session keys are also destroyed once the communication is complete. These facts ensure that even if someone gets access to the server's private key, they would not be able to arrive at the session key and hence would not be able to decrypt the past data.



Explanation with example

Assume that we are using DHE for attaining PFS. The DH algorithm ensures that even though a hacker gets hold of the server's private key, the hacker will not be able to arrive at the session key because the session key and the random numbers (used to arrive at the session key) are kept secret at both ends and never exchanged over the wire.
PFS can be achieved by using the Ephemeral Diffie-Hellman key exchange which creates new temporary keys for each SSL session.

The flip side of creating a key for each session is that it requires extra computation but this can be overcome by using the Elliptic Curve which has smaller key sizes.

Configure PFS on Citrix ADC appliance

PFS can be configured on a Citrix ADC by configuring DHE or ECDHE ciphers. These ciphers ensure that the secret session key created is not shared on the wire (DH algorithm) and that the session key remains alive only for a short time (Ephemeral). Both the configurations are explained in the following sections.

Note: Using ECDHE ciphers instead of DHE makes the communication more secure with smaller key sizes.

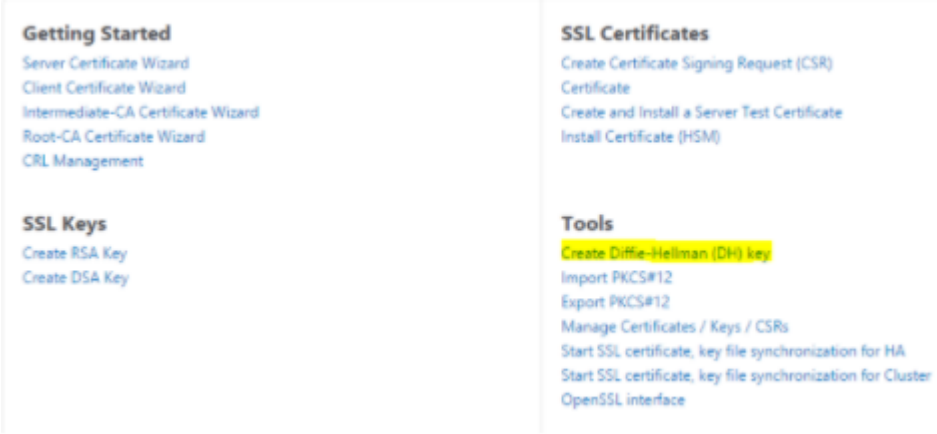
Configure DHE by using the GUI

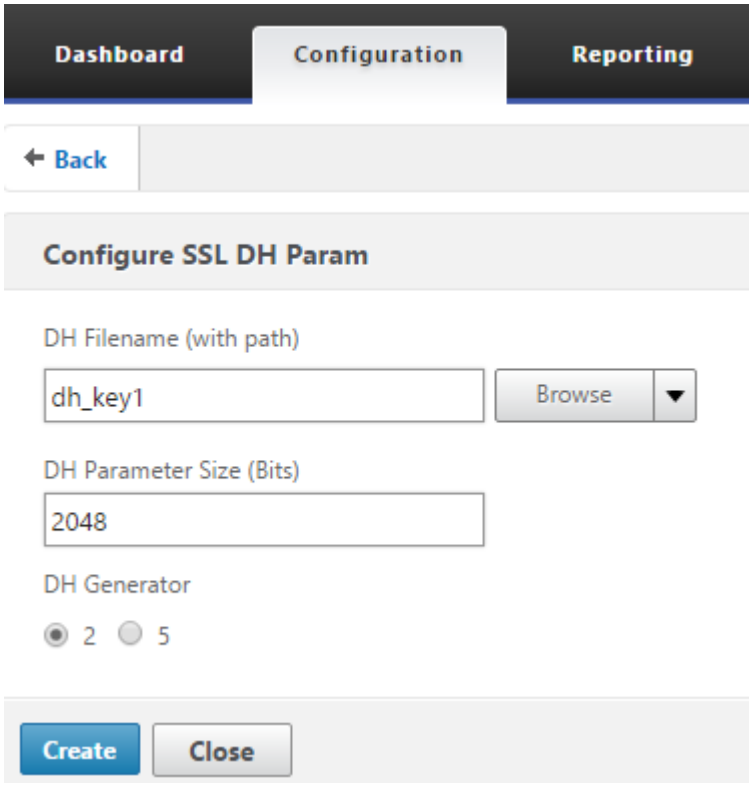
1. Generate a DH key.

   a. Navigate to **Traffic Management > SSL > Tools**.

   b. Click **Create Diffie Helman (DH) Key**.

   Note: Generating a 2048-bit DH key can take up to 30 minutes.



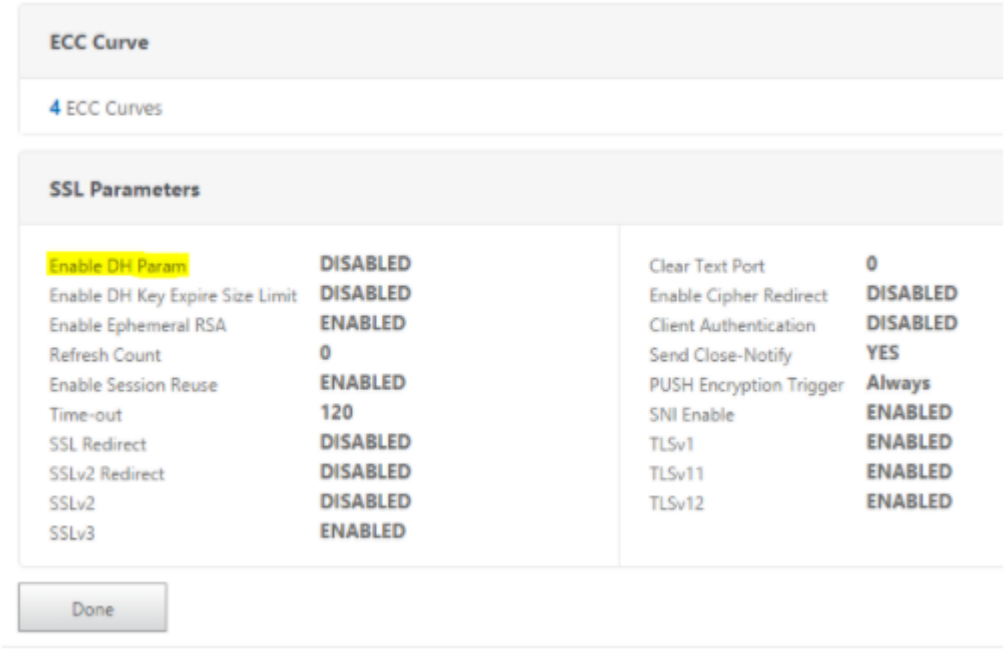


2. Enable DH Param for the SSL virtual server and attach the DH key to the SSL virtual server.

   a. Navigate to **Configuration > Traffic Management > Virtual Servers**.

   b. Select the virtual server on which you want to enable DH.

   c. Click **Edit**, click **SSL Parameters**, and click **Enable DH Param**.



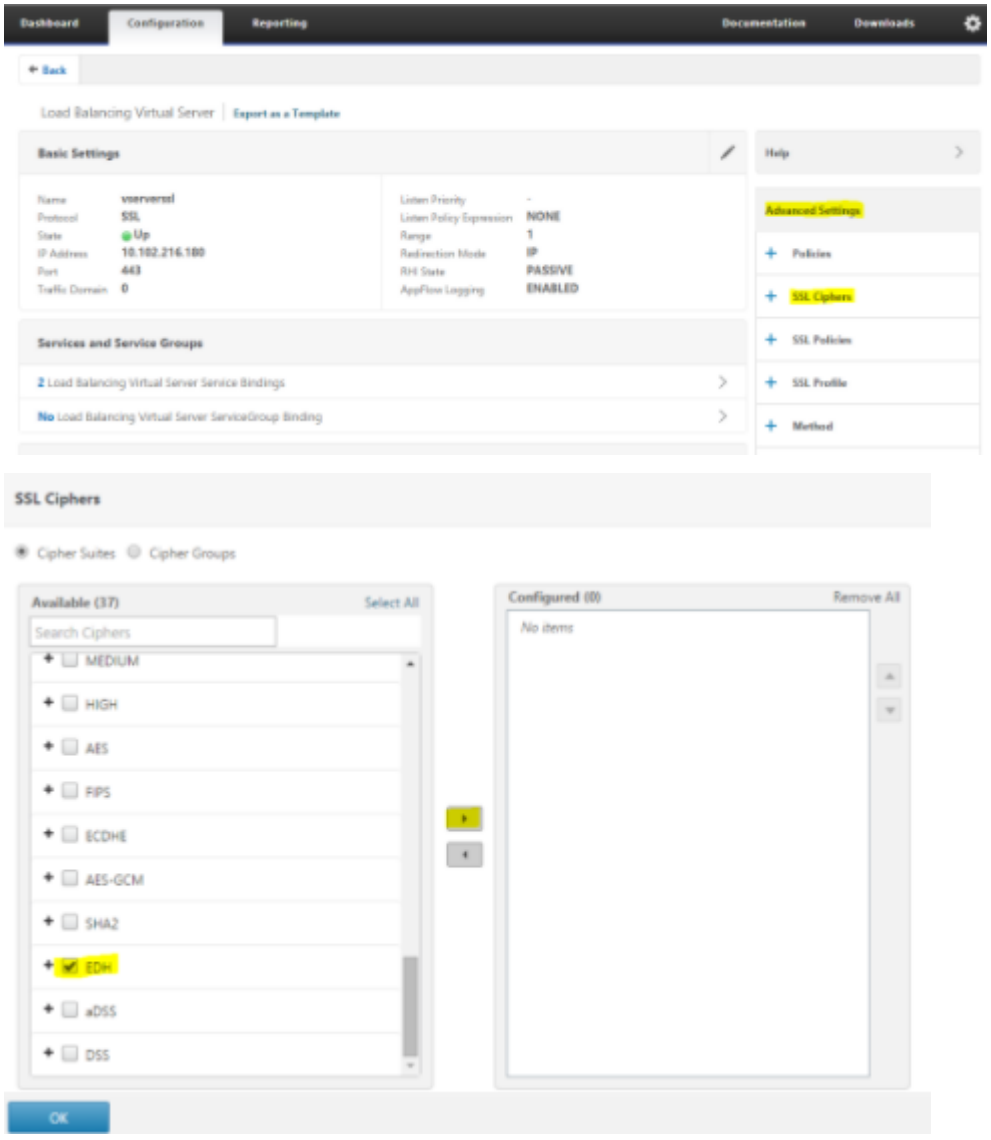
3. Bind the DHE ciphers to the virtual server.

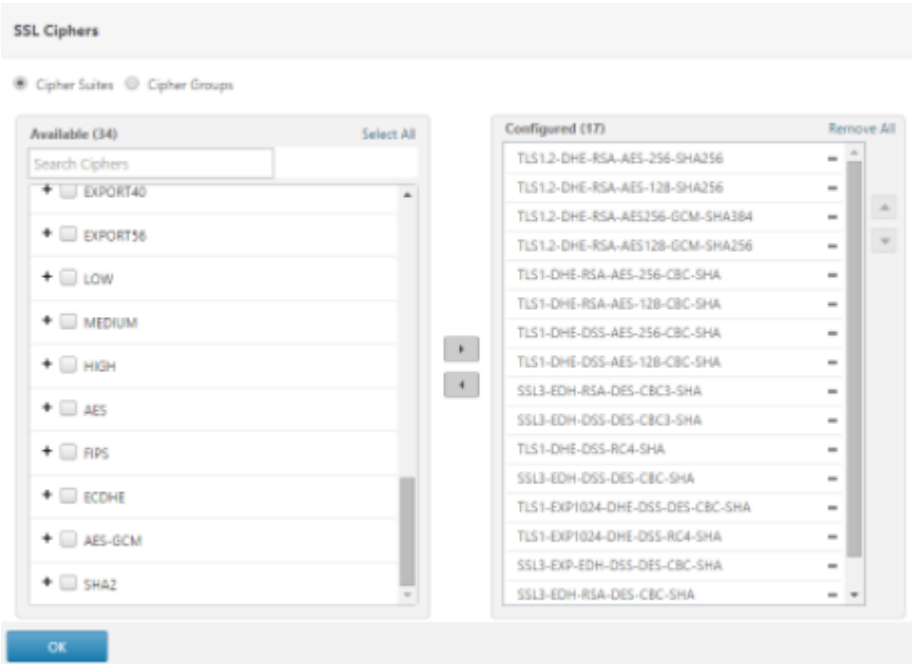
   a. Navigate to **Configuration > Traffic Management > Virtual Servers**.

   b. Select the virtual server on which you want to enable DH and click the pencil icon to edit.

   c. Under **Advanced Settings**, click plus icon next to **SSL Ciphers** and select the DHE cipher groups and click **OK** to bind.

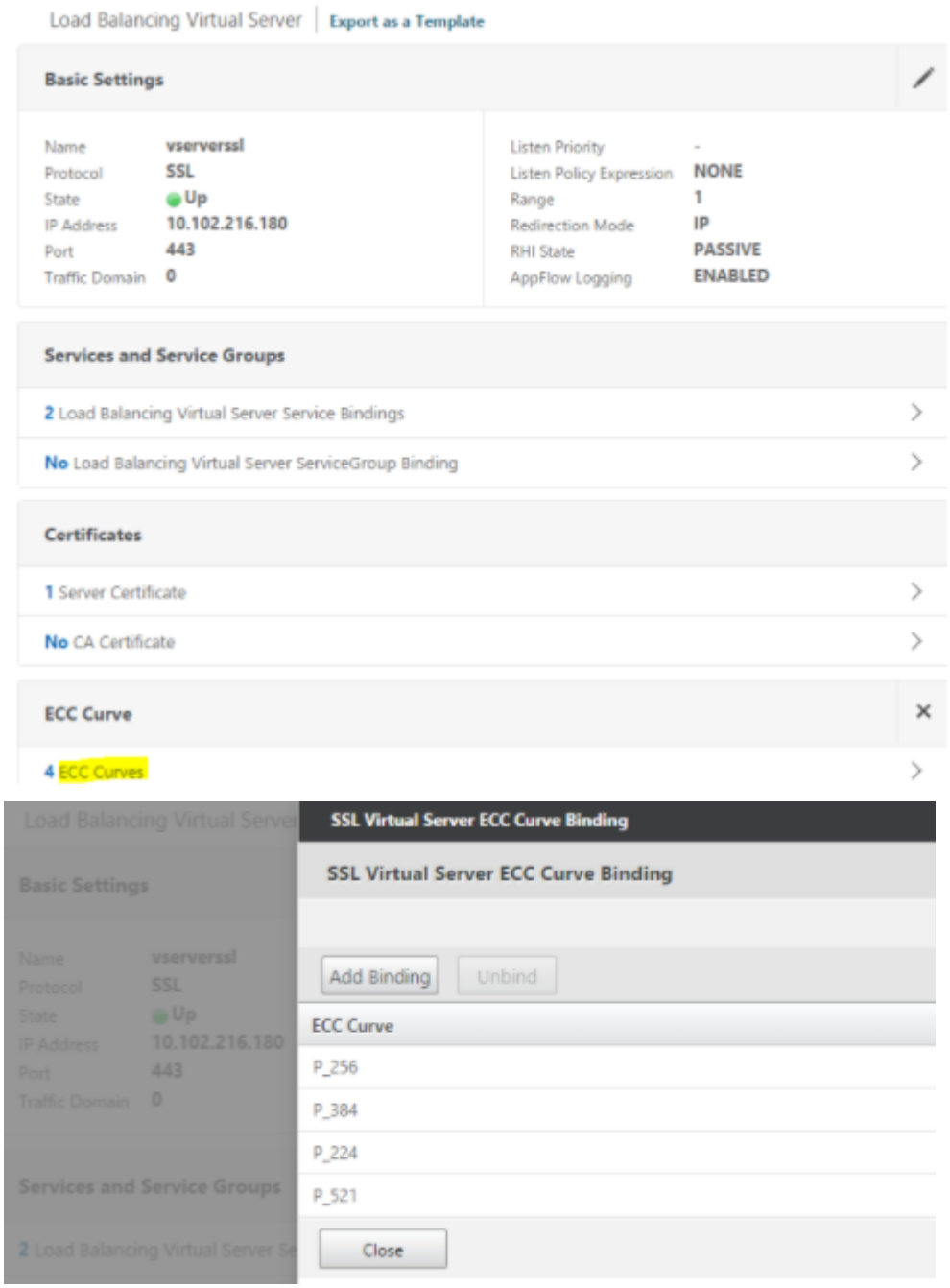
   Note: Ensure that the DHE ciphers are at the top of the cipher list bound to the virtual server.





Configure ECDHE by using the GUI

1. Bind the ECC curves to the SSL virtual server.
   a. Navigate to **Configuration > Traffic Management > Load Balancing > Virtual Servers**.
   b. Select the SSL virtual server which you want to edit, click **ECC Curve** and click **Add Binding**.
   c. Bind the required ECC curve to the virtual server.

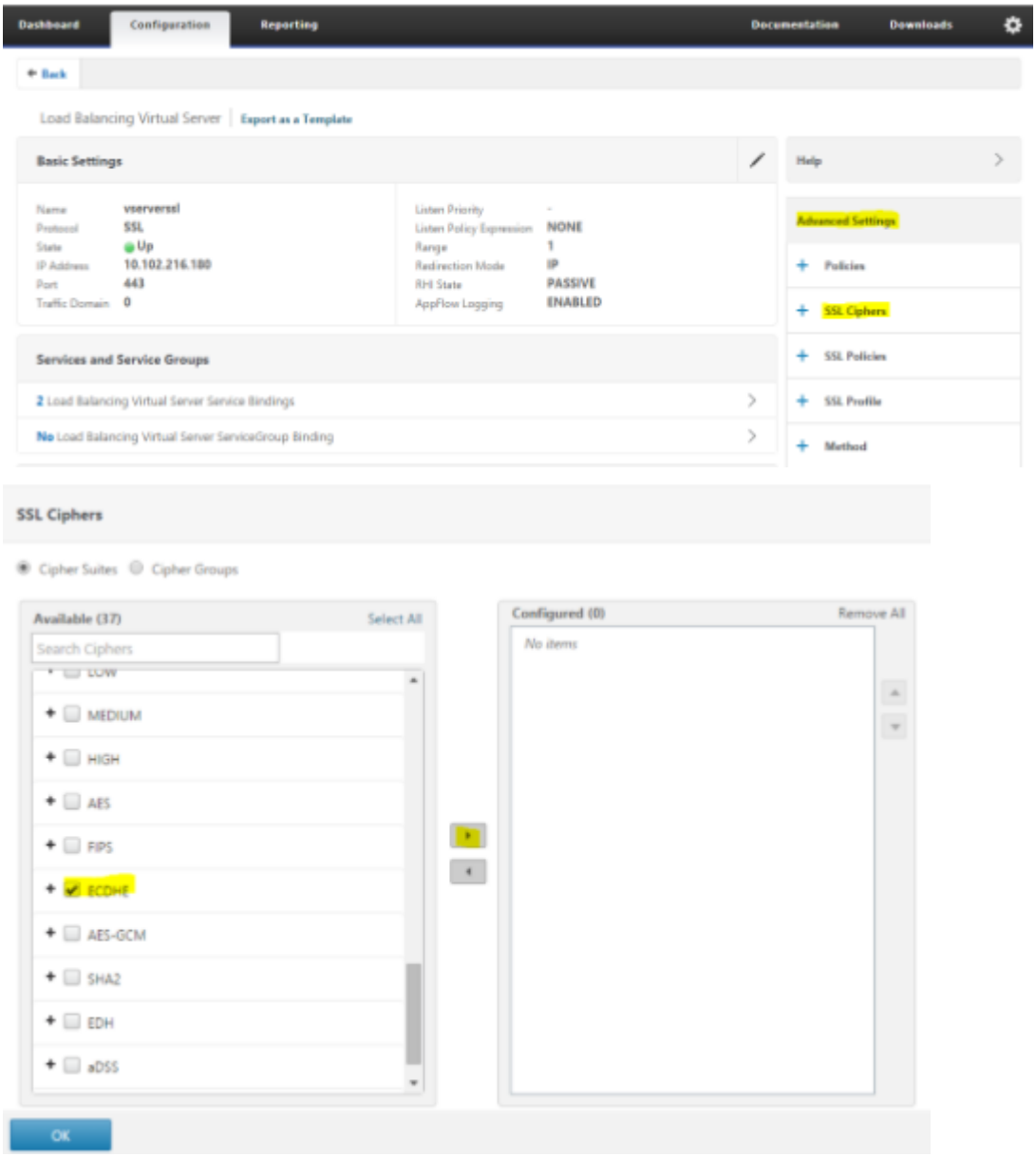


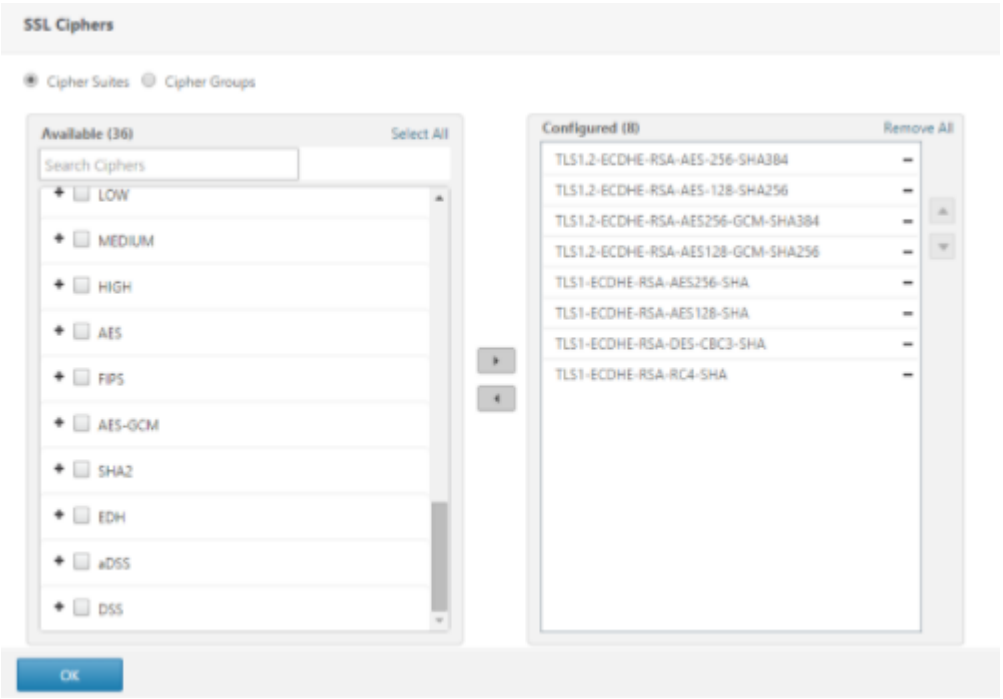
2. Bind the ECDHE ciphers to the virtual server.

   a. Navigate to **Configuration > Traffic Management > Virtual Servers** and select the virtual server on which you want to enable DH.

   b. Click **Edit > SSL Ciphers** and select the ECDHE cipher groups and click **Bind**.

   Note: Ensure that the ECDHE ciphers are at the top of the cipher list bound to the virtual server.





Note: For each case verify that the Citrix ADC appliance supports the ciphers you would like to use for the communication.

Configure PFS using an SSL profile

Note: Option to configure PFS (cipher or ECC) using an SSL profile is introduced from 11.0 64.x release onwards. Ignore the following section if on older versions.

To enable PFS using an SSL profile, a similar configuration (as explained in earlier configuration sections) needs to be done but on the SSL profile instead of directly configuring on a virtual server.

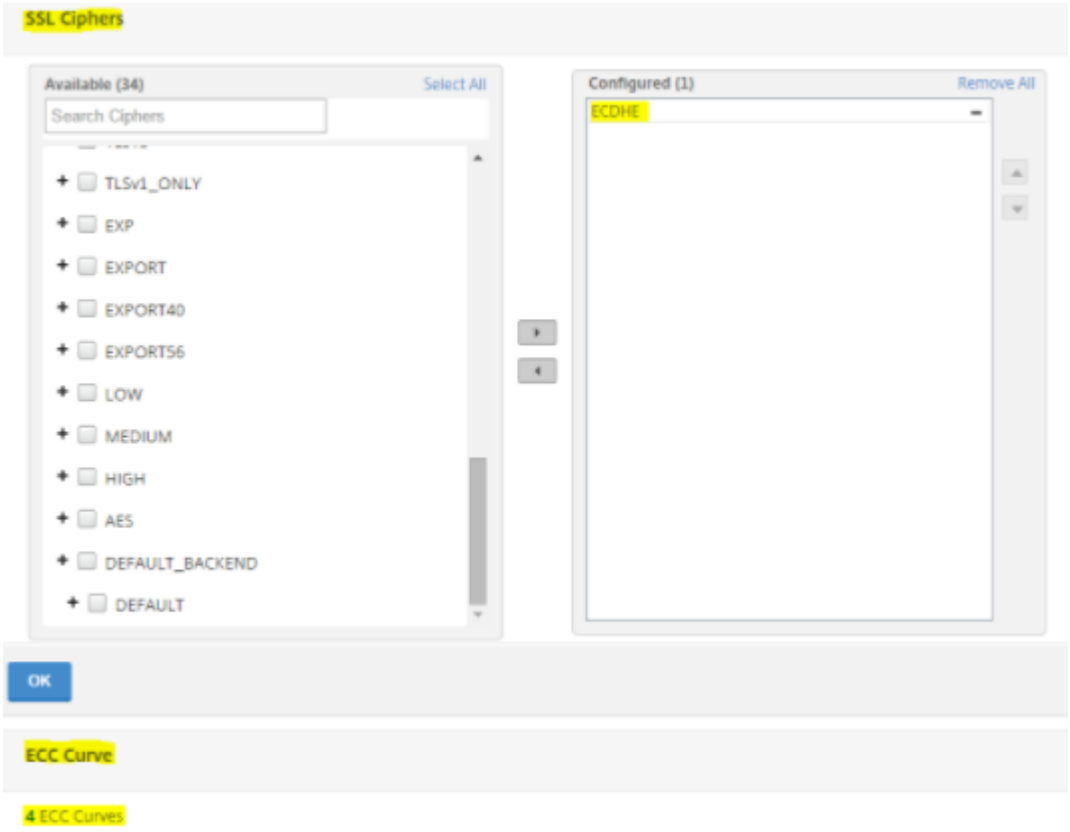
Configure PFS using an SSL profile by using the GUI

1. Bind the ECC curves and the ECDHE ciphers on the SSL profile.

   Note: ECC curves are already bound by default to all the SSL profiles.

   a. Navigate to **System > Profiles > SSL Profiles** and choose the profile you want to enable PFS on.

   b. Bind the ECDHE ciphers.



2. Bind the SSL profile to the virtual server.

   a. Go to **Configuration > Traffic Management > Virtual Servers** and select the virtual server.

   b. Click the pencil icon to edit the SSL profile.

   c. Click **OK** and click **Done**.

Configure PFS using SSL using the CLI

At the command prompt, type:

1. Bind ECC curves to the SSL profile.

```
bind sslprofile <SSLProfileName> -eccCurveName <Name_of_curve>
<!--NeedCopy-->
```


2. Bind the ECDHE cipher group.

```
bind sslprofile <SSLProfileName> cipherName <ciphergroupName>
<!--NeedCopy-->
```

3. Set the priority of the ECDHE cipher as 1.

```
set sslprofile <SSLProfileName> cipherName <ciphergroupName>
   cipherPriority <positive_integer>
<!--NeedCopy-->
```

4. Bind the SSL profile to the virtual server.

```
set SSL vserver <vservername> sslProfile <SSLProfileName>
<!--NeedCopy-->
```

ECDHE ciphers

September 14, 2021

All Citrix ADC appliances support the ECDHE cipher group on the front end and the back end. On an SDX appliance, if an SSL chip is assigned to a VPX instance, the cipher support of an MPX appliance applies. Otherwise, the normal cipher support of a VPX instance applies.

For more information about the builds and platforms that support these ciphers, see Ciphers available on the Citrix ADC appliances.

ECDHE cipher suites use elliptical curve cryptography (ECC). Because of its smaller key size, ECC is especially useful in a mobile (wireless) environment or an interactive voice response environment, where every millisecond is important. Smaller key sizes save power, memory, bandwidth, and computational cost.

A Citrix ADC appliance supports the following ECC curves:

- P_256
- P_384
- P_224
- P_521

Note: If you upgrade from a build earlier than release 10.1 build 121.10, you must explicitly bind ECC curves to your existing SSL virtual servers and services. The curves are bound by default to any virtual servers and services that you create after the upgrade.

You can bind an ECC curve to SSL front end and back end entities. By default all four curves are bound, in the following order: P_256, P_384,P_224,P_521. To change the order, you must first unbind all the curves, and then bind them in the desired order.

Bind ECC curves to an SSL virtual server by using the CLI

At the command prompt, type:

```
bind ssl vserver <vServerName > -eccCurveName <eccCurveName >
```

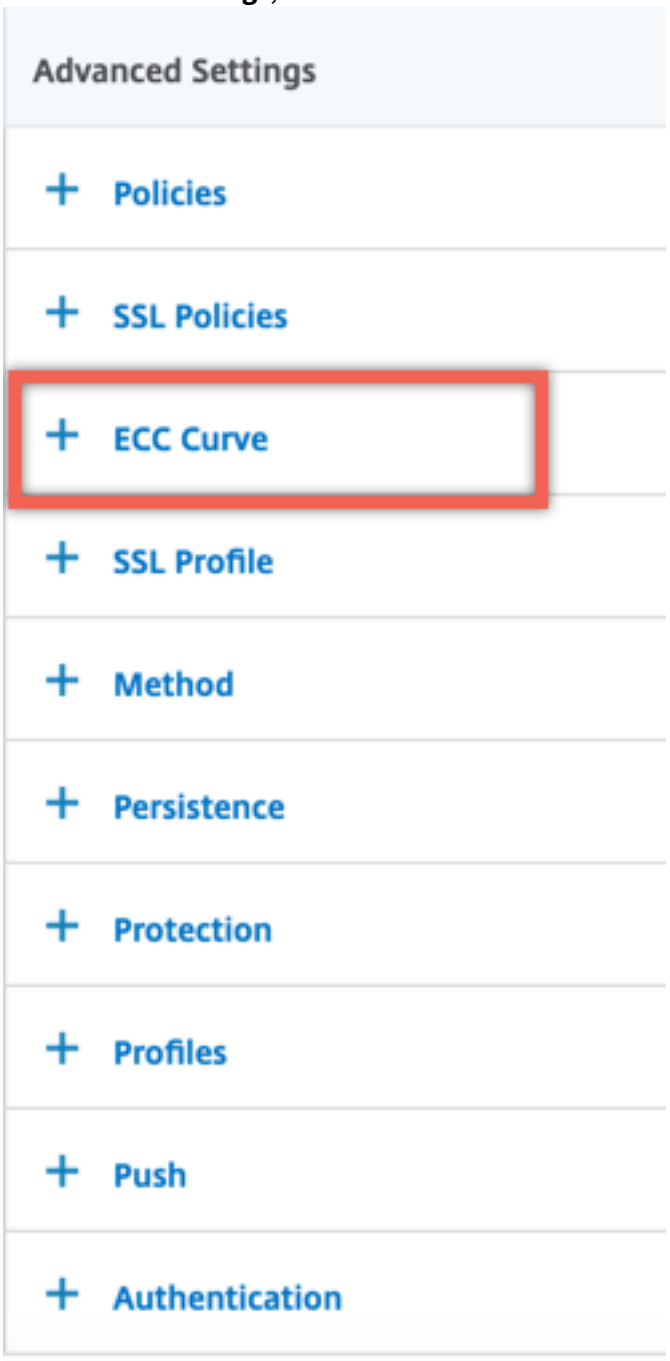
Example:

```
bind ssl vserver  v1 -eccCurveName P_224

sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED
   TLSv1.2: DISABLED
Push Encryption Trigger: Always
Send Close-Notify: YES
ECC Curve: P_224

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

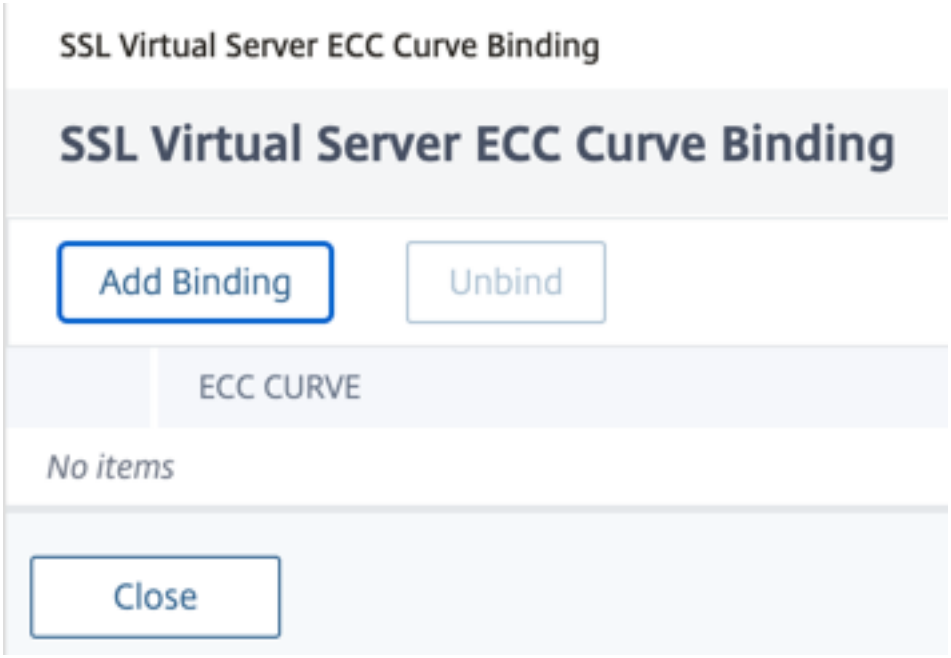
Bind ECC curves to an SSL virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Select an SSL virtual server and click **Edit**.

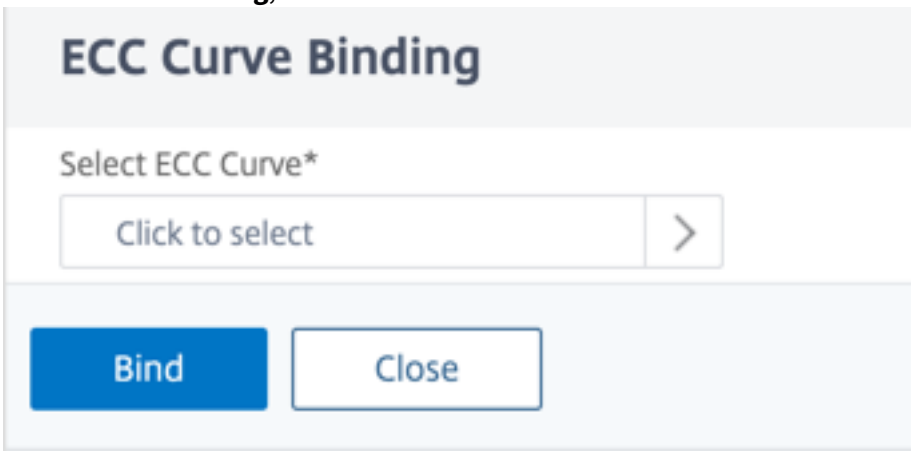
3. In **Advanced Settings**, click **ECC Curve**.



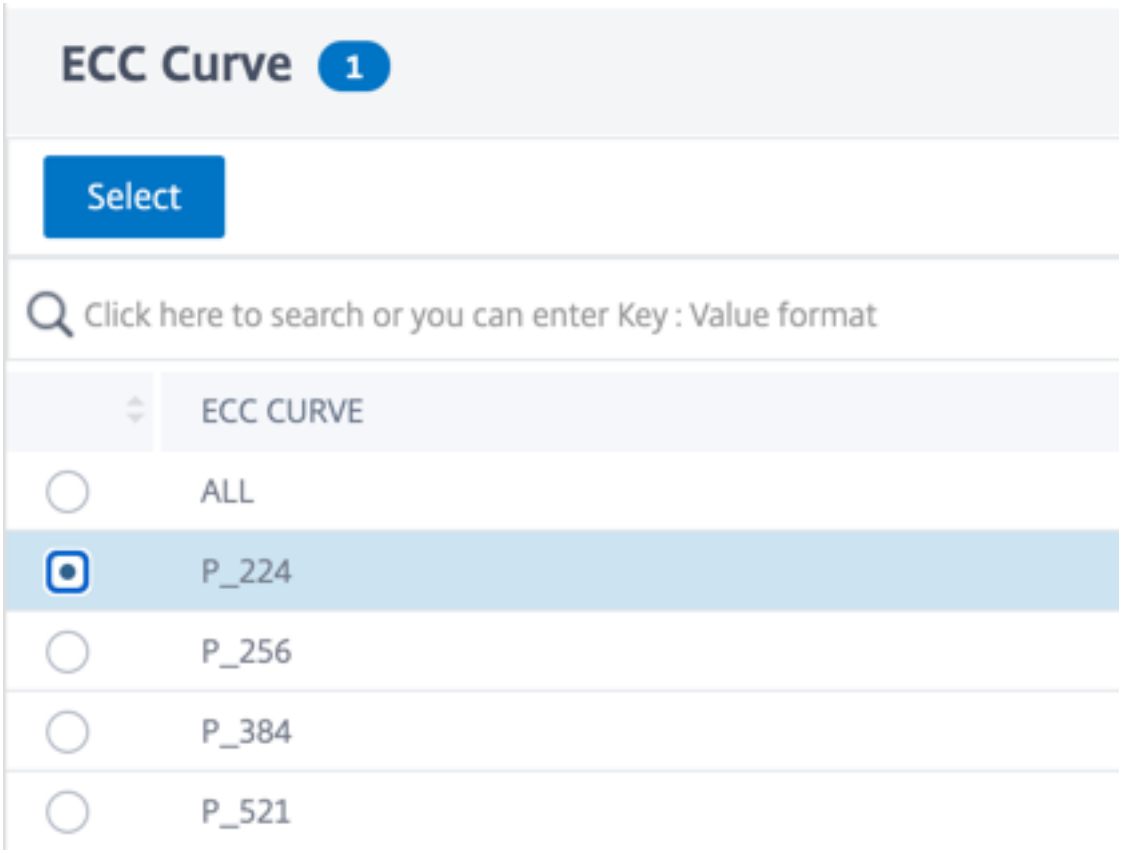
4. Click inside the ECC curve section.
5. In the **SSL Virtual Server ECC Curve Binding** page, click **Add Binding**.



6. In **ECC Curve Binding**, click **Select ECC Curve**.



7. Select a value, and then click **Select**.



8. Click **Bind**.
9. Click **Close**.
10. Click **Done**.

Bind ECC curves to an SSL service by using the CLI

At the command prompt, type:

```
bind ssl service <vServerName > -eccCurveName <eccCurveName >
```

Example:

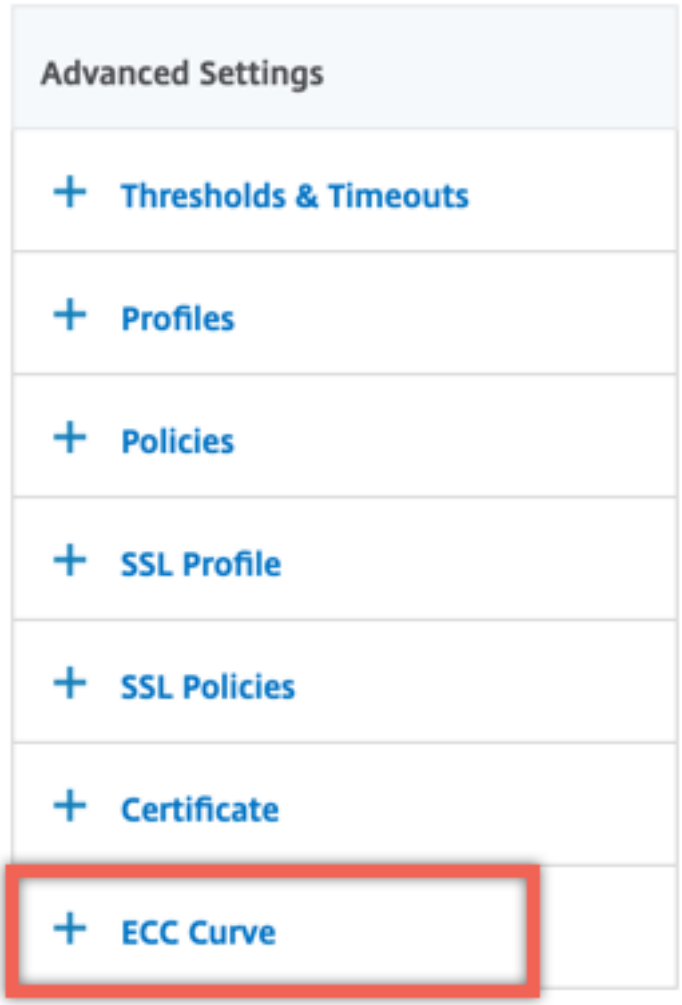
```
> bind ssl service sslsvc -eccCurveName P_224
 Done
> sh ssl service sslsvc

    Advanced SSL configuration for Back-end SSL Service sslsvc:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       DISABLED
    Session Reuse: ENABLED      Timeout: 300 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
```



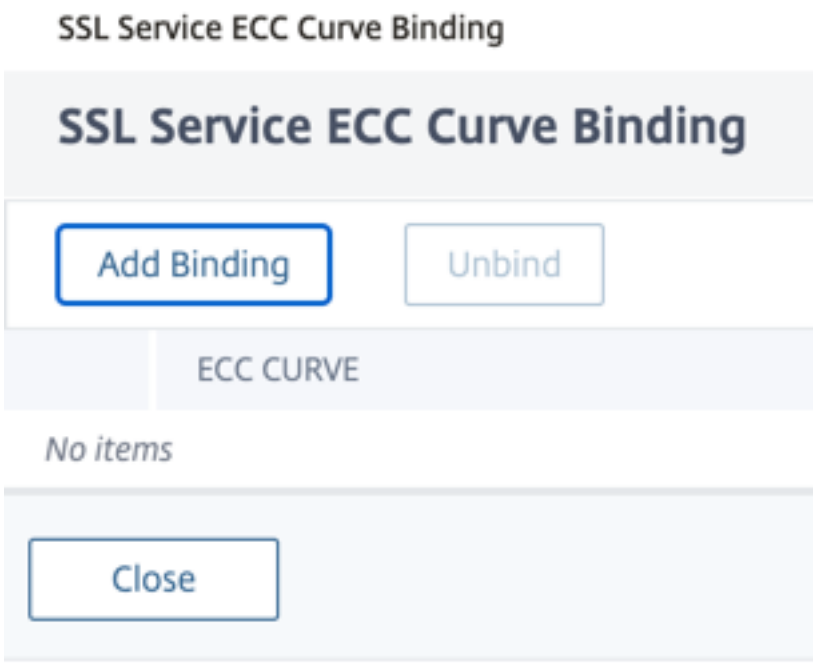
```
11      Server Auth: DISABLED
12      SSL Redirect: DISABLED
13      Non FIPS Ciphers: DISABLED
14      SNI: DISABLED
15      OCSP Stapling: DISABLED
16      SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
            ENABLED  TLSv1.3: DISABLED
17      Send Close-Notify: YES
18      Strict Sig-Digest Check: DISABLED
19      Zero RTT Early Data: ???
20      DHE Key Exchange With PSK: ???
21      Tickets Per Authentication Context: ???
22
23      ECC Curve: P_224
24
25
26  1)  Cipher Name: DEFAULT_BACKEND
27      Description: Default cipher list for Backend SSL session
28   Done
29  <!--NeedCopy-->
```

Bind ECC curves to an SSL service by using the GUI

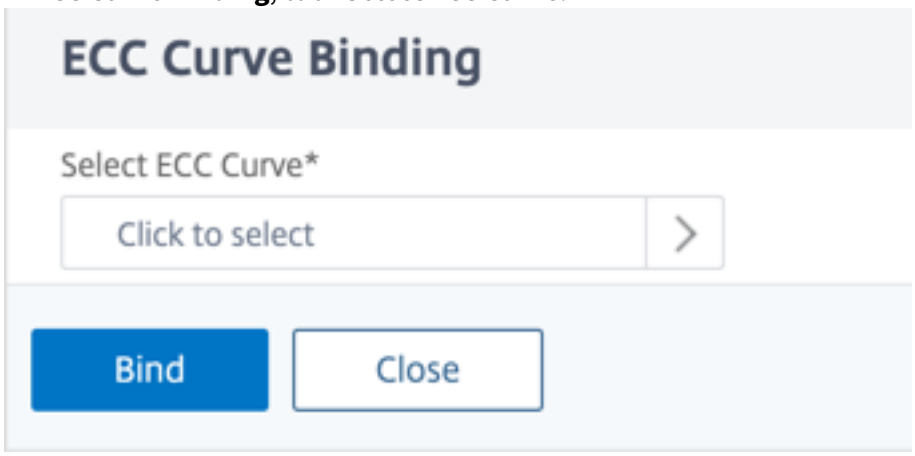
1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Select an SSL service and click **Edit**.
3. In **Advanced Settings**, click **ECC Curve**.



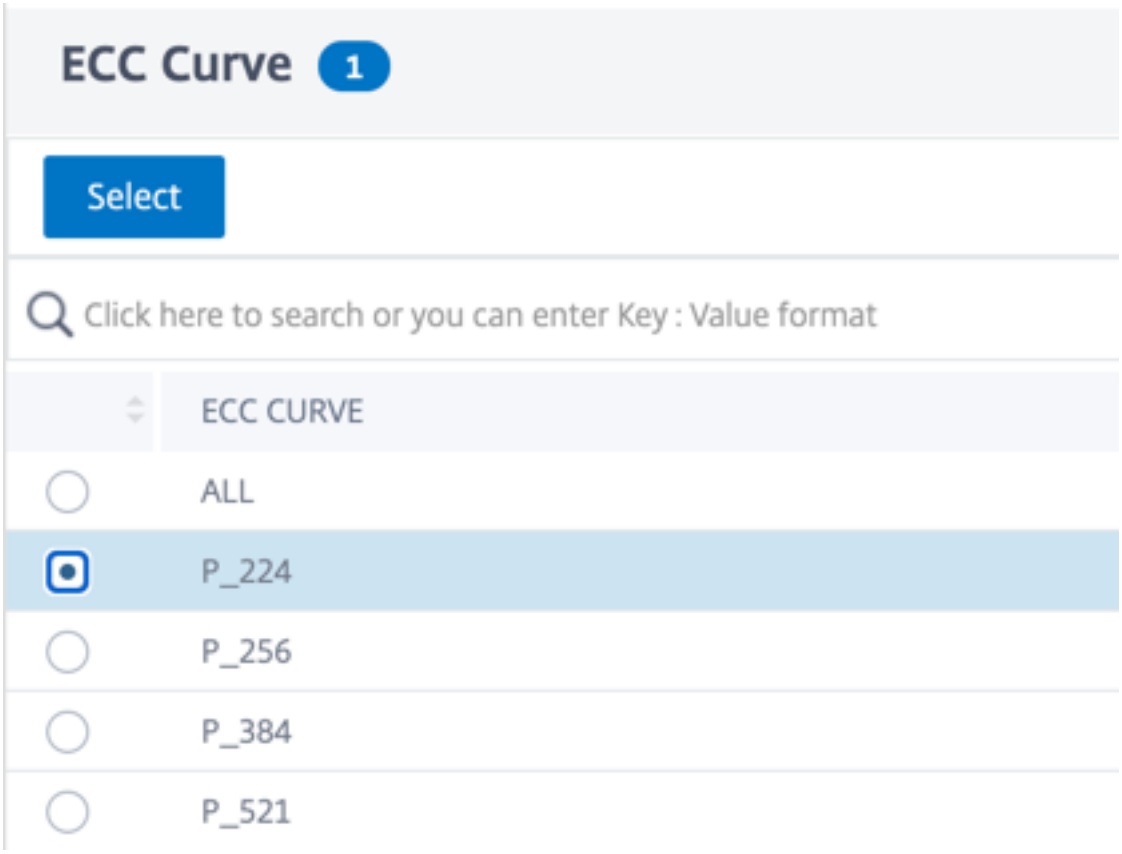
4. Click inside the ECC curve section.
5. In the **SSL Service ECC Curve Binding** page, click **Add Binding**.



6. In **ECC Curve Binding**, click **Select ECC Curve**.



7. Select a value, and then click **Select**.



8. Click **Bind**.
9. Click **Close**.
10. Click **Done**.

Diffie-Hellman parameters generation and achieving PFS with DHE

September 14, 2021

The Diffie-Hellman (DH) key exchange is a way for two parties involved in an SSL transaction to agree upon a shared secret over an insecure channel. These parties have no prior knowledge about each other. This secret can be converted into cryptographic keying material for symmetric key cipher algorithms that require such a key exchange.

This feature is disabled by default. Configured the feature to support ciphers that use DH as the key exchange algorithm.

> **Note**:
>
> Generating 2048-bit DH parameters might take a long time (up to 30 minutes).

Generate DH parameters by using the CLI

At the command prompt, type the following command:

```
create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]
<!--NeedCopy-->
```

Example:

```
create ssl dhparam Key-DH-1 512 -gen 2
<!--NeedCopy-->
```

Generate DH parameters by using the GUI

Navigate to **Traffic Management** > **SSL** and, in the **Tools** group, select **Create Diffie-Hellman (DH) key**, and **Configure SSL DH Param**.

> **Note:**
>
> For information about DH parameters, see Diffie-Hellman parameters.

Achieve perfect forward secrecy with DHE

Generating DH parameters is a CPU-intensive operation. In earlier releases, parameter generation, on a VPX appliance, took a long time because it was done in the software. Parameter generation is optimized by setting the `dhKeyExpSizeLimit` parameter. You can set this parameter for an SSL virtual server or an SSL profile and then bind the profile to a virtual server.

You can maintain perfect forward secrecy (PFS) on Citrix ADC MPX appliances by setting the DH count equal to zero. As a result, DH parameters are generated for each transaction (minimum `DHcount` is 0) on Citrix ADC MPX appliances. Thee parameters are generated without a significant drop in performance, because the operation is optimized. Earlier, the minimum DH count allowed was 500. That is, you cannot regenerate the key for up to 500 transactions.

On a Citrix ADC VPX appliance, you can generate DH parameters for every 500 transaction at the minimum (`DHcount` = 500). If you set `DHcount` equal to 0, then the DH parameters are not regenerated.

> **Limitation:**
>
> You cannot achieve PFS in VPX today with DH ciphers.

Optimize DH parameters generation by using the CLI

At the command prompt, type commands 1 and 2, or type command 3:


```
1. add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )] [-
   dhCount <positive_integer>] [-dh ( ENABLED | DISABLED) -dhFile <
   string>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED)]
2. set ssl vserver <vServerName> [-sslProfile <string>]
<!--NeedCopy-->
```

```
3. set ssl vserver <vServerName> [-dh ( ENABLED | DISABLED) -dhFile <
   string>] [-dhCount <positive_integer>] [-dhKeyExpSizeLimit ( ENABLED
    | DISABLED )]
<!--NeedCopy-->
```

Optimize DH parameters generation by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable DH Key Expire Size Limit**.

Cipher redirection

September 14, 2021

During the SSL handshake, the SSL client (usually a web browser) announces the suite of ciphers that it supports, in the configured order of cipher preference. From that list, the SSL server then selects a cipher that matches its own list of configured ciphers.

If the ciphers announced by the client does not match the ciphers configured on the SSL server, the SSL handshake fails. The failure is announced by a cryptic error message displayed in the browser. These messages rarely mention the exact cause of the error.

With cipher redirection, you can configure an SSL virtual server to deliver accurate, meaningful error messages when an SSL handshake fails. When an SSL handshake fails, the ADC appliance redirects the user to a previously configured URL or, if no URL is configured, displays an internally generated error page.

Configure cipher redirection by using the CLI

At the command prompt, type the following commands to configure cipher redirection and verify the configuration:

```
- set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED>
  -cipherURL < URL>
```



```
-  show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://
   redirectURl

Done

show ssl vserver vs-ssl

Advanced SSL configuration for VServer vs-ssl:
DH: DISABLED
Ephemeral RSA: ENABLED          Refresh Count: 1000
Session Reuse: ENABLED          Timeout: 600 seconds
Cipher Redirect: ENABLED        Redirect URL: http://redirectURl
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED
   TLSv1.2: ENABLED
    1)      CertKey Name: Auth-Cert-1       Server Certificate
    1)      Cipher Name: DEFAULT
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Configure cipher redirection by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable Cipher Redirect**, and specify a redirect URL.

Use hardware and software to improve ECDHE and ECDSA cipher performance

September 14, 2021

> **Note:**
>
> This enhancement is applicable only to the following platforms:
>
> - MPX/SDX 11000
> - MPX/SDX 14000
> - MPX 22000, MPX 24000, and MPX 25000
> - MPX/SDX 14000 FIPS

Previously, ECDHE and ECDSA computation on a Citrix ADC appliance was performed only on the hardware (Cavium chips), which limited the number of SSL sessions at any given time. With this enhancement, some operations are also performed in the software. That is, processing is done both on the Cavium chips and on the CPU cores to improve ECDHE and ECDSA cipher performance.

The processing is first performed in software, up to the configured software crypto threshold. After this threshold is reached, the operations are offloaded to the hardware. Therefore, this hybrid model uses both hardware and software to improve SSL performance. You can enable the hybrid model by setting the “softwareCryptoThreshold” parameter to suit your requirement. To disable the hybrid model, set this parameter to 0.

Benefits are greatest if the current CPU utilization is not too high, because the CPU threshold is not exclusive to ECDHE and ECDSA computation. For example, if the current workload on the appliance consumes 50% of the CPU cycles, and the threshold is set to 80%, ECDHE and ECDSA computation can only use 30%. After the configured software crypto threshold of 80% is reached, further ECDHE and ECDSA computation is offloaded to the hardware. In that case, actual CPU utilization might exceed 80%, because performing ECDHE and ECDSA computations in hardware consumes some CPU cycles.

Enable the hybrid model by using the CLI

At the command prompt, type:

```
set ssl parameter -softwareCryptoThreshold <positive_integer>

Synopsis:

softwareCryptoThreshold:

Citrix ADC CPU utilization threshold (as a percentage) beyond which
    crypto operations are not done in software. A value of zero implies
```



```
      that CPU is not utilized for doing crypto in software.

Default = 0

Min = 0

Max = 100
<!--NeedCopy-->
```

Example:

```
set ssl parameter - softwareCryptoThreshold 80
Done

show ssl parameter
Advanced SSL Parameters

SSL quantum size                    : 8 KB
Max CRL memory size                 : 256 MB
Strict CA checks                    : NO
Encryption trigger timeout          : 100 ms
Send Close-Notify                   : YES
Encryption trigger packet c         : 45
Deny SSL Renegotiation              : ALL
Subject/Issuer Name Insertion Format : Unicode
OCSP cache size                     : 10 MB
Push flag                           : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout     : 1 ms
Crypto Device Disable Limit         : 0
Global undef action for control policies : CLIENTAUTH
Global undef action for data policies : NOOP
Default profile                     : DISABLED
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services : NO
Software Crypto acceleration CPU Threshold : 80
Signature and Hash Algorithms supported by TLS1.2 : ALL
<!--NeedCopy-->
```

Enable the hybrid model by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**.
2. Enter a value for **Software Crypto Threshold (%)**.

Set an SNMP alarm for ECDHE exchange rate

ECDHE-based key exchange can cause the transactions per second on the appliance to drop. From release 13.0 build 52.x, you can configure an SNMP alarm for ECDHE-based transactions. In this alarm, you can set the threshold and normal limits for the ECDHE exchange rate. A new counter `nsssl_tot_sslInfo_ECDHE_Tx` is added. This counter is the sum of all the ECDHE-based transaction counters on the front-end and back-end of the appliance. When the ECDHE-based key exchange crosses the configured limits an SNMP trap is sent. Another trap is sent when the value is back to the configured normal value.

Set an SNMP alarm for ECDHE exchange rate using the CLI

At the command prompt, type:

```
set snmp alarm ECDHE-EXCHANGE-RATE -logging ( ENABLED | DISABLED ) -
    severity <severity>
 -state ( ENABLED | DISABLED ) -thresholdValue <positive_integer>  [-
     normalValue <positive_integer>] -time <secs>
<!--NeedCopy-->
```

Example:

```
set snmp alarm ECDHE-EXCHANGE-RATE -logging eNABLED -severity critical
    -state eNABLED -thresholdValue 100 -normalValue 50
<!--NeedCopy-->
```

ECDSA cipher suites support

September 14, 2021

ECDSA cipher suites use elliptical curve cryptography (ECC). Because of its smaller size, it is helpful in environments where processing power, storage space, bandwidth, and power consumption are constrained.

When the ECDHE_ECDSA cipher group is used, the server's certificate must contain an ECDSA-capable public key.

The following table lists the ECDSA ciphers that are supported on the Citrix ADC MPX and SDX appliances with N3 chips, Citrix ADC VPX appliances, MPX 5900/26000, and MPX/SDX 8900/15000 appliances.

Cipher Name	Priority	Description	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm	HexCode
TLS1-ECDHE-ECDSA-AES128-SHA	1	SSLv3	ECC-DHE	ECDSA	AES(128)	SHA1	0xc009
TLS1-ECDHE-ECDSA-AES256-SHA	2	SSLv3	ECC-DHE	ECDSA	AES(256)	SHA1	0xc00a
TLS1.2-ECDHE-ECDSA-AES128-SHA256	3	TLSv1.2	ECC-DHE	ECDSA	AES(128)	SHA-256	0xc023
TLS1.2-ECDHE-ECDSA-AES256-SHA384	4	TLSv1.2	ECC-DHE	ECDSA	AES(256)	SHA-384	0xc024
TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	5	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(128)	SHA-256	0xc02b
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	6	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(256)	SHA-384	0xc02c

Cipher Name	Priority	Description	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm	HexCode
TLS1-ECDHE-ECDSA-RC4-SHA	7	SSLv3	ECC-DHE	ECDSA	RC4(128)	SHA1	0xc007
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	8	SSLv3	ECC-DHE	ECDSA	3DES(168)	SHA1	0xc008
TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305	9	TLSv1.2	ECC-DHE	ECDSA	CHACHA20/	AEAD	0xcca9

ECDSA/RSA cipher and certificate selection

You can bind both ECDSA and RSA server certificates at the same time to an SSL virtual server. When both ECDSA and RSA certificates are bound to the virtual server, it automatically selects the appropriate server certificate to present to the client. If the client cipher list includes RSA ciphers, but does not include ECDSA ciphers, the virtual server presents the RSA server certificate. If both ciphers are present in the client's list, then the server certificate presented depends on the cipher priority set on the virtual server. That is, if RSA has a higher priority, the RSA certificate is presented. If ECDSA has a higher priority, the ECDSA certificate is presented to the client.

Client authentication by using an ECDSA or an RSA certificate

For client authentication, the CA certificate bound to the virtual server can be ECDSA or RSA signed. The appliance supports a mixed certificate chain. For example, the following certificate chain is supported.

Client certificate (ECDSA) <-> CA certificate (RSA) <-> Intermediate certificate (RSA) <-> Root certificate (RSA)

The following table shows the elliptical curves supported on the different Citrix ADC appliances with ECDSA cipher groups and ECDSA certificates:

Elliptical curves	Platforms supported
prime256v1	All platforms, including FIPS.
secp384r1	All platforms, including FIPS.
secp521r1	MPX 5900, MPX/SDX 8900, MPX/SDX 15000, MPX/SDX 26000, VPX
secp224r1	MPX 5900, MPX/SDX 8900. MPX/SDX 15000, MPX/SDX 26000, VPX

Create an ECDSA certificate-key pair

You can create an ECDSA certificate-key pair directly on a Citrix ADC appliance by using the CLI or the GUI. Earlier, you were able to install and bind an ECC certificate-key pair on the appliance, but you had to use OpenSSL to create a certificate-key pair.

Only P_256 and P_384 curves are supported.

> **Note**
>
> This support is available on all platforms except MPX 9700/1050/12500/15500.

To create an ECDSA certificate-key pair by using the CLI:

At the command prompt, type:

```
create ssl ecdsaKey <keyFile> -curve ( P_256 | P_384 ) [-keyform ( DER | PEM )] [-des | -des3] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

Example:

```
create ecdsaKey ec_p256.ky -curve P_256 -pkcs8
Done
create ecdsaKey ec_p384.ky -curve P_384
Done
<!--NeedCopy-->
```


To create an ECDSA certificate-key pair by using the GUI:

1. Navigate to **Traffic Management > SSL > SSL Files > Keys** and click **Create ECDSA Key**.
2. To create a key in PKCS#8 format, select **PKCS8**.

Configure user-defined cipher groups on the ADC appliance

September 14, 2021

A cipher group is a set of cipher suites that you bind to an SSL virtual server, service, or service group on the Citrix ADC appliance. A cipher suite comprises a protocol, a key exchange (Kx) algorithm, an authentication (Au) algorithm, an encryption (Enc) algorithm, and a message authentication code (Mac) algorithm. Your appliance ships with a predefined set of cipher groups. When you create an SSL service or SSL service group, the ALL cipher group is automatically bound to it. However, when you create an SSL virtual server or a transparent SSL service, the DEFAULT cipher group is automatically bound to it. In addition, you can create a user-defined cipher group and bind it to an SSL virtual server, service, or service group.

Note: If your MPX appliance does not have any licenses, then only the EXPORT cipher is bound to your SSL virtual server, service, or service group.

To create a user-defined cipher group, first you create a cipher group and then you bind ciphers or cipher groups to this group. If you specify a cipher alias or a cipher group, all the ciphers in the cipher alias or group are added to the user-defined cipher group. You can also add individual ciphers (cipher suites) to a user-defined group. However, you cannot modify a predefined cipher group. Before removing a cipher group, unbind all the cipher suites in the group.

Binding a cipher group to an SSL virtual server, service, or service group, appends the ciphers to the existing ciphers that are bound to the entity. To bind a specific cipher group to the entity, you must first unbind the ciphers or cipher group that is bound to the entity. Then bind the specific cipher group to the entity. For example, to bind only the AES cipher group to an SSL service, you perform the following steps:

1. Unbind the default cipher group ALL that is bound by default to the service when the service is created.

```
unbind ssl service <service name> -cipherName ALL
<!--NeedCopy-->
```

2. Bind the AES cipher group to the service

```
bind ssl service <Service name> -cipherName AE
<!--NeedCopy-->
```


If you want to bind the cipher group DES in addition to AES, at the command prompt, type:

```
bind ssl service <service name> -cipherName DES
<!--NeedCopy-->
```

Note: The free Citrix ADC virtual appliance supports only the DH cipher group.

Configure a user-defined cipher group by using the CLI

At the command prompt, type the following commands to add a cipher group, or to add ciphers to a previously created group, and verify the settings:

```
add ssl cipher <cipherGroupName>
bind ssl cipher <cipherGroupName> -cipherName <cipherGroup/cipherName>
show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

Example:

```
add ssl cipher test

Done

bind ssl cipher test -cipherName ECDHE

Done

sh ssl cipher test

1)      Cipher Name: TLS1-ECDHE-RSA-AES256-SHA  Priority : 1
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1   HexCode
   =0xc014
2)      Cipher Name: TLS1-ECDHE-RSA-AES128-SHA  Priority : 2
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1   HexCode
   =0xc013
3)      Cipher Name: TLS1.2-ECDHE-RSA-AES-256-SHA384    Priority : 3
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA-384
   HexCode=0xc028
4)      Cipher Name: TLS1.2-ECDHE-RSA-AES-128-SHA256    Priority : 4
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA-256
   HexCode=0xc027
5)      Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 5
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256) Mac=AEAD
   HexCode=0xc030
6)      Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 6
```



```
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128) Mac=AEAD
    HexCode=0xc02f
7)      Cipher Name: TLS1-ECDHE-ECDSA-AES256-SHA        Priority : 7
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA1
    HexCode=0xc00a
8)      Cipher Name: TLS1-ECDHE-ECDSA-AES128-SHA        Priority : 8
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA1
    HexCode=0xc009
9)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-SHA384   Priority : 9
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA-384
    HexCode=0xc024
10)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-SHA256   Priority : 10
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA-256
    HexCode=0xc023
11)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
    Priority : 11
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256) Mac=AEAD
    HexCode=0xc02c
12)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
    Priority : 12
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128) Mac=AEAD
    HexCode=0xc02b
13)     Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA        Priority : 13
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1   HexCode
    =0xc012
14)     Cipher Name: TLS1-ECDHE-ECDSA-DES-CBC3-SHA      Priority : 14
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=3DES(168) Mac=SHA1
    HexCode=0xc008
15)     Cipher Name: TLS1-ECDHE-RSA-RC4-SHA     Priority : 15
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=RC4(128)  Mac=SHA1   HexCode
    =0xc011
16)     Cipher Name: TLS1-ECDHE-ECDSA-RC4-SHA   Priority : 16
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=RC4(128)  Mac=SHA1
    HexCode=0xc007
17)     Cipher Name: TLS1.2-ECDHE-RSA-CHACHA20-POLY1305 Priority : 17
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=CHACHA20/POLY1305(256) Mac
    =AEAD   HexCode=0xcca8
18)     Cipher Name: TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
    Priority : 18
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=CHACHA20/POLY1305(256)
    Mac=AEAD   HexCode=0xcca9
Done

bind ssl cipher test -cipherName TLS1-ECDHE-RSA-DES-CBC3-SHA
<!--NeedCopy-->
```


Unbind ciphers from a cipher group by using the CLI

At the command prompt, type the following commands to unbind ciphers from a user-defined cipher group, and verify the settings:

```
show ssl cipher <cipherGroupName>

unbind ssl cipher <cipherGroupName> -cipherName <string>

show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

Remove a cipher group by using the CLI

Note: You cannot remove a built-in cipher group. Before removing a user-defined cipher group, make sure that the cipher group is empty.

At the command prompt, type the following commands to remove a user-defined cipher group, and verify the configuration:

```
rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
show ssl cipher <cipherGroupName>

<!--NeedCopy-->
```

Example:

```
rm ssl cipher test Done

sh ssl cipher test ERROR: No such resource [cipherGroupName, test]
<!--NeedCopy-->
```

Configure a user-defined cipher group by using the GUI

1. Navigate to **Traffic Management > SSL > Cipher Groups**.
2. Click **Add**.
3. Specify a name for the cipher group.
4. Click **Add** to view the available ciphers and cipher groups.
5. Select a cipher or cipher group, and click the arrow button to add them.
6. Click **Create**.

7. Click **Close**.

To bind a cipher group to an SSL virtual server, service, or service group by using the CLI:

At the command prompt, type one of the following:

```
bind ssl vserver <vServerName> -cipherName <string>

bind ssl service <serviceName> -cipherName <string>

bind ssl serviceGroup <serviceGroupName> -cipherName <string>

<!--NeedCopy-->
```

Example:

```
bind ssl vserver ssl_vserver_test -cipherName test
Done

bind ssl service  nshttps -cipherName test
Done

bind ssl servicegroup  ssl_svc  -cipherName test
Done
<!--NeedCopy-->
```

To bind a cipher group to an SSL virtual server, service, or service group by using the GUI:

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.

   For service, replace virtual servers with services. For service groups, replace virtual servers with service groups.

   Open the virtual server, service, or service group.

2. In **Advanced Settings**, select **SSL Ciphers**.
3. Bind a cipher group to the virtual server, service, or service group.

Binding individual ciphers to an SSL virtual server or service

You can also bind individual ciphers, instead of a cipher group, to a virtual server or service.

To bind a cipher by using the CLI:

At the command prompt, type:

```
bind ssl vserver <vServerName> -cipherName <string>
bind ssl service <serviceName> -cipherName <string>
<!--NeedCopy-->
```


Example:

```
bind ssl vserver v1 -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done

bind ssl service sslsvc -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done
<!--NeedCopy-->
```

To bind a cipher to an SSL virtual server by using the GUI:

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Select an SSL virtual server and click **Edit**.
3. In **Advanced Settings**, select **SSL Ciphers**.
4. In **Cipher Suites**, select **Add**.
5. Search for the cipher in the available list and click the arrow to add it to the configured list.
6. Click **OK**.
7. Click **Done**.

To bind a cipher to an SSL service, repeat the preceding steps after replacing virtual server with service.

Server certificate support matrix on the ADC appliance

September 14, 2021

From release 13.0 build 41.x, the Citrix ADC appliance supports server certificate messages that are fragmented into more than one record so long as the total size is within 32 KB. Earlier, the maximum supported size was 16 KB and fragmentation was not supported.

The Citrix ADC appliance supports the following server certificates.

Table 1: Support on front-end (FE) and back-end (BE) service

Server certificate/-Platform	MPX/SDX (N2 CHIPS) FE	MPX/SDX (N2 CHIPS) BE	MPX/SDX (N3 CHIPS) FE	MPX/SDX (N3 CHIPS) BE	VPX FE	VPX BE
MD5	Y	Y	Y	Y	Y	Y
SHA1	Y	Y	Y	Y	Y	Y
SHA224	Y	Y	Y	Y	Y	Y
SHA256	Y	Y	Y	Y	Y	Y
SHA384	Y	Y	Y	Y	Y	Y

Server certificate/-Platform	MPX/SDX (N2 CHIPS) FE	MPX/SDX (N2 CHIPS) BE	MPX/SDX (N3 CHIPS) FE	MPX/SDX (N3 CHIPS) BE	VPX FE	VPX BE
SHA512	Y	Y	Y	Y	Y	Y
RSA Key	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits
DH Key	1024 bits and 2048 bits	1024 bits and 2048 bits	1024 bits and 2048 bits	1024 bits and 2048 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits

Server certificate/Platform	MPX 9700/10500/12500/15500/ FIPS with FW 2.2 FE	MPX 9700/10500/12500/15500/ FIPS with FW 2.2 BE	MPX/SDX 14030/14060/14080 FIPS FE	MPX/SDX 14030/14060/14080 FIPS BE
MD5	Y	Y	Y	Y
SHA1	Y	Y	Y	Y
SHA224	Y	Y	Y	Y
SHA256	Y	Y	Y	Y
SHA384	Y	Y	Y	Y
SHA512	Y	Y	Y	Y
RSA Key	2048 bits	2048 bits	2048 bits and 3072 bits	2048 bits and 3072 bits
DH Key	N	N	N	N

Note:

- In release 11.1 and earlier, a Citrix ADC appliance supports the following “signature algorithms” extensions in the back end client hello message: RSA-MD5, RSA-SHA1, and RSA-SHA256.
  Because the Citrix ADC appliance does not support SHA 384 and SHA 512 signature algorithms extensions, some servers, such as Windows IIS servers, reset the connection.
- Starting release 12.0, a Citrix ADC appliance supports all the signature_algorithms extensions.

Client authentication

September 14, 2021

In a typical SSL transaction, the client that is connecting to a server over a secure connection checks the validity of the server. To do so, it checks the server's certificate before initiating the SSL transaction. Sometimes, however, you might want to configure the server to authenticate the client that is connecting to it.

Note: From release 13.0 build 41.x, the Citrix ADC appliance supports certificate request messages that are fragmented into more than one record provided the total size is within 32 KB. Earlier, the maximum supported size was 16 KB and fragmentation was not supported.

With client authentication enabled on an SSL virtual server, the Citrix ADC appliance asks for the client certificate during the SSL handshake. The appliance checks the certificate presented by the client for normal constraints, such as the issuer signature and expiration date.

Note: For the appliance to verify issuer signatures, the certificate of the CA that issued the client certificate must be installed on the appliance and bound to the virtual server that the client is transacting with.

If the certificate is valid, the appliance allows the client to access all secure resources. But if the certificate is invalid, the appliance drops the client request during the SSL handshake.

The appliance verifies the client certificate by first forming a chain of certificates, starting with the client certificate, and ending with the root CA certificate for the client (for example, Verisign). The root CA certificate might contain one or more intermediate CA certificates (if the root CA does not directly issue the client certificate).

Before you enable client authentication on the Citrix ADC appliance, make sure that a valid client certificate is installed on the client. Then, enable client authentication for the virtual server that handles the transactions. Finally, bind the certificate of the CA that issued the client certificate to the virtual server on the appliance.

Note: A Citrix ADC MPX appliance supports a certificate-key pair size from 512 bits to 4096 bits. The certificate must be signed by using one of the following hash algorithms:

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

On an SDX appliance, if an SSL chip is assigned to a VPX instance, the certificate-key pair size support

of an MPX appliance applies. Otherwise, the normal certificate-key pair size support of a VPX instance applies.

A Citrix ADC virtual appliance (VPX instance) supports certificates of at least 512 bits, up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate
- 4096-bit certificate on the physical server

Note: From release 13.0 build 79.x, client authentication with 4096-bit RSA client certificate is supported during an SSL handshake on VPX platform.

Notes:

- For MPX FIPS limitations, see MPX FIPS limitations.
- For SDX FIPS limitations, see SDX FIPS limitations.

Provide the client certificate

Before you configure client authentication, a valid client certificate must be installed on the client. A client certificate includes details about the specific client system that creates secure sessions with the Citrix ADC appliance. Each client certificate is unique and must be used by only one client system.

Whether you obtain the client certificate from a CA, use an existing client certificate, or generate a client certificate on the Citrix ADC appliance, you must convert the certificate to the correct format. On the Citrix ADC appliance, certificates are stored in either the PEM or DER format and must be converted to PKCS#12 format before they are installed on the client system. After converting the certificate and transferring it to the client system, ensure that it is installed on that system and configured for the client application. The application, such as a web browser, must be part of the SSL transactions.

For instructions on how to convert a certificate from PEM or DER format to PKCS#12 format, see Import and convert SSL files.

For instructions on how to generate a client certificate, see Create a certificate.

Enable client-certificate based authentication

By default, client authentication is disabled on the Citrix ADC appliance, and all SSL transactions proceed without authenticating the client. You can configure client authentication to be either optional or mandatory as part of the SSL handshake.

If client authentication is optional, the appliance requests the client certificate but proceeds with the SSL transaction even if the client presents an invalid certificate. If client authentication is mandatory, the appliance terminates the SSL handshake if the SSL client does not provide a valid certificate.

Caution: Citrix recommends that you define proper access control policies before changing the client-certificate-based authentication check to optional.

Note: Client authentication is configured for individual SSL virtual servers, not globally.

Enable client-certificate based authentication by using the CLI

At the command prompt, type the following commands to enable the client-certificate-based authentication and verify the configuration:

```
set ssl vserver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-
    clientCert (MANDATORY | OPTIONAL)]
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
Done
show ssl vserver vssl

Advanced SSL configuration for VServer vssl:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.2: ENABLED TLSv1
    .2: ENABLED

1) CertKey Name: sslckey Server Certificate

1) Policy Name: client_cert_policy Priority: 0

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
```



```
28 Done
29 <!--NeedCopy-->
```

Enable client-certificate based authentication by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the SSL Parameters section, select Client Authentication, and in the Client Certificate list, select Mandatory.

> **Note:**
>
> If client authentication is set to mandatory and if the client certificate contains policy extensions, certificate validation fails. From release 12.0-56.x, you can set a parameter in the front-end SSL profile to skip this check. The parameter is disabled by default. That is, the check is performed by default.

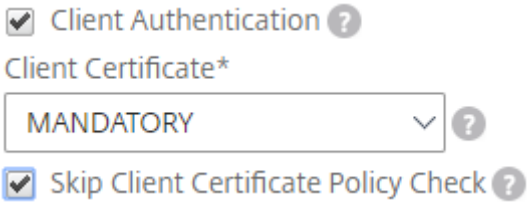
Skip the policy extension check during client authentication by using the CLI

At the command prompt, type:

```
1  set ssl profile ns_default_ssl_profile_frontend -clientauth ENABLED -
       skipClientCertPolicyCheck ENABLED
2
3  Parameter
4
5  skipClientCertPolicyCheck
6
7                Control policy extension check, if present inside the
                     X509 certificate chain. Applicable only if client
                     authentication is enabled and client certificate is
                     set to mandatory. Possible values functions as follows
                     :
8
9  - ENABLED: Skip the policy check during client authentication.
10
11 - DISABLED: Perform policy check during client authentication.
12
13 Possible values: ENABLED, DISABLED
14
15 Default: DISABLED
16 <!--NeedCopy-->
```


Skip the policy extension check during client authentication by using the GUI

1. Navigate to **System** > **Profiles** > **SSL Profiles**.
2. Create a new front-end profile or edit an existing front-end profile.
3. Verify that client authentication is enabled and client certificate is set to mandatory.
4. Select **Skip Client Certificate Policy Check**.



Bind CA certificates to the virtual server

A CA whose certificate is present on the Citrix ADC appliance must issue the client certificate used for client authentication. Bind this certificate to the Citrix ADC virtual server that carries out client authentication.

Bind the CA certificate to the SSL virtual server in such a way that the appliance can form a complete certificate chain when it verifies the client certificate. Otherwise, certificate chain formation fails and the client is denied access even if its certificate is valid.

You can bind CA certificates to the SSL virtual server in any order. The appliance forms the proper order during client certificate verification.

For example, if the client presents a certificate issued by **CA_A**, where **CA_A** is an intermediate CA whose certificate is issued by **CA_B**, whose certificate is in turn issued by a trusted root CA, **Root_CA**, a chain of certificates that contain all three of these certificates must be bound to the virtual server on the Citrix ADC appliance.

For instructions on binding one or more certificates to the virtual server, see Bind the certificate-key pair to the SSL virtual server.

For instructions on creating a chain of certificates, see Create a chain of certificates.

Stricter control on client certificate validation

The Citrix ADC appliance accepts valid Intermediate-CA certificates if a single Root-CA issues them. That is, if only the Root-CA certificate is bound to the virtual server, and that Root-CA valiadates any intermediate certificate sent with the client certificate, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, none of the intermediate certificates can be validated by using a CRL or OCSP responder unless that certificate is bound to the SSL

virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that is signed by one of the CA certificates bound to that virtual server. To do so, you must enable the **ClientAuthUseBoundCAChain** setting in the SSL profile bound to the virtual server. The handshake fails if one of the CA certificates bound to the virtual server has not signed the client certificate.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, the Citrix ADC appliance only accepts clientcert1.

Enable stricter control on client certificate validation by using the CLI

At the command prompt, type:

```
set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
<!--NeedCopy-->
```

Enable stricter control on client certificate validation by using the GUI

1. Navigate to **System** > **Profiles**, select the **SSL Profiles** tab, and create an SSL profile, or select an existing profile.
2. Select **Enable Client Authentication using bound CA Chain**.

Server authentication

September 14, 2021

Since the Citrix ADC appliance performs SSL offload and acceleration on behalf of a web server, the appliance does not usually authenticate the Web server's certificate. However, you can authenticate the server in deployments that require end-to-end SSL encryption.

In such a situation, the appliance becomes the SSL client and carries out a secure transaction with the SSL server. It verifies that a CA whose certificate is bound to the SSL service has signed the server certificate, and checks the validity of the server certificate.

To authenticate the server, enable server authentication and bind the certificate of the CA that signed the server's certificate to the SSL service on the ADC appliance. When binding the certificate, you must specify the bind as a CA option.

Enable (or disable) server certificate authentication

You can use the CLI and the GUI to enable and disable server certificate authentication.

Enable (or disable) server certificate authentication using the CLI

At the command prompt, type the following commands to enable server certificate authentication and verify the configuration:

```
set ssl service <serviceName> -serverAuth ( ENABLED | DISABLED )
show ssl service <serviceName>
<!--NeedCopy-->
```

Example:

```
set ssl service ssl-service-1 -serverAuth ENABLED
<!--NeedCopy-->
```

```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
               service-1:`
            DH: DISABLED
            Ephemeral RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            Server Auth: ENABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
    1)      Cipher Name: ALL
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Enable (or disable) server certificate authentication by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open an SSL service.
2. In the SSL Parameters section, select Enable Server Authentication, and specify a Common Name.
3. In Advanced Settings, select Certificates, and bind a CA certificate to the service.

Bind the CA certificate to the service by using the CLI

At the command prompt, type the following commands to bind the CA certificate to the service and verify the configuration:

```
bind ssl service <serviceName> -certkeyName <string> -CA

show ssl service <serviceName>
<!--NeedCopy-->
```

Example:

```
bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
<!--NeedCopy-->
```

```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
               service-1:
            DH: DISABLED
            Ephemeral RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            Server Auth: ENABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
    1)      CertKey Name: samplecertkey     CA Certificate
       CRLCheck: Optional
    1)      Cipher Name: ALL
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Configure a common name for server certificate authentication

In end-to-end encryption with server authentication enabled, you can include a common name in the configuration of an SSL service or service group. The name that you specify is compared to the common name in the server certificate during an SSL handshake. If the two names match, the handshake is successful.

If the common names do not match, the common name specified for the service or service group is compared to the values in the subject alternative name (SAN) field in the certificate. If it matches one

of those values, the handshake is successful. This configuration is especially useful if there are, for example, two servers behind a firewall and one of the servers spoofs the identity of the other. If the common name is not checked, a certificate presented by either server is accepted if the IP address matches.

Note: Only domain name, URL, and email ID DNS entries in the SAN field are compared.

Configure common-name verification for an SSL service or service group by using the CLI

At the command prompt, type the following commands to specify server authentication with common-name verification and verify the configuration:

1. To configure a common name in a service, type:

```
set ssl service <serviceName> -commonName <string> -serverAuth
   ENABLED
show ssl service <serviceName>
<!--NeedCopy-->
```

2. To configure a common name in a service group, type:

```
set ssl serviceGroup <serviceGroupName> -commonName <string> -
   serverAuth ENABLED
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
> set ssl service svc1 -commonName xyz.com -serverAuth ENABLED
<!--NeedCopy-->
```

```
show ssl service svc

     Advanced SSL configuration for Back-end SSL Service svc1:
     DH: DISABLED
     Ephemeral RSA: DISABLED
     Session Reuse: ENABLED Timeout: 300 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     Server Auth: ENABLED Common Name: www.xyz.com
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: DISABLED
     SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
    1) CertKey Name: cacert CA Certificate OCSPCheck: Optional
```



```
    1) Cipher Name: ALL
     Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Configure common-name verification for an SSL service or service group by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services** or Navigate to **Traffic Management > Load Balancing > Service Groups**, and open a service or service group.
2. In the SSL Parameters section, select Enable Server Authentication, and specify a common name.

SSL actions and policies

September 14, 2021

An SSL policy evaluates incoming traffic and applies a predefined action to requests that match a rule (expression). Configure the actions before creating the policies, so that you can specify an action when you create a policy. To put a policy into effect, do one of the following:

- Bind the policy to a virtual server on the appliance, so that it applies only to traffic flowing through that virtual server.
- Bind the policy globally, so that it applies to all traffic flowing through the appliance.

SSL actions define SSL settings that you can apply to the selected requests. You associate an action with one or more policies. Data in client connection requests or responses is compared to a rule specified in the policy, and the action is applied to connections that match the rule (expression).

You can configure classic policies with classic expressions and default syntax policies with default syntax expressions for SSL.

Note: Users who are not experienced in configuring policies at the CLI usually find using the configuration utility to be considerably easier.

You can associate a user-defined action or a built-in action to a default syntax policy. Classic policies allow only user-defined actions. In default syntax policy, you can also group policies under a policy label, in which case they are applied only when invoked from another policy.

Common uses of SSL actions and policies include per-directory client authentication, support for Outlook web access, and SSL-based header insertions. SSL-based header insertions contain SSL settings required by a server whose SSL processing has been offloaded to the Citrix ADC appliance.

SSL policies

September 14, 2021

Policies on the Citrix ADC appliance help identify specific connections that you want to process. The processing is based on the actions that are configured for that particular policy. Once you create the policy and configure an action for it, you must do one of the following:

- Bind the policy to a virtual server on the appliance, so that it applies only to traffic flowing through that virtual server.
- Bind the policy globally, so that it applies to all traffic flowing through any virtual server configured on the Citrix ADC appliance.

The Citrix ADC appliance SSL feature supports default syntax (advanced) policies. For a complete description of default syntax expressions, how they work, and how to configure them manually, see Policies and Expressions.

> **Note:**
>
> Users who are not experienced in configuring policies at the CLI usually find using the configuration utility considerably easier.

SSL policies require that you create an action before creating a policy, so that you can specify the actions when you create the policies.
In SSL default syntax policies, you can also use the built-in actions. For more information about built-in actions, see SSL built-in actions and user-defined actions.

SSL default syntax policies

An SSL default syntax policy, also known as an advanced policy, defines a control or a data action to be performed on requests. SSL policies can therefore be categorized as control policies and data policies:

- **Control policy**. A control policy uses a control action, such as forcing client authentication. Note: In release 10.5 or later, deny SSL renegotiation (denySSLReneg) is set, by default, to ALL. However, control policies, such as CLIENTAUTH, trigger a renegotiation handshake. If you use such policies, you must set denySSLReneg to NO.
- **Data policy**. A data policy uses a data action, such as inserting some data into the request.

The essential components of a policy are an expression and an action. The expression identifies the requests on which the action is to be performed.

You can configure a default syntax policy with a built-in action or a user-defined action. You can configure a policy with a built-in action without creating a separate action. However, to configure a policy with a user-defined action, first configure the action and then configure the policy.

You can specify an extra action, called an UNDEF action, to be performed when applying the expression to a request has an undefined result.

SSL policy configuration

You can configure an SSL default syntax policy by using the CLI and the GUI.

Configure an SSL policy by using the CLI

At the command prompt, type:

```
1 add ssl policy <name> -rule <expression> -Action <string> [-undefAction
      <string>] [-comment <string>]
2 <!--NeedCopy-->
```

Configure an SSL policy by using the GUI

Navigate to **Traffic Management > SSL > Policies** and, on the **Polices** tab, click *Add*.

Support for SSL policies with TLS1.3 protocol

From release 13.0 build 71.x and later, support is added for SSL policies with the TLS1.3 protocol. When the TLSv1.3 protocol is negotiated for a connection, policy rules that inspect TLS data received from the client now trigger the configured action.

For example, if the following policy rule returns true, the traffic is forwarded to the virtual server defined in the action.

```
1 add ssl action action1 -forward vserver2
2 add ssl policy pol1 -rule client.ssl.client_hello.sni.contains( “xyz” )
      -action action1
3 <!--NeedCopy-->
```

Limitations

- Control policies are not supported.
- The following actions are not supported:
  - DOCLIENTAUTH
  - NOCLIENTAUTH
  - caCertGrpName
  - clientCertVerification
  - ssllogProfile

SSL built-in actions and user-defined actions

September 14, 2021

Unless you need only the built-in actions in your policies, you have to create the actions before creating the policies. Then, you can specify the actions when you create the policies. The built-in actions are of two types, control actions and data actions. You use control actions in control policies, and data actions in data policies.

The built-in control actions are:

- DOCLIENTAUTH—Perform client certificate authentication. (Not supported for TLS1.3)
- NOCLIENTAUTH—Do not perform client certificate authentication. (Not supported for TLS1.3)

The built-in data actions are:

- RESET—Close the connection by sending an RST packet to the client.
- DROP—Drop all packets from the client. The connection remains open until the client closes it.
- NOOP—Forward the packet without performing any operation on it.

Note: Any dependent actions to client authentication, such as clientCertVerification and ssllogProfile, are not supported with the TLS 1.3 protocol.

You can create user-defined data actions. If you enable client authentication, you can create an SSL action to insert client-certificate data into the request header before forwarding the request to the web server.

If a policy evaluation results in an undefined state, an UNDEF action is performed. For either a data policy or a control policy, you can specify RESET, DROP, or NOOP as the UNDEF action. For a control policy, you also have the option of specifying DOCLIENTAUTH or NOCLIENTAUTH.

Examples of built-in actions in a policy

In the following example, if the client sends a cipher other than an EXPORT category cipher, the Citrix ADC appliance requests client authentication. The client has to provide a valid certificate for a successful transaction.

```
1 add ssl policy pol1 -rule CLIENT.SSL.CIPHER_EXPORTABLE.NOT -reqAction
     DOCLIENTAUTH
2 <!--NeedCopy-->
```

The following examples assume that client authentication is enabled.

If the version in the certificate provided by the user matches the version in the policy, no action is taken and the packet is forwarded:


```
1 add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
     reqAction NOOP
2 <!--NeedCopy-->
```

If the version in the certificate provided by the user matches the version in the policy, the connection is dropped:

```
1 add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
     reqAction DROP
2 <!--NeedCopy-->
```

If the version in the certificate provided by the user matches the version in the policy, the connection is reset:

```
1 add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
     reqAction RESET
2 <!--NeedCopy-->
```

Client certificate verification with policy based client authentication

You can set client certificate verification to mandatory or option when you have configured policy based client authentication. Default is mandatory.

Set client certificate verification to optional using the CLI

At the command prompt, type:

```
1 add ssl action <name> ((-clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) [-
     clientCertVerification ( Mandatory | Optional )]
2 <!--NeedCopy-->
```

Example:

```
1 add ssl action sslact -clientauth DOCLIENTAUTH -clientcertverification
     OPTIONAL
2 <!--NeedCopy-->
```

Set client certificate verification to optional using the GUI

1. Navigate to **Traffic Management > SSL > Policies**.
2. On the **SSL Actions** tab, click **Add**.

3. Specify a name and in the **Client Certificate Verification** list, select **Optional**.

User-defined SSL actions

In addition to built-in actions, you can also configure other SSL actions depending on your deployment. These actions are called user-defined actions.

Configure a user-defined SSL action by using the CLI

At the command prompt, type the following commands to configure an action and verify the configuration:

```
add SSL action <name> -clientAuth(DOCLIENTAUTH | NOCLIENTAUTH) -
    clientCert (ENABLED | DISABLED) certHeader <string> -clientHeader <
    string> -clientCertSerialNumber (ENABLED | DISABLED) -
    certSerialHeader <string> -clientCertSubject (ENABLED | DISABLED) -
    certSubjectHeader <string> -clientCertHash (ENABLED | DISABLED) -
    certHashHeader <string> -clientCertIssuer (ENABLED | DISABLED) -
    certIssuerHeader <string> -sessionID (ENABLED | DISABLED) -
    sessionIDheader <string> -cipher (ENABLED | DISABLED) -cipherHeader
    <string> -clientCertNotBefore (ENABLED | DISABLED) -
    certNotBeforeHeader <string> -clientCertNotAfter (ENABLED | DISABLED
    ) -certNotAfterHeader <string> -OWASupport (ENABLED | DISABLED)
<!--NeedCopy-->
```

```
show ssl action [<name>]
<!--NeedCopy-->
```

Example:

```
add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X
    -Client-Cert"
<!--NeedCopy-->
```

```
show ssl action Action-SSL-ClientCert

1)      Name: Action-SSL-ClientCert
        Data Insertion Action:
        Cert Header: ENABLED            Cert Tag: X-Client-Cert
Done
<!--NeedCopy-->
```


Configure a user-defined SSL action by using the GUI

Navigate to **Traffic Management > SSL > Policies** and, on the **Actions** tab, click **Add**.

Configure an SSL action to forward client traffic to another virtual server

Admins can configure an SSL action to forward the client traffic received on an SSL virtual server to another virtual server to avoid SSL offloading. Or for terminating the connection on the ADC appliance. This virtual server can be of the type: SSL, TCP, or SSL_BRIDGE. For example, admins can choose to forward the request to another virtual server for further action instead of terminating the connection if any of the following cases:

- The appliance does not have a certificate.
- The appliance does not support a specific cipher.

To achieve the above, a new bind point 'CLIENTHELLO_REQ' is added to evaluate client traffic when a client hello is received. If the policy bound to the virtual server receiving client traffic evaluates to true after parsing the client hello, the traffic is forwarded to another virtual server. If this virtual server is of type SSL, it performs the handshake. If this virtual server is of type TCP or SSL_BRIDGE, the back-end server performs the handshake.

In release 12.1-49.x, only the forward and reset actions are supported for the CLIENTHELLO_REQ bind point. The following expression prefixes are available:

- CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
- CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION
- CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE
- CLIENT.SSL.CLIENT_HELLO.IS_REUSE
- CLIENT.SSL.CLIENT_HELLO.IS_SCSV
- CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET
- CLIENT.SSL.CLIENT_HELLO.LENGTH
- CLIENT.SSL.CLIENT_HELLO.SNI
- CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPROTOCOL (from release 13.0 build 61.x)

For a description of these prefixes, see Advanced policy expressions: parsing SSL.

A parameter `forward` is added to the `add SSL action` command and a new bind point `CLIENTHELLO_REQ` is added to the `bind ssl vserver` command.

Configuration using the CLI

At the command prompt, type:

```
add ssl action <name> -forward <virtual server name>
```



```
3 add ssl policy <name> -rule <expression> -action <string>
4
5 bind ssl vserver <vServerName> -policyName <string> -priority <
     positive_integer> -type <type>
6 <!--NeedCopy-->
```

EXAMPLE:

```
1 add ssl action act1 -forward v2
2
3 add ssl policy pol1 -rule client.ssl.client_hello.ciphers.has_hexcode(0
     x002f) -action act1
4
5 bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
6 <!--NeedCopy-->
```

Configuration using the GUI

Navigate to **Traffic Management > SSL > Policies**.

Create SSL action:

1. In **SSL Actions**, click **Add**.
2. In **Create SSL Action**, specify a name for the action.
3. In **Forward Action Virtual Server**, select an existing virtual server or add a new virtual server to forward the traffic to.
4. Optionally, set other parameters.
5. Click **Create**.

Create SSL policy:

1. In **SSL Policies**, click **Add**.
2. In **Create SSL Policy**, specify a name for the policy.
3. In **Action**, select the action that you created earlier.
4. In **Expression Editor**, enter the rule to evaluate.
5. Click **Create**.

Create or add a virtual server and bind policy:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Add or select a virtual server.
3. In **Advanced Settings**, click **SSL Policies**.
4. Click in the SSL Policy section.
5. In **Select Policy**, select the policy that you created earlier.
6. In **Policy Binding**, specify a priority for the policy.

7. In **Type**, select **CLIENTHELLO_REQ**.
8. Click **Bind**.
9. Click **Done**.

For the end-to-end configuration for the most popular use cases, see the following topics:

- Configure SSL action to forward client traffic if the appliance does not have a domain specific (SNI) certificate.
- Configure an SSL action to forward client traffic based on the protocol in the ALPN extension of the client hello message.
- Configure SSL action to forward client traffic if a cipher is not supported on the ADC.

SSL action to selectively pick CAs based on SNI for client authentication

You can send only the list of CAs based on SNI (domain) in the client certificate request rather than the list of all the CAs bound to an SSL virtual server. For example, when a client hello is received, only the CA certificates based on the SSL policy expression (for example, SNI) are sent. To send a specific set of certificates, you must create a CA certificates group. Then, bind this group to an SSL action, and bind the action to an SSL policy. If the policy bound to the virtual server receiving client traffic evaluates to true after parsing the client hello, only a specific CA certificates group is sent in the client request certificate.

Earlier, you had to bind CA certificates to an SSL virtual server. With this enhancement, you can simply add CA certificate groups and associate them to an SSL action.

Note: Enable client authentication and SNI on the SSL virtual server. Bind the correct SNI certificates to the virtual server.

Perform the following steps:

1. Add a CA certificate group.
2. Add certificate-key pairs.
3. Bind the certificate-key pairs to this group.
4. Add an SSL action.
5. Add an SSL policy. Specify the action in the policy.
6. Bind the policy to an SSL virtual server. Specify the bind point as CLIENTHELLO_REQ.

Configuration using the CLI

At the command prompt, type the following commands in a sequence:


```
add ssl caCertGroup <caCertGroupName>
add ssl certkey <certkey_name> -cert <cert> -key <key>
bind ssl caCertGroup <caCertGroupName> <certkey_name>
add ssl action <name> -caCertGrpName <string>
add ssl policy <name> -rule <expression> -action <string>
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

Example:

```
add ssl cacertGroup ca_cert_group

add ssl certkey ca_certkey1 -cert cacert1 -key cakey1
add ssl certkey ca_certkey2 -cert cacert2 -key cakey2
add ssl certkey snicert -cert snicert -key snikey

bind ssl cacertGroup ca_cert_group ca_certkey1
bind ssl caCertGroup ca_cert_group ca_certkey2
<!--NeedCopy-->
```

```
sh ssl caCertGroup ca_cert_group

CA GROUP NAME:     ca_cert_group
ACTIONS REFERRING: 1

1) CertKey Name: ca_certkey1   CA Certificate   CRLCheck: Optional
   CA_Name Sent
2) CertKey Name: ca_certkey2   CA Certificate   CRLCheck: Optional
   CA_Name Sent
<!--NeedCopy-->
```

```
add ssl action pick_ca_group -cacertGrpName ca_cert_group
<!--NeedCopy-->
```

```
sh ssl action pick_ca_group
1) Name: pick_ca_group
   Type: Data Insertion
   PickCaCertGroup: ca_cert_group
   Hits: 0
   Undef Hits: 0
   Action Reference Count: 1
<!--NeedCopy-->
```



```
add ssl policy snipolicy -rule client.ssl.client_hello.sni.contains("
    abc") -action pick_ca_group
bind ssl vserver v_SSL -policyName snipolicy -type CLIENTHELLO_REQ -
    priority 10
<!--NeedCopy-->
```

```
sh ssl policy snipolicy
    Name: snipolicy
    Rule: client.ssl.client_hello.sni.contains("abc")
    Action: pick_ca_group
    UndefAction: Use Global
    Hits: 0
    Undef Hits: 0


    Policy is bound to following entities
1)  Bound to: CLIENTHELLO_REQ VSERVER v_SSL
    Priority: 10
<!--NeedCopy-->
```

```
set ssl vserver v_SSL -clientauth ENABLED -SNIEnable ENABLED
bind ssl vserver v_SSL -certkeyName snicert -sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v_SSL

    Advanced SSL configuration for VServer v_SSL:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED   Refresh Count: 0
    Session Reuse: ENABLED   Timeout: 120 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: ENABLED   Client Cert Required: Mandatory
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: ENABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED
```



```
            TLSv1.2: ENABLED  TLSv1.3: DISABLED
19      Push Encryption Trigger: Always
20      Send Close-Notify: YES
21      Strict Sig-Digest Check: DISABLED
22      Zero RTT Early Data: DISABLED
23      DHE Key Exchange With PSK: NO
24      Tickets Per Authentication Context: 1
25
26      ECC Curve: P_256, P_384, P_224, P_521
27
28  1)  CertKey Name: snicert   Server Certificate for SNI
29
30
31      Data policy
32  1)  Policy Name: snipolicy  Priority: 10
33
34
35
36  1)  Cipher Name: DEFAULT
37      Description: Default cipher list with encryption strength >= 128bit
38  <!--NeedCopy-->
```

Configuration using the GUI

Create CA certificates group and bind certificates to the group:

1. Navigate to **Traffic Management > SSL > CA Certificates Group**.
2. Click **Add** and specify a name for the group.
3. Click **Create**.
4. Select the **CA certificate group** and then click **Show Bindings**.
5. Click **Bind**.
6. In the **CA Certificate Binding** page, select an existing certificate or click Add to add a new certificate.
7. Click **Select** and then click **Bind**.
8. To bind another certificate, repeat steps 5 through 7.
9. Click **Close**.

Navigate to **Traffic Management > SSL > Policies**.

Create SSL action:

1. In **SSL Actions**, click **Add**.
2. In **Create SSL Action**, specify a name for the action.

3. In **Forward Action Virtual Server**, select an existing virtual server or add a virtual server to forward the traffic to.
4. Optionally, set other parameters.
5. Click **Create**.

Create SSL policy:

1. In **SSL Policies**, click **Add**.
2. In **Create SSL Policy**, specify a name for the policy.
3. In **Action**, select the action created earlier.
4. In **Expression Editor**, enter the rule to evaluate.
5. Click **Create**.

Create or add a virtual server and bind policy:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Add or select a virtual server.
3. In **Advanced Settings**, click **SSL Policies**.
4. Click in the SSL Policy section.
5. In **Select Policy**, select the policy that you created earlier.
6. In **Policy Binding**, specify a priority for the policy.
7. In **Type**, select **CLIENTHELLO_REQ**.
8. Click **Bind**.
9. Click **Done**.

Unbind a CA certificate group by using the GUI

1. Navigate to **Traffic Management > SSL > CA Certificates Group**.
2. Select a certificate group and click **Show Bindings**.
3. Select the certificate to remove from the group and click **Unbind**.
4. If prompted for confirmation, click **Yes••.
5. Click **Close**.

Remove a CA certificate group by using the GUI

1. Navigate to **Traffic Management > SSL > CA Certificates Group**.
2. Select a certificate group and click **Delete**.
3. If prompted for confirmation, click **Yes**.

SSL policy binding

September 14, 2021

You can bind SSL policies globally or to an SSL type virtual server only. Globally bound policies are evaluated after all policies bound to services, virtual servers, or other Citrix ADC bind points are evaluated. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered, and the action associated with it is carried out.

When binding an SSL policy to a virtual server, you must select from one of the following bind points:

- REQUEST (Default bind point. Policy evaluation is done in the HTTP layer after the SSL handshake is completed.)
- INTERCEPT_REQ (This option applies to a Citrix Secure Web Gateway setup. For more information, see SSL policy infrastructure for SSL interception).
- CLIENTHELLO_REQ

Similarly, when unbinding a policy from a virtual server, you must specify the bind point.

If you specify CLIENTHELLO_REQ as the bind point, the policy is evaluated when a client hello message is received. The allowed actions are RESET, FORWARD, and `caCertGrpName`. The reset action terminates the connection. The forward action forwards the request to a load balancing virtual server for processing. The `caCertGrpName` action selectively picks CAs based on SNI for client authentication. For more information about SSL actions, see SSL built-in actions and user-defined actions.

Note: The action caCertGrpName is not supported with the TLS 1.3 protocol.

Bind an SSL policy globally by using the CLI

At the command prompt, type the following command to bind a global SSL policy and verify the configuration:

```
bind ssl global - policyName <string> [- priority <positive_integer>]
show ssl global
<!--NeedCopy-->
```

Example:

```
bind ssl global -policyName Policy-SSL-2 -priority 90
Done

sh ssl global

      1) Name: Policy-SSL-2 Priority: 90
```



```
      2) Name: Policy-SSL-1 Priority: 100
      Done
<!--NeedCopy-->
```

Bind an SSL policy globally by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Policies**.
2. In the details pane, click **Global Bindings**.
3. In the **Bind/Unbind SSL Policies to Global** dialog box, click **Insert Policy**.
4. In the **Policy Name** list, select a policy.
5. Optionally, drag the entry to a new position in the policy bank to automatically update the priority level.
6. Click **OK**. A message appears in the status bar, stating that the policy has been bound successfully.

Bind or unbind an SSL policy to a virtual server by using the CLI

At the command prompt, type the following command to bind an SSL policy to a virtual server and verify the configuration:

```
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type <type>

unbind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type <type>

<!--NeedCopy-->
```

Example:

```
bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
unbind ssl vserver v1 -policyName pol1 -priority 1 -type
   CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
show ssl vserver vs-server

Advanced SSL configuration for VServer vs-server:

DH: DISABLED
```



```
6
7  Ephemeral RSA: ENABLED          Refresh Count: 1000
8
9  Session Reuse: ENABLED          Timeout: 120 seconds
10
11 Cipher Redirect: DISABLED
12
13 SSLv2 Redirect: DISABLED
14
15 ClearText Port: 80
16
17 Client Auth: DISABLED
18
19 SSL Redirect: ENABLED
20
21 SSL-REDIRECT Port Rewrite: ENABLED
22
23 Non FIPS Ciphers: DISABLED
24
25 SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
26
27 1)      Policy Name: ssl-policy-1        Priority: 10
28
29 1)      Cipher Name: DEFAULT
30
31             Description: Predefined Cipher Alias
32
33 Done
34 <!--NeedCopy-->
```

Bind an SSL policy to a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open an SSL virtual server.
2. In **Advanced Settings**, select **SSL Policy**. Click in the **SSL policy** section to bind a policy to the virtual server.
3. In the **Policy Binding** page, select an existing policy or add a new policy.
4. Specify priority and type (bind point) for the policy.
5. Select **Bind**.
6. Select **Done**.

SSL policy labels

September 14, 2021

Policy labels are holders for policies. A policy label helps in managing a group of policies, called a policy bank, which can be invoked from another policy. SSL policy labels can be control labels or data labels, depending on the type of policies that are included in the policy label. You can add only data policies in a data policy label and only control policies in a control policy label. To create the policy bank, bind policies to the label and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. At the CLI, you enter two commands to create a policy label and bind policies to the policy label. In the configuration utility, you select options from a dialog box.

Note: Policy labels of type control are not supported with the TLS 1.3 protocol.

Create an SSL policy label and bind policies to the label by using the CLI

At the command prompt, type:

```
add ssl policylabel <labelName> -type ( CONTROL | DATA )

bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>) ]
<!--NeedCopy-->
```

Example:

```
add ssl policylabel cpl1 -type CONTROL

add ssl policylabel dpl1 -type DATA

bind ssl policylabel cpl1 -policyName ctrlpol -priority 1

bind ssl policylabel dpl1 -policyName datapol -priority 1
<!--NeedCopy-->
```

Configure an SSL policy label and bind policies to the label by using the GUI

Navigate to **Traffic Management > SSL > Policy Labels**, and configure an SSL policy label.

Selective SSL logging

September 14, 2021

In a large deployment comprising thousands of virtual servers, all SSL-related information is logged. Earlier, filtering the client authentication and SSL handshake successes and failures for a few critical virtual servers was not easy. Perusing through the entire log to get this information was a time-consuming and tedious task because the infrastructure did not offer the control to filter the logs. Now, you can log SSL-related information in `ns.log`, for a specific virtual server or for a group of virtual servers. This information is especially helpful in debugging failures. To log this information, you must add an SSL log profile.

See sample ns.log output for successful client authentication at the end of this page.

Important: Set the syslog log level to DEBUG. At the command prompt, type:

```
set audit syslogParams -logLevel DEBUG
```

SSL log profile

An SSL log profile provides control over logging the following events for a virtual server or a group of virtual servers:

- Client authentication success and failures, or only failures.
- SSL handshake success and failures, or only failures.

By default, all the parameters are disabled.

An SSL log profile can be set on an SSL profile, or on an SSL action. If set to an SSL profile, you can log both client authentication and SSL handshake success and failure information. If set to an SSL action, you can only log client authentication success and failure information because the handshake is complete before the policy is evaluated.

Client authentication and SSL handshake success and failures are logged even if you do not configure an SSL log profile. However, selective logging is possible only if an SSL log profile is used.

> **Note:**
>
> SSL log profile is supported in high availability and cluster setups.

Add an SSL log profile by using the CLI

At the command prompt, type:


```
add ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )] [-
    ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
    DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

Parameters:

Name:

Name for the SSL log profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created.

Name is a mandatory argument. Maximum Length: 127

sslLogClAuth:

Log all client authentication events. Includes both success and failure events.

Possible values: ENABLED, DISABLED

Default value: DISABLED

ssllogClAuthFailures:

Log all client authentication failure events.

Possible values: ENABLED, DISABLED

Default value: DISABLED

sslLogHS:

Log all SSL handshake related events. Includes both success and failure events.

Possible values: ENABLED, DISABLED

Default value: DISABLED

sslLogHSfailures:

Log all SSL handshake related failure events.

Possible values: ENABLED, DISABLED

Default value: DISABLED

Example:

```
> add ssl logprofile ssllog10 -sslLogClAuth ENABLED -sslLogHS ENABLED

 Done

sh ssllogprofile ssllog10
```



```
6
7  1)      Name: ssllog10
8
9          SSL log ClientAuth [Success/Failures] : ENABLED
10
11         SSL log ClientAuth [Failures] : DISABLED
12
13         SSL log Handshake [Success/Failures] : ENABLED
14
15         SSL log Handshake [Failures] : DISABLED
16
17  Done
18 <!--NeedCopy-->
```

Add an SSL log profile by using the GUI

Navigate to **System > Profiles > SSL Log Profile** and add a profile.

Modify an SSL log profile by using the CLI

At the command prompt type:

```
1 set ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )][-
     ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
     DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
2 <!--NeedCopy-->
```

Example:

```
1 set ssllogprofile ssllog10 -ssllogClAuth en -ssllogClAuthFailures en -
     ssllogHS en -ssllogHSfailures en
2
3 Done
4
5 sh ssllogprofile ssllog10
6
7     1)            Name: ssllog10
8
9                     SSL log ClientAuth [Success/Failures] : ENABLED
10                    SSL log ClientAuth [Failures] : ENABLED
11                    SSL log Handshake [Success/Failures] : ENABLED
12                    SSL log Handshake [Failures] : ENABLED
13      Done
14 <!--NeedCopy-->
```


Modify an SSL log profile by using the GUI

1. Navigate to **System > Profiles > SSL Log Profile**, select a profile, and click **Edit**.
2. Make changes and click **OK**.

View all the SSL log profiles by using the CLI

At the command prompt, type:

```
sh ssl logprofile
<!--NeedCopy-->
```

Example:

```
sh ssl logprofile

    1)            Name: ssllogp1
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : ENABLED

    2)            Name: ssllogp2
                    SSL log ClientAuth [Success/Failures] : DISABLED
                    SSL log ClientAuth [Failures] : DISABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : DISABLED

    3)            Name: ssllogp3
                    SSL log ClientAuth [Success/Failures] : DISABLED
                    SSL log ClientAuth [Failures] : DISABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : DISABLED

    4)            Name: ssllog10
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : ENABLED
                    SSL log Handshake [Failures] : ENABLED
Done
<!--NeedCopy-->
```


View all the SSL log profiles by using the GUI

Navigate to **System > Profiles > SSL Log Profile**. All the profiles are listed.

Attach an SSL log profile to an SSL profile

You can attach (set) an SSL log profile on an SSL profile when you are creating an SSL profile, or later by editing the SSL profile. You can log both client authentication and handshake successes and failures.

> **Important:**
>
> The default SSL profile must be enabled before you can attach an SSL log profile.

Attach an SSL log profile on an SSL profile by using the CLI

At the command prompt, type:

```
set ssl profile <name> [-ssllogProfile <string>]
<!--NeedCopy-->
```

Example:

```
set ssl profile fron_1 -ssllogProfile ssllog10
<!--NeedCopy-->
```

Attach an SSL log profile to an SSL profile by using the GUI

1. Navigate to **System > Profiles > SSL Profile**.
2. Click **Edit** and in **SSL Log Profile**, specify a profile.

Attach an SSL log profile to an SSL action

You can set an SSL log profile only while creating an SSL action. You cannot modify an SSL action to set the log profile. Associate the action to a policy. You can only log client authentication successes and failures.

Attach an SSL log profile to an SSL action by using the CLI

At the command prompt, type:

```
add ssl action <name> -clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) -
    ssllogProfile <string>
<!--NeedCopy-->
```


Example:

```
> add ssl action act1 -clientAuth DoCLIENTAUTH -ssllogProfile ssllog10

Done

> sh ssl action act1

    1)            Name: act1
                    Type: Client Authentication (DOCLIENTAUTH)
                    Hits: 0
                    Undef Hits: 0
                    Action Reference Count: 0
                    SSLlogProfile: ssllog10
Done
<!--NeedCopy-->
```

Attach an SSL log profile to an SSL action by using the GUI

1. Navigate to **Traffic Management > SSL > Policies** and click **SSL Actions**.
2. Click **Add**.
3. In Client Authentication, select **ENABLED**.
4. In SSL Log Profile, select a profile from the list, or click “+” to create a profile.
5. Click **Create**.

Sample output from the log file

The following is a sample log output from `ns.log` for successful client authentication.

```
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 158 0 :  SPCBId 671 -
    ClientIP 10.102.1.98 - ClientPort 49451 - VserverServiceIP
    10.102.57.82 - VserverServicePort 443 - ClientVersion TLSv1.2 -
    CipherSuite "AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit" - Session
    New - CLIENT_AUTHENTICATED -SerialNumber "2A" - SignatureAlgorithm "
    sha1WithRSAEncryption" - ValidFrom "Sep 22 09:15:20 2008 GMT" -
    ValidTo "Feb  8 09:15:20 2036 GMT" - HandshakeTime 10 ms
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 159 0 :  SPCBId 671
    - IssuerName " C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix"
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 160 0 :  SPCBId 671
     - SubjectName " C=IN,ST=KAR,O=Citrix Pvt Ltd,OU=A,CN=B"
```



```
4 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 161 0 :  Backend SPCBId
     674 - ServerIP 10.102.57.85 - ServerPort 443 - ProtocolVersion
    TLSv1.2 - CipherSuite "AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit" -
     Session Reuse - SERVER_AUTHENTICATED -SerialNumber "3E" -
    SignatureAlgorithm "sha1WithRSAEncryption" - ValidFrom "Sep 24
    06:40:37 2008 GMT" - ValidTo "Feb 10 06:40:37 2036 GMT" -
    HandshakeTime 1 ms
5 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 162 0 :  SPCBId 674
    - IssuerName " C=IN,ST=KAR,O=Citrix Pvt Ltd"
6 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 163 0 :  SPCBId 674
     - SubjectName " C=IN,ST=P,L=Q,O=R"
7 <!--NeedCopy-->
```

Support for DTLS protocol

September 14, 2021

Notes:

- DTLSv1.0 protocol is supported on Citrix ADC MPX/SDX (N2 and N3 based), VPX, and MPX 14000 FIPS appliances. It is not supported on external HSMs.
- DTLS 1.0 protocol is supported on Citrix ADC appliances containing Intel Coleto SSL chips (from release 12.1 build 50.x).
- DTLSv1.2 protocol is supported on the front-end of Citrix ADC VPX appliances (from release 13.0 build 47.x).
- DTLS 1.2 protocol is supported on the front-end of Citrix ADC appliances containing Intel Coleto SSL chips (from release 13.0 build 52.x). For more information about the platforms containing Intel Coleto SSL chips, see Support for Intel Coleto SSL chip based platforms.
- Service groups of type DTLS are not supported.
- DTLSv1.2 protocol is supported on the front-end of Citrix ADC MPX (N3 based) appliances (from release 13.0 build 58.x).
- For information about Enlightened Data Transport (EDT) support for Citrix Gateway, see HDX enlightened data transport support.
- There are changes made to the DTLS profile from release 13.0 build 79.x. For more information, see DTLS profile.
- From release 13.0 build 82.x, a new parameter “maxBadmacIgnorecount “ is introduced in the DTLS profile to ignore bad MAC records received in a DTLS session. For more informa-

> tion, see DTLS profile.

The SSL and TLS protocols have traditionally been used to secure streaming traffic. Both of these protocols are based on TCP, which is slow. Also, TLS cannot handle lost or reordered packets.

UDP is the preferred protocol for audio and video applications, such as Lync, Skype, iTunes, YouTube, training videos, and flash. However, UDP is not secure or reliable. The DTLS protocol is designed to secure data over UDP and is used for applications such as media streaming, VOIP, and online gaming for communication. In DTLS, each handshake message is assigned a specific sequence number within that handshake. When a peer receives a handshake message, it can quickly determine whether that message is the next one expected. If it is, the peer processes the message. If not, the message is queued for handling after all the previous messages have been received.

Create a DTLS virtual server and a service of type UDP. By default, a DTLS profile (nsdtls_default_profile) is bound to the virtual server. Optionally, you can create and bind a user-defined DTLS profile to the virtual server.

Note: RC4 ciphers are not supported on a DTLS virtual server.

DTLS configuration

You can use the command line (CLI) or the configuration utility (GUI) to configure DTLS on your ADC appliance.

Note: From release 13.0 build 47.x, the DTLS 1.2 protocol is supported on the front end of a Citrix ADC VPX appliance. While configuring a DTLSv1.2 virtual server, specify DTLS12. Default is DTLS1.

At the command prompt, type:

```
set ssl vserver DTLS [-dtls1 ( ENABLED | DISABLED )] [-dtls12 ( ENABLED | DISABLED )]
```

Create a DTLS configuration by using the CLI

At the command prompt, type:

```
add lb vserver <vserver_name> DTLS <IPAddress>  <port>
add service  <service_name>  <IPAddress> UDP 443
bind lb vserver  <vserver_name>  <udp_service_name>
<!--NeedCopy-->
```

The following steps are optional:

```
add dtlsProfile dtls-profile -maxretryTime <positive_integer>
set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
<!--NeedCopy-->
```


Create a DTLS configuration by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Create a virtual server of type DTLS, and bind a UDP service to the virtual server.
3. A default DTLS profile is bound to the DTLS virtual server. To bind a different profile, in SSL Parameters, select a different DTLS profile. To create a profile, click the plus (+) next to DTLS Profile.

Support for SNI on a DTLS virtual server

For information about SNI, see Configure an SNI virtual server for secure hosting of multiple sites.

Configure SNI on a DTLS virtual server by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName> -SNIEnable ENABLED
bind ssl vserver <vServerName> -certkeyName <string> -SNICert
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver v1 -sniEnable ENABLED
bind ssl vserver  v1 -certkeyName san2 -sniCert
bind ssl vserver  v1 -certkeyName san13  - sniCert
bind ssl vserver  v1 -certkeyName san17  - sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED
Refresh Count: 0
Session Reuse: ENABLED
Timeout: 1800 seconds
Cipher Redirect: DISABLED

ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
```



```
SNI: ENABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
DTLSv1: ENABLED
Send Close-Notify: YES
Strict Sig-Digest Check: DISABLED
Zero RTT Early Data: DISABLED
DHE Key Exchange With PSK: NO
Tickets Per Authentication Context: 1

DTLS profile name: nsdtls_default_profile

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca
CA Certificate  OCSPCheck: OptionalCA_Name Sent
2) CertKey Name: san2 Server Certificate for SNI
3) CertKey Name: san17 Server Certificate for SNI
4) CertKey Name: san13 Server Certificate for SNI


1) Cipher Name: DEFAULT
Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

Configure SNI on a DTLS virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a DTLS virtual server and, in Certificates, click **Server Certificate**.
3. Add a certificate or select a certificate from the list and select **Server Certificate for SNI**.
4. In **Advanced Settings**, click **SSL Parameters**.
5. Select **SNI Enable**.

Features not supported by a DTLS virtual server

The following options cannot be enabled on a DTLS virtual server:

- SSLv2
- SSLv3
- TLSv1

- TLSv1.1
- TLSv1.2
- Push encrypt trigger
- SSLv2Redirect
- SSLv2URL

Parameters not used by a DTLS virtual server

A DTLS virtual server ignores the following SSL parameters, even if set:

- Encryption trigger packet count
- PUSH encryption trigger timeout
- SSL quantum size
- Encryption trigger timeout
- Subject/Issuer Name Insertion Format

Configure renegotiation on a DTLS service

Non-secure renegotiation is supported on a DTLS service. You can use the CLI or the GUI to configure this setting.

Configure renegotiation on a DTLS service by using the CLI

At the command prompt, type:

```
set ssl parameter -denysslreneg NONSECURE
<!--NeedCopy-->
```

Example:

```
set ssl parameter -denysslreneg NONSECURE


sh ssl parameter
Advanced SSL Parameters
-----------------------
SSL quantum size                                  : 8 KB
Max CRL memory size                               : 256 MB
Strict CA checks                                  : NO
Encryption trigger timeout                        : 100 ms
Send Close-Notify                                 : YES
Encryption trigger packet count                   : 45
Deny SSL Renegotiation                            : NONSECURE
```



```
Subject/Issuer Name Insertion Format                  : Unicode
OCSP cache size                                       : 10 MB
Push flag                                             : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout                       : 1 ms
Crypto Device Disable Limit                           : 0
Global undef action for control policies              : CLIENTAUTH
Global undef action for data policies                 : NOOP
Default profile                                       : DISABLED
SSL Insert Space in Certificate Header                : YES
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services      : NO
Software Crypto acceleration CPU Threshold            : 0
Hybrid FIPS Mode                                      : DISABLED
Signature and Hash Algorithms supported by TLS1.2     : ALL
SSL Interception Error Learning and Caching           : DISABLED
SSL Interception Maximum Error Cache Memory           : 0 Bytes
Done
<!--NeedCopy-->
```

Configure renegotiation on a DTLS service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Select a DTLS service and click **Edit**.
3. Navigate to **SSL** > **Advanced Settings**.
4. Select **Deny SSL Renegotiation**.

Features not supported by a DTLS service

The following options cannot be enabled on a DTLS service:

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- Push encrypt trigger
- SSLv2Redirect
- SSLv2URL
- SNI
- Secure renegotiation

Parameters not used by a DTLS service

A DTLS service ignores the following SSL parameters, even if set:

- Encryption trigger packet count
- PUSH encryption trigger timeout
- SSL quantum size
- Encryption trigger timeout
- Subject/Issuer Name Insertion Format

> **Note:**
>
> SSL session reuse handshake fails on a DTLS service because session reuse is not currently supported on DTLS services.
>
> **Workaround:** Manually disable session reuse on a DTLS service. At the CLI, type:
>
> `set ssl service <dtls-service-name> -sessReuse DISABLED`

DTLS profile

A DTLS profile with the default settings is automatically bound to a DTLS virtual server. However, you can create a DTLS profile with specific settings to suit your requirement.

Use a DTLS profile with a DTLS virtual server or a VPN DTLS virtual server. You cannot use an SSL profile with a DTLS virtual server.

> **Note:**
>
> Change the maximum record size setting in the DTLS profile based on the changes in MTU and packet size. For example, the default max record size of 1459 bytes is calculated based on an IPv4 address header size. With IPv6 records, the header size is larger and therefore the maximum record size must be reduced to meet the following criteria.

`max record size + UDP header(8bytes)+ IP header size < MTU`

Example:

```
Default DTLS profile
    1) Name: nsdtls_default_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 3 sec
    Hello Verify Request: ENABLED
    Terminate Session: DISABLED
    Max Packet Count: 120 bytes

Custom DTLS profile
```



```
    1) Name: ns_dtls_profile_ipv6_1
    PMTU Discovery: DISABLED
    Max Record Size: 1450 bytes
    Max Retry Time: 3 sec
    Hello Verify Request: ENABLED
    Terminate Session: DISABLED
    Max Packet Count: 120 bytes
<!--NeedCopy-->
```

Create a DTLS profile by using the CLI

Notes:

From release 13.0 build 79.x, the changes to the DTLS profile are as follows:

- The `helloverifyrequest` parameter is enabled by default. Enabling this parameter helps mitigate the risk of an attacker or bots overwhelming the network throughput, potentially leading to outbound bandwidth exhaustion. That is, it helps mitigate the DTLS DDoS amplification attack.
- The `maxHoldQlen` parameter is added. This parameter defines the number of datagrams that can be queued at the DTLS layer for processing. A high value of the `maxHoldQlen` parameter can cause memory buildup at the DTLS layer if UDP multiplexing is transmitting high UDP traffic. Therefore, configuring a lower value is recommended. Minimum value is 32, maximum value is 65535, and default value is 32.

From release 13.0 build 82.x, a new parameter `maxBadmacIgnorecount` is introduced in the DTLS profile to ignore bad MAC records received in a DTLS session. Using this parameter, bad records up to the value set in the parameter are ignored. The appliance terminates the session only after the limit is reached and sends an alert.

This parameter setting is effective only when the “terminateSession” parameter is enabled.

```
ssl dtlsProfile <name> -maxRetryTime <positive_integer> -
    helloVerifyRequest ( ENABLED | DISABLED ) -terminateSession (ENABLED
     | DISABLED ) -maxHoldQLen <positive_integer> -maxBadmacIgnorecount
    <positive_integer>

helloVerifyRequest
              Send a Hello Verify request to validate the client.
              Possible values: ENABLED, DISABLED
              Default value: ENABLED

terminateSession
              Terminate the session if the message authentication code
                  (MAC)
```



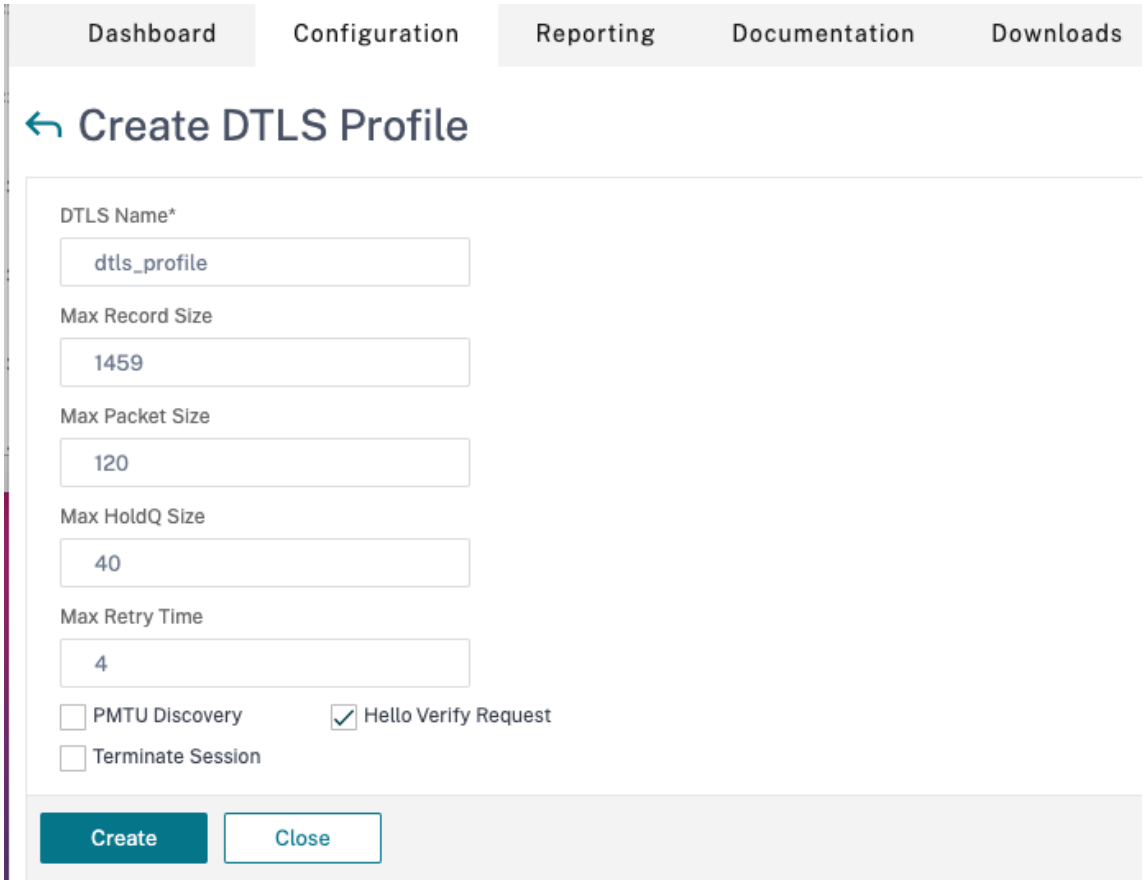
```
                of the client and server do not match.
                Possible values: ENABLED, DISABLED
                Default value: DISABLED

maxHoldQLen
                Maximum number of datagrams that can be queued at DTLS
                    layer for
                processing
                Default value: 32
                Minimum value: 32
                Maximum value: 65535

maxBadmacIgnorecount
                Maximum number of bad MAC errors to ignore for a
                    connection prior disconnect. Disabling parameter
                    terminateSession
terminates session immediately when bad MAC is detected in the
    connection.
                Default value: 100
                Minimum value: 1
                Maximum value: 65535
<!--NeedCopy-->
```

Example:

```
> add ssl dtlsprofile dtls_profile -maxRetryTime 4 -helloVerifyRequest
    ENABLED -terminateSession ENABLED -maxHoldQLen 40 -
    maxBadmacIgnorecount 150
 Done
> sh dtlsprofile dtls_profile
1)  Name: dtls_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 4 sec
    Hello Verify Request: ENABLED
    Terminate Session: ENABLED
    Max Packet Count: 120 bytes
    Max HoldQ Size: 40 datagrams
    Max bad-MAC Ignore Count: 150

 Done
<!--NeedCopy-->
```


Create a DTLS profile by using the GUI

1. Navigate to **System > Profiles > DTLS Profiles** and click **Add**.
2. In the **Create DTLS Profile** page, type values for the different parameters.



3. Click **Create**.

Example for an end-to-end DTLS configuration

```
enable ns feature SSL LB

add server s1 198.51.100.2

en ns mode usnip

add service svc_dtls s1 DTLS 443

add lb vserver v1 DTLS 10.102.59.244 443

bind ssl vserver v1 -ciphername ALL

```



```
13 add ssl certkey servercert -cert servercert_aia_valid.pem -key
      serverkey_aia.pem
14
15 bind ssl vserver v1 -certkeyname servercert
16
17 bind lb vserver lb1 svc_dtls
18
19 sh lb vserver v1
20
21                     v1 (10.102.59.244:4433) - DTLS     Type: ADDRESS
22                     State: UP
23                     Last state change was at Fri Apr 27 07:00:27 2018
24                     Time since last state change: 0 days, 00:00:04.810
25                     Effective State: UP
26                     Client Idle Timeout: 120 sec
27                     Down state flush: ENABLED
28                     Disable Primary Vserver On Down : DISABLED
29                     Appflow logging: ENABLED
30                     No. of Bound Services :  1 (Total) 0 (Active)
31                     Configured Method: LEASTCONNECTION
32                     Current Method: Round Robin, Reason: A new service
                          is bound         BackupMethod: ROUNDROBIN
33                     Mode: IP
34                     Persistence: NONE
35                     L2Conn: OFF
36                     Skip Persistency: None
37                     Listen Policy: NONE
38                     IcmpResponse: PASSIVE
39                     RHIstate: PASSIVE
40                     New Service Startup Request Rate: 0 PER_SECOND,
                          Increment Interval: 0
41                     Mac mode Retain Vlan: DISABLED
42                     DBS_LB: DISABLED
43                     Process Local: DISABLED
44                     Traffic Domain: 0
45                     TROFS Persistence honored: ENABLED
46                     Retain Connections on Cluster: NO
47
48     1) svc_dtls (10.102.59.190: 4433) - DTLS State: UP  Weight: 1
49 Done
50
51
52 sh ssl vserver v1
53
54                     Advanced SSL configuration for VServer v1:
```



```
55                      DH: DISABLED
56                      DH Private-Key Exponent Size Limit: DISABLED
                                  Ephemeral RSA: ENABLED
                                                        Refresh Count: 0
57                      Session Reuse: ENABLED                Timeout:
                           1800 seconds
58                      Cipher Redirect: DISABLED
59                      ClearText Port: 0
60                      Client Auth: DISABLED
61                      SSL Redirect: DISABLED
62                      Non FIPS Ciphers: DISABLED
63                      SNI: DISABLED
64                      OCSP Stapling: DISABLED
65                      HSTS: DISABLED
66                      HSTS IncludeSubDomains: NO
67                      HSTS Max-Age: 0
68                      DTLSv1: ENABLED
69                      Send Close-Notify: YES
70                      Strict Sig-Digest Check: DISABLED
71                      Zero RTT Early Data: DISABLED
72                      DHE Key Exchange With PSK: NO
73                      Tickets Per Authentication Context: 1
74                      DTLS profile name: nsdtls_default_profile
75
76                      ECC Curve: P_256, P_384, P_224, P_521
77
78      1)            CertKey Name: servercert               Server
           Certificate
79
80      1)            Cipher Name: DEFAULT
81                      Description: Default cipher list with encryption
                           strength >= 128bit
82
83      2)            Cipher Name: ALL
84                      Description: All ciphers supported by NetScaler,
                           excluding NULL ciphers
85       Done
86
87  sh service svc_dtls
88
89                      svc_dtls (10.102.59.190:4433) - DTLS
90                      State: UP
91                      Last state change was at Fri Apr 27 07:00:26 2018
92                      Time since last state change: 0 days, 00:00:22.790
93                      Server Name: s1
```



```
94                        Server ID : None                  Monitor Threshold
                              : 0
95                        Max Conn: 0        Max Req: 0           Max
                              Bandwidth: 0 kbits
96                        Use Source IP: NO
97                        Client Keepalive(CKA): NO
98                        Access Down Service: NO
99                        TCP Buffering(TCPB): NO
100                       HTTP Compression(CMP): NO
101                       Idle timeout: Client: 120 sec           Server: 120
                              sec
102                       Client IP: DISABLED
103                       Cacheable: NO
104                       SC: OFF
105                       SP: OFF
106                       Down state flush: ENABLED
107                       Monitor Connection Close : NONE
108                       Appflow logging: ENABLED
109                       Process Local: DISABLED
110                       Traffic Domain: 0
111
112     1)            Monitor Name: ping-default
113                                     State: UP               Weight: 1
                                                        Passive: 0
114                                   Probes: 5             Failed [Total
                                          : 0 Current: 0]
115                                    Last response: Success - ICMP echo
                                          reply received.
116                          Response Time: 2.77 millisec
117      Done
118
119 sh ssl service svc_dtls
120
121                       Advanced SSL configuration for Back-end SSL Service
                              svc_dtls:
122                       DH: DISABLED
123                       DH Private-Key Exponent Size Limit: DISABLED
                                    Ephemeral RSA: DISABLED
124                       Session Reuse: ENABLED                Timeout:
                              1800 seconds
125                       Cipher Redirect: DISABLED
126                       ClearText Port: 0
127                       Server Auth: DISABLED
128                       SSL Redirect: DISABLED
129                       Non FIPS Ciphers: DISABLED
```



```
130                     SNI: DISABLED
131                     OCSP Stapling: DISABLED
132                     DTLSv1: ENABLED
133                     Send Close-Notify: YES
134                     Strict Sig-Digest Check: DISABLED
135                     Zero RTT Early Data: ???
136                     DHE Key Exchange With PSK: ???
137                     Tickets Per Authentication Context: ???
138                     DTLS profile name: nsdtls_default_profile
139                     ECC Curve: P_256, P_384, P_224, P_521
140     1)            Cipher Name: DEFAULT_BACKEND
141                     Description: Default cipher list for Backend SSL
                           session
142      Done
143
144
145  > sh dtlsProfile nsdtls_default_profile
146  1)  Name: nsdtls_default_profile
147      PMTU Discovery: DISABLED
148      Max Record Size: 1459 bytes
149      Max Retry Time: 3 sec
150      Hello Verify Request: DISABLED
151      Terminate Session: ENABLED
152      Max Packet Count: 120 bytes
153      Max HoldQ Size: 32 datagrams
154      Max bad-MAC Ignore Count: 10
155
156   Done
157  <!--NeedCopy-->
```

DTLS support for IPv6 address

DTLS is supported with IPv6 addresses also. However, to use DTLS with IPv6 addresses the maximum record size must be adjusted in the DTLS profile.

If the default value is used for the maximum record size, the initial DTLS connection might fail. Adjust the maximum record size using a DTLS profile.

DTLS cipher support

By default, a DTLS cipher group is bound when you create a DTLS virtual server or service. DEFAULT_DTLS contains the ciphers that a front-end DTLS entity supports. This group is bound by default when you create a DTLS virtual server. DEFAULT_DTLS_BACKEND contains the ciphers that

are supported to a back-end DTLS entity. This group is bound by default to a DTLS back-end service. DTLS_FIPS contains the ciphers that are supported on the Citrix ADC FIPS platform. This group is bound by default to a DTLS virtual server or service created on a FIPS platform.

DTLS cipher support on Citrix ADC VPX, MPX/SDX (N2 and N3 based) appliances

How to read the tables:

Unless a build number is specified, a cipher suite is supported for all builds in a release.

Example:

- **10.5, 11.0, 11.1, 12.0, 12.1, 13.0**: All builds of 10.5, 11.0, 11.1, 12.0, 12.1, 13.0 releases.
- **-NA-**: not applicable.

DTLS cipher support on Citrix ADC VPX, MPX/SDX (N2 and N3 based) appliances

Cipher Suite Name	Hex Code	Wireshark Cipher Suite Name	Builds Supported (front end)	Builds Supported (back end)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AI	11.0, 11.1, 12.0, 12.1, 13.0	12.0, 12.1, 13.0
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.0, 11.1, 12.0, 12.1, 13.0	12.0, 12.1, 13.0
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_D	11.0, 11.1, 12.0, 12.1, 13.0	-NA-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.0, 11.1, 12.0, 12.1, 13.0	12.0, 12.1, 13.0
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.0, 11.1, 12.0, 12.1, 13.0	-NA-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	11.0, 11.1, 12.0, 12.1, 13.0	-NA-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1, 13.0	12.1, 13.0
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0	12.1, 13.0

Cipher Suite Name	Hex Code	Wireshark Cipher Suite Name	Builds Supported (front end)	Builds Supported (back end)
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1, 13.0	-NA-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0	12.1, 13.0
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1, 13.0	12.1, 13.0

To view the list of default ciphers supported on the front end, at the command prompt, type:

```
show ssl cipher DEFAULT_DTLS
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0039
6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x0033
7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```


To view the list of default ciphers supported on the back end, at the command prompt, type:

```
show ssl cipher DEFAULT_DTLS_BACKEND
1)  Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0039
6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x0033
7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

DTLS cipher support on the Citrix ADC MPX 14000 FIPS platform

Note: Enlightened Data Support (EDT) is supported on the FIPS platform if the following conditions are met:

- UDT MSS value set on StoreFront is 900.
- Windows client version is 4.12 or later.
- DTLS enabled VDA version is 7.17 or later.
- Non-DTLS VDA version is 7.15 LTSR CU3 or later.

How to read the tables:

Unless a build number is specified, a cipher suite is supported for all builds in a release.

Example:

- **10.5, 11.0, 11.1, 12.0, 12.1, 13.0**: All builds of 10.5, 11.0, 11.1, 12.0, 12.1, 13.0 releases.

- **-NA-**: not applicable.

Cipher suite Name	Hex Code	Wireshark Cipher suite Name	Builds Supported (front end)	Builds Supported (back end)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AI	11.0, 11.1, 12.0, 12.1–49.x, 13.0	12.0, 12.1–49.x, 13.0
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.0, 11.1, 12.0, 12.1–49.x, 13.0	12.0, 12.1–49.x, 13.0
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_D	11.0, 11.1, 12.0, 12.1–49.x, 13.0	-NA-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.0, 11.1, 12.0, 12.1–49.x, 13.0	12.0, 12.1–49.x, 13.0
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.0, 11.1, 12.0, 12.1–49.x, 13.0	-NA-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	11.0, 11.1, 12.0, 12.1–49.x, 13.0	-NA-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1–49.x, 13.0	12.1–49.x, 13.0
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0	12.1–49.x, 13.0
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1–49.x, 13.0	-NA-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0	12.1–49.x, 13.0
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1–49.x, 13.0	12.1–49.x, 13.0

To view the list of default ciphers supported on a Citrix ADC FIPS appliance, at the command prompt, type:

```
1 show ssl cipher DTLS_FIPS
```



```
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA  Priority : 5
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
6)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 6
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

DTLSv1.2 cipher support on the front-end VPX appliances, MPX/SDX (Coleto and N3 based) appliances

The following table lists the additional ciphers supported for the DTLSv1.2 protocol.

Cipher suite Name	Hex Code	Wireshark Cipher suite Name	Builds Supported (VPX front end)	Builds Supported (Coleto based)	Builds Supported (N3 based)
TLS1.2-AES256-GCM-SHA384	0x009d	TLS_RSA_WITH	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-AES128-GCM-SHA256	0x009c	TLS_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384	0xc030	TLS_ECDHE_RS	13.0–47.x	13.0–52.x	13.0–58.x

Cipher suite Name	Hex Code	Wireshark Cipher suite Name	Builds Supported (VPX front end)	Builds Supported (Coleto based)	Builds Supported (N3 based)
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256	0xc02f	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-DHE-RSA-AES256-GCM-SHA384	0x009f	TLS_DHE_RSA_	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-DHE-RSA-AES128-GCM-SHA256	0x009e	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-AES-256-SHA256	0x003d	TLS_RSA_WITH	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-AES-128-SHA256	0x003c	TLS_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-ECDHE-RSA-AES-256-SHA384	0xc028	TLS_ECDHE_RS	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-ECDHE-RSA-AES-128-SHA256	0xc027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-DHE-RSA-AES-256-SHA256	0x006b	TLS_DHE_RSA_	13.0–47.x	13.0–52.x	13.0–58.x
TLS1.2-DHE-RSA-AES-128-SHA256	0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x	13.0–52.x	13.0–58.x

Support for Intel Coleto SSL chip based platforms

September 14, 2021

The following appliances ship with Intel Coleto chips:

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 15000
- MPX/SDX 15000-50G
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

Use the 'show hardware' command to identify whether your appliance has Coleto (COL) chips.

```
> sh hardware

    Platform: NSMPX-8900 8*CPU+4*F1X+6*E1K+1*E1K+1*COL 8955 30010
    Manufactured on: 10/18/2016
    CPU: 2100MHZ
    Host Id: 0
    Serial no: CRAC5CR8UA
    Encoded serial no: CRAC5CR8UA
 Done
<!--NeedCopy-->
```

Note: Secure renegotiation is supported on the back end for these platforms.

Limitations:

- DH 512 cipher is not supported.
- SSLv3 protocol is not supported.
- Hardware security module (HSM) is not supported.
- GnuTLS is not supported.
- ECDSA certificates with ECC curves P_224 and P521 are not supported (Not supported on platforms with Cavium chips also.)
- DNSSEC offload is not supported. (DNSSEC is supported in software but offload to hardware is not supported.)

View the SSL chip utilization on Citrix ADC MPX platforms

From release 13.0 build 47.x, you can view the SSL chip utilization on MPX platforms that ship with Intel Coleto chips. This feature is not supported on the SDX platform and on an MPX cluster.

At the command prompt, type:

```
> stat ssl


SSL Summary


# SSL cards present                              4

# SSL cards UP                                   4

SSL engine status                                1

SSL sessions (Rate)                            0

SSL Crypto Utilization Asym (%)                  67

SSL Crypto Utilization Symm (%)                  19
<!--NeedCopy-->
```

MPX 9700/10500/12500/15500 FIPS appliances

September 14, 2021

Important! The MPX 9700/10500/12500/15500 FIPS platform has reached end of life.

The Federal Information Processing Standard (FIPS), issued by the US National Institute of Standards and Technologies, specifies the security requirements for a cryptographic module used in a security system. The Citrix ADC FIPS appliance complies with the second version of this standard, FIPS-140-2.

Note: Henceforth, all references to FIPS imply FIPS-140-2.

The FIPS appliance is equipped with a tamper-proof (tamper-evident) cryptographic module—and a Cavium CN1620-NFBE3-2.0-G on the MPX 9700/10500/12500/15500 FIPS appliances—designed to comply with the FIPS 140-2 Level-2 specifications. The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser can perform operations on the keys stored inside the HSM.

The following table summarizes the differences between standard Citrix ADC and Citrix ADC FIPS appliances.

Setting	Citrix ADC appliance	Citrix ADC FIPS appliance
Key storage	On the hard disk	On the FIPS card
Cipher support	All ciphers	FIPS approved ciphers
Accessing keys	From the hard disk	Not accessible

Configuring a FIPS appliance involves configuring the HSM immediately after completing the generic configuration process. You then create or import a FIPS key. After creating a FIPS key, you must export it for backup. You might also need to export a FIPS key so that you can import it to another appliance. For example, configuring FIPS appliances in a high availability setup requires transferring the FIPS key from the primary node to the secondary node immediately after completing the standard high availability setup.

You can upgrade the firmware version on the FIPS card from version 4.6.0 to 4.6.1. You can also reset an HSM that has been locked to prevent unauthorized logon. Only FIPS approved ciphers are supported on a Citrix ADC FIPS appliance.

HSM configuration

Before you can configure the HSM of your Citrix ADC FIPS appliance, you must complete the initial hardware configuration. For more information about MPX appliances, see Initial Configuration. For information about SDX appliances, click here.

Configuring the HSM of your Citrix ADC FIPS appliance erases all existing data on the HSM. To configure the HSM, you must be logged on to the appliance as the superuser. The HSM is preconfigured with default values for the Security Officer (SO) password and User password, which you use to configure the HSM or reset a locked HSM. The maximum length allowed for the password is 14 alphanumeric characters. Symbols are not allowed.

Important: Run the `set ssl fips` command only after first resetting the FIPS card and restarting the MPX FIPS appliance.

Although the FIPS appliance can be used with the default password values, you must modify them before using it. The HSM can be configured only when you log on to the appliance as the superuser and specify the SO and User passwords.

Important: Due to security constraints, the appliance does not provide a means for retrieving the SO password. Store a copy of the password safely. If you need to reinitialize the HSM, you need to specify this password as the old SO password.

Before initializing the HSM, you can upgrade to the latest build of the software. To upgrade to the latest build, see Upgrading or Downgrading the System Software.

After upgrading, verify that the /nsconfig/fips directory has been successfully created on the appliance.

Configure the HSM on the MPX 9700/10500/12500/15500 FIPS platform by using the CLI

After logging on to the appliance as the superuser and completing the initial configuration, at the command prompt, type the following commands to configure the HSM and verify the configuration:

```
show ssl fips

reset ssl fips

reboot

set ssl fips -initHSM Level-2 <newSOpassword> <oldSOpassword> <
   userPassword> [-hsmLabel <string>]

save ns config

reboot

show ssl fips
<!--NeedCopy-->
```

Example:

```
show fips

    FIPS Card is not configured
    Done
    reset fips
    reboot
    Are you sure you want to restart NetScaler (Y/N)? [N]:y

    set ssl fips -initHSM Level-2 sopin12345 so12345 user123 -hsmLabel
       cavium

    This command will erase all data on the FIPS card. You must save
       the configuration

    (saveconfig) after executing this command.


    Do you want to continue?(Y/N)y
    Done
```



```
18
19      save ns config
20
21      reboot
22
23      Are you sure you want to restart NetScaler (Y/N)? [N]:y
24
25      show fips
26
27              FIPS HSM Info:
28      HSM Label               : Citrix ADC FIPS
29      Initialization          : FIPS-140-2 Level-2
30      HSM Serial Number       : 2.1G1008-IC000021
31      HSM State               : 2
32      HSM Model               : NITROX XL CN1620-NFBE
33      Firmware Version        : 1.1
34      Firmware Release Date   : Jun04,2010
35      Max FIPS Key Memory     : 3996
36      Free FIPS Key Memory    : 3994
37      Total SRAM Memory       : 467348
38      Free SRAM Memory        : 62564
39      Total Crypto Cores       : 3
40      Enabled Crypto Cores     : 1
41      Done
42
43      Note: If you upgrade the firmware to version 2.2, the firmware
           release date is replaced with the firmware build.
44
45
46
47      > show fips
48
49      FIPS HSM Info:
50
51      HSM Label                 : Citrix ADC FIPS
52      Initialization               : FIPS-140-2 Level-2
53      HSM Serial Number     : 3.0G1235-ICM000264
54      HSM State                 : 2
55      HSM Model                : NITROX XL CN1620-NFBE
56      Hardware Version        : 2.0-G
57      Firmware Version         : 2.2
58      Firmware Build            : NFBE-FW-2.2-130009
59      Max FIPS Key Memory : 3996
60      Free FIPS Key Memory : 3958
61      Total SRAM Memory     : 467348
```



```
62      Free SRAM Memory     : 50524
63      Total Crypto Cores      : 3
64      Enabled Crypto Cores  : 3
65      Done
66  <!--NeedCopy-->
```

Configure the HSM on the MPX 9700/10500/12500/15500 FIPS platform by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the **FIPS Infotab**, click **Reset FIPS**.
3. In the navigation pane, click **System**.
4. In the details pane, click **Reboot**.
5. In the details pane, on the FIPS Info tab, click **Initialize HSM**.
6. In the Initialize HSM dialog box, specify values for the following parameters:
   - Security Officer (SO) Password*—new SO password
   - Old SO Password*—old SO password
   - User Password*—user password
   - Level—initHSM (Currently set to Level2 and cannot be changed)
   - HSM Label—hsmLabel

   *A required parameter
7. Click **OK**.
8. In the details pane, click **Save**.
9. In the navigation pane, click **System**.
10. In the details pane, click **Reboot**.
11. Under FIPS HSM Info, verify that the information displayed is correct for the FIPS HSM that you configured.

Create and transfer FIPS keys

After configuring the HSM of your FIPS appliance, you are ready to create a FIPS key. The FIPS key is created in the appliance's HSM. You can then export the FIPS key to the appliance's CompactFlash card as a secured backup. Exporting the key also enables you to transfer it by copying it to the /flash of another appliance and then importing it into the HSM of that appliance. Enable SIM between two standalone nodes before you export and transfer the keys. In a high availability setup, if one of the nodes is replaced with a new one, you must perform the following steps:

1. Enable SIM between this new appliance and the existing appliance of the high availability setup.
2. Export or import FIPS keys.

Instead of creating a FIPS key, you can import an existing FIPS key or import an external key as a FIPS key. If you are adding a certificate-key pair of 2048 bits on the MPX 9700/10500/12500/15500 FIPS appliances, make sure that you have the correct certificate and key pair.

Note: If you are planning a high availability setup, make sure that the FIPS appliances are configured in a high availability setup before creating a FIPS key.

Create FIPS keys

Before creating a FIPS key, make sure that the HSM is configured.

Specify the key type (RSA or ECDSA) and specify the curve for ECDSA keys.

Create a FIPS key by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Keys tab, click **Add**.
3. In the Create FIPS Key dialog box, specify values for the following parameters:
   - FIPS Key Name*—fipsKeyName
   - Modulus*—modulus
   - Exponent*—exponent

   *A required parameter
4. Click **Create**, and then click **Close**.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you created are correct.

Create a FIPS key by using the CLI

At the command prompt, type the following commands to create a FIPS key and verify the settings:

```
create ssl fipsKey <fipsKeyName> -modulus <positive_integer> [-exponent
    ( 3 | F4 )]

show ssl fipsKey [<fipsKeyName>]
<!--NeedCopy-->
```

Example:


```
    create fipskey Key-FIPS-1 -keytype RSA -modulus 2048 -exponent 3

    show ssl fipsKey Key-FIPS-1

                FIPS Key Name: Key-FIPS-1 Key Type: RSA Modulus: 2048
                     Public Exponent: F4 (Hex: 0x10001)
<!--NeedCopy-->
```

Export FIPS keys

Citrix recommends that you create a backup of any key created in the FIPS HSM. If a key in the HSM is deleted, there is no way to create the same key again, and all the certificates associated with it are rendered useless.

In addition to exporting a key as a backup, you might need to export a key for transfer to another appliance.

The following procedure provides instructions on exporting a FIPS key to the `/nsconfig/ssl` folder on the appliance's CompactFlash and securing the exported key by using a strong asymmetric key encryption method.

Export a FIPS key by using the CLI

At the command prompt, type:

```
export ssl fipsKey <fipsKeyName> -key <string>
<!--NeedCopy-->
```

Example:

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
<!--NeedCopy-->
```

Export a FIPS key by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Keys tab, click **Export**.
3. In the Export FIPS key to a file dialog box, specify values for the following parameters:
   - FIPS Key Name*—fipsKeyName
   - File Name*—key (To put the file in a location other than the default, you can either specify the complete path or click the Browse button and navigate to a location.)

*A required parameter

4. Click **Export**, and then click **Close**.

Import an existing FIPS key

To use an existing FIPS key with your FIPS appliance, you need to transfer the FIPS key from the hard disk of the appliance into its HSM.

Note: To avoid errors when importing a FIPS key, make sure that the name of the key imported is the same as the original key name when it was created.

Import a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the CLI

At the command prompt, type the following commands to import a FIPS key and verify the settings:

```
1 -  import ssl fipsKey <fipsKeyName> -key <string> -inform SIM -exponent
     (F4 | 3)
2 -  show ssl fipskey <fipsKeyName>
3 <!--NeedCopy-->
```

Example:

```
1 import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
2 show ssl fipskey key-FIPS-2
3 FIPS Key Name: Key-FIPS-2 Modulus: 2048   Public Exponent: F4 (Hex
     value 0x10001)
4 <!--NeedCopy-->
```

Import a FIPS key by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Keys tab, click **Import**.
3. In the Import as a FIPS Key dialog box, select FIPS key file and set values for the following parameters:
   - FIPS Key Name*
   - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.
   - Exponent*

   *A required parameter
4. Click **Import**, and then click **Close**.

5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you imported are correct.

Import an external key

You can transfer FIPS keys that are created within the Citrix ADC appliance's HSM. You can also transfer external private keys (such as keys created on a standard Citrix ADC, Apache, or IIS) to a Citrix ADC FIPS appliance. External keys are created outside the HSM, by using a tool such as OpenSSL. Before importing an external key into the HSM, copy it to the appliance's flash drive under `/nsconfig/ssl`.

On the MPX 9700/10500/12500/15500 FIPS appliances, the -exponent parameter in the **`import`** `ssl fipskey` command is not required while importing an external key. The correct public exponent is detected automatically when the key is imported, and the value of the -exponent parameter is ignored.

The Citrix ADC FIPS appliance does not support external keys with a public exponent other than 3 or F4.

You do not need a wrap key on the MPX 9700/10500/12500/15500 FIPS appliances.

You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 FIPS appliance. To import the key you need to first decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

```
1 openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>
2 <!--NeedCopy-->
```

Note: If you import an RSA key as a FIPS key, Citrix recommends that you delete the RSA key from the appliance for security purposes.

Import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the CLI

1. Copy the external key to the appliance's flash drive.
2. If the key is in .pfx format, you must first convert it to PEM format. At the command prompt, type:

```
1 convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx
     file name> -password <password>
2 <!--NeedCopy-->
```

3. At the command prompt, type the following commands to import the external key as a FIPS key and verify the settings:


```
import ssl fipsKey <fipsKeyName> -key <string> -informPEM
show ssl fipskey<fipsKeyName>
<!--NeedCopy-->
```

Example:

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx

import fipskey Key-FIPS-2 -key iis.pem -inform PEM

show ssl fipskey key-FIPS-2

FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0
   x10001)
<!--NeedCopy-->
```

Import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the GUI

1. If the key is in .pfx format, you must first convert it to PEM format.
   a) Navigate to **Traffic Management > SSL**.
   b) In the details pane, under Tools, click **Import PKCS#12**.
   c) In the Import PKCS12 File dialog box, set the following parameters:
      - Output File Name*
      - PKCS12 File Name*—Specify the .pfx file name.
      - Import Password*
      - Encoding Format

      *A required parameter
2. Navigate to **Traffic Management > SSL > FIPS**.
3. In the details pane, on the FIPS Keys tab, click **Import**.
4. In the Import as a FIPS Key dialog box, select PEM file, and set values for the following parameters:
   - FIPS Key Name*
   - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.

   *A required parameter
5. Click **Import**, and then click **Close**.

6. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you imported are correct.

Configure FIPS on appliances in a high availability setup

September 14, 2021

Important! The MPX 9700/10500/12500/15500 FIPS platform has reached end of life.

You can configure two appliances in a high availability (HA) pair as FIPS appliances.

Prerequisites

- The Hardware Security Module (HSM) must be configured on both the appliances. For more information, see Configure the HSM.
- When using the GUI, ensure that the appliances are already in an HA setup. For more information about configuring an HA setup, see High availability.

Note: Citrix recommends that you use the configuration utility (GUI) for this procedure. If you use the command line (CLI), make sure that you carefully follow the steps as listed in the procedure. Changing the order of steps or specifying an incorrect input file might cause an inconsistency that requires an appliance restart. In addition, if you use the CLI, the `create ssl fipskey` command is not propagated to the secondary node. When you run the command with the same input values for modulus size and exponent on two different FIPS appliances, the keys generated are not the same. Create the FIPS key on one of the nodes and then transfer it to the other node. But if you use the configuration utility to configure FIPS appliances in an HA setup, the FIPS key that you create is automatically transferred to the secondary node. The process of managing and transferring the FIPS keys is known as secure information management (SIM).

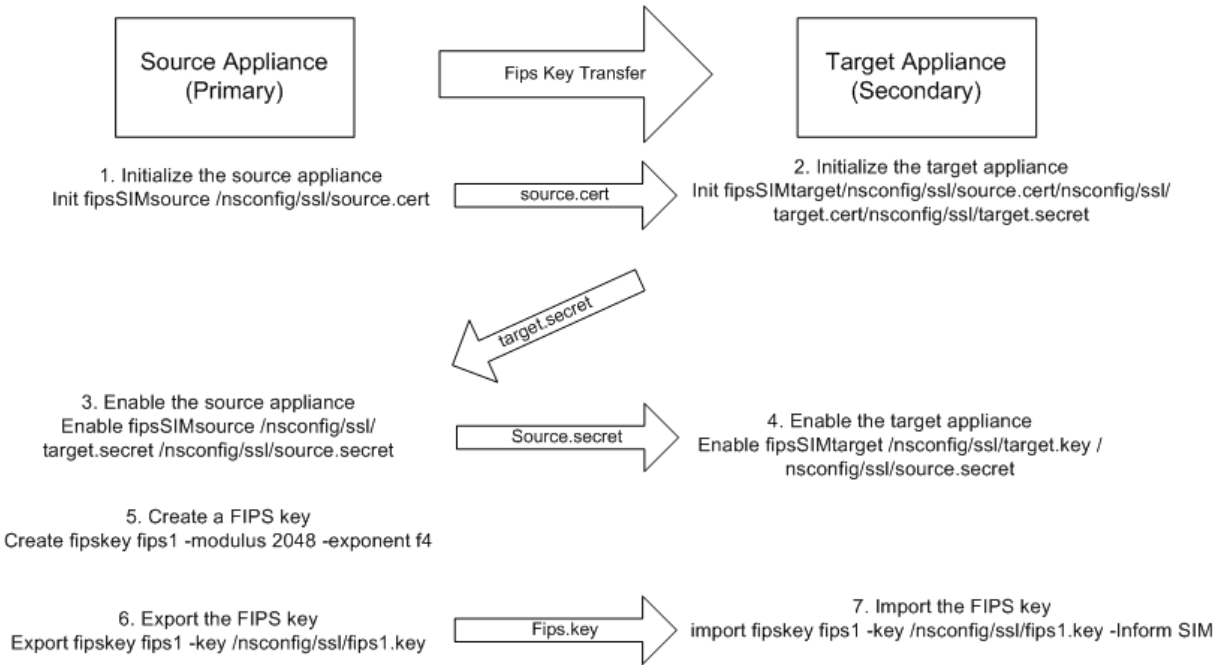
> **Important:** The HA setup must be completed within six minutes. If the procedure fails at any step, do the following:
>
> 1. Restart the appliance or wait for 10 minutes.
> 2. Remove all the files created by the procedure.
> 3. Repeat the HA setup procedure.
>
> Do not reuse existing file names.

In the following procedure, appliance A is the primary node and appliance B is the secondary node.

Configure FIPS on appliances in a high availability setup by using the CLI

The following diagram summarizes the transfer process on the CLI.

Figure 1. Transfer the FIPS key-summary



1. **On appliance A**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance, using the administrator credentials.
3. Initialize appliance A as the source appliance. At the command prompt, type:

```
1 init ssl fipsSIMsource <certFile>
2 <!--NeedCopy-->
```

Example:

```
init fipsSIMsource /nsconfig/ssl/nodeA.cert
```

4. Copy this <certFile> file to appliance B, in the /nconfig/ssl folder.

Example:

```
scp /nsconfig/ssl/nodeA.cert nsroot@198.51.100.10:/nsconfig/ssl
```

5. **On appliance B**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
6. Log on to the appliance, using the administrator credentials.
7. Initialize appliance B as the target appliance. At the command prompt, type:

```
1 init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
2 <!--NeedCopy-->
```

Example:


```
init fipsSIMtarget /nsconfig/ssl/nodeA.cert /nsconfig/ssl/nodeB.key /
nsconfig/ssl/nodeB.secret
```

8. Copy this <targetSecret> file to appliance A.

   Example:

   ```
   scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.20:/nsconfig/
   ssl
   ```

9. **On appliance A**, enable appliance A as the source appliance. At the command prompt, type:

   ```
   enable ssl fipsSIMSource <targetSecret> <sourceSecret>
   <!--NeedCopy-->
   ```

   Example:

   ```
   enable fipsSIMsource /nsconfig/ssl/nodeB.secret /nsconfig/ssl/nodeA.
   secret
   ```

10. Copy this <sourceSecret> file to appliance B.

    Example:

    ```
    scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.10:/nsconfig/
    ssl
    ```

11. **On appliance B**, enable appliance B as the target appliance. At the command prompt, type:

    ```
    enable ssl fipsSIMtarget <keyVector> <sourceSecret>
    <!--NeedCopy-->
    ```

    Example:

    ```
    enable fipsSIMtarget /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeA.secret
    ```

12. **On appliance A**, create a FIPS key, as described in Create a FIPS key.
13. Export the FIPS key to the appliance's hard disk, as described in Export a FIPS key.
14. Copy the FIPS key to the hard disk of the secondary appliance by using a secure file transfer utility, such as SCP.
15. **On appliance B**, import the FIPS key from the hard disk into the HSM of the appliance, as described in Import an existing FIPS key.

Configure FIPS on appliances in a high availability setup by using the GUI

1. On the appliance to be configured as the source (primary) appliance, navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Info tab, click **Enable SIM**.

3. In the **Enable SIM for HA Pair** dialog box, in the **Certificate File Name** text box, type the file name. The file name must contain the path to the location at which the FIPS certificate must be stored on the source appliance.
4. In the **Key Vector File Name** text box, type the file name. The file name must contain the path to the location at which the FIPS key vector must be stored on the source appliance.
5. In the **Target Secret File Name** text box, type the location for storing the secret data on the target appliance.
6. In the **Source Secret File Name** text box, type the location for storing the secret data on the source appliance.
7. Under **Secondary System Login Credential**, enter the values for **User Name** and **Password**.
8. Click **OK**. The FIPS appliances are now configured in HA mode.

Note: After configuring the appliances in HA, create a FIPS key, as described in Create a FIPS key. The FIPS key is automatically transferred from the primary to the secondary appliance.

Update the firmware to version 2.2 on a FIPS card

September 14, 2021

Important! The MPX 9700/10500/12500/15500 FIPS platform has reached end of life.

FIPS firmware version 2.2 supports TLS protocol versions 1.1 and 1.2. From the command line, you can update the firmware version of the FIPS card of a Citrix ADC MPX 9700/10500/12500/15500 FIPS appliance from version 1.1 to version 2.2.

For successful SIM key propagation from primary to secondary in a high availability (HA) pair, the Cavium firmware version on each appliance must be identical. Perform the firmware update on the secondary appliance first. If performed on the primary appliance first, the long-running update process causes a failover.

Limitations

- Secure renegotiation is supported only on SSL virtual servers and front-end SSL services.
- Creating a certificate signing request by using a key that was created on firmware version 1.1 and updated to firmware version 2.2 fails.
- You cannot create a 1024-bit RSA key on firmware version 2.2. However, if you have imported or created a 1024-bit FIPS key on firmware version 1.1 and you then update to firmware version 2.2, you can use that FIPS key on firmware version 2.2.
- Only 2048-bit RSA keys are supported.

- 4096-bit client certificate is not supported (if client authentication is enabled on the back-end server).
- Secure renegotiation using the SSLv3 protocol is not supported.
- After you upgrade the firmware, TLSv1.1 and TLSv1.2 are disabled by default on the existing virtual server, internal, front end, and back-end services. To use TLS 1.1/1.2, you must explicitly enable these protocols, on the SSL entities, after the upgrade.
- FIPS keys that are created in firmware version 2.2 are not available if you downgrade the firmware to version 1.1.

Prerequisites

Download the following files from the download page on www.citrix.com. The files must be stored in the /var/nsinstall directory on the appliance.

- FW 2.2 File: FW-2.2-130013
- FW 2.2 Signature File: FW-2.2-130013.sign

FW-2.2-130013 is the recommended firmware version. It includes fixes to improve DRBG.

Update the FIPS firmware to version 2.2 on a standalone appliance

1. Log on to the appliance by using the administrator credentials.
2. At the prompt, type the following command to confirm that the FIPS card is initialized.

```
show fips

FIPS HSM Info:
HSM Label       : Citrix ADC FIPS
Initialization      : FIPS-140-2 Level-2
HSM Serial Number   : 3.0G1235-ICM000264
HSM State       : 2
HSM Model       : NITROX XL CN1620-NFBE

Hardware Version    : 2.0-G
Firmware Version    : 1.1
Firmware Release Date   : Jun04,2010

Max FIPS Key Memory : 3996
Free FIPS Key Memory    : 3992
Total SRAM Memory   : 467348
Free SRAM Memory    : 62512
Total Crypto Cores  : 3
```



```
Enabled Crypto Cores    : 1
Done
<!--NeedCopy-->
```

3. Save the configuration. At the prompt, type:

```
save config
<!--NeedCopy-->
```

4. Perform the update. At the prompt, type:

```
update ssl fips -fipsFW <path to the extracted contents>/CN16XX-
   NFBE-FW-2.2-1300013
<!--NeedCopy-->
```

   Press Y when the following prompt appears:

```
This command will update compatible version of the FIPS firmware.
     You must save the current configuration (saveconfig) before
   executing this command. You must reboot the system after
   execution of this command, for the firmware update to take
   effect. Do you want to continue?(Y/N)Y

Done
<!--NeedCopy-->
```

Note: You only need to specify the firmware file, because the firmware signature file is placed in the same location.

The update takes up to 10 seconds. The update command is blocking, which means that no other actions are performed until the command finishes. The command prompt reappears when execution of the command is completed.

1. Restart the appliance. At the prompt, type:

```
reboot

Are you sure you want to restart NetScaler (Y/N)? [N]:Y
<!--NeedCopy-->
```

2. Verify that the update is successful. At the prompt, type:

```
show fips
<!--NeedCopy-->
```

   The firmware version displayed in the output must be 2.2. For example:


```
sh fips
    FIPS HSM Info:
        HSM Label       : Citrix ADC FIPS
        Initialization      : FIPS-140-2 Level-2
        HSM Serial Number   : 2.1G1207-IC002429
        HSM State       : 2
        HSM Model       : NITROX XL CN1620-NFBE

        Hardware Version    : 2.0-G
        Firmware Version    : 2.2
        Firmware Build          : NFBE-FW-2.2-130013
        Max FIPS Key Memory : 3996
        Free FIPS Key Memory    : 3982
        Total SRAM Memory   : 467348
        Free SRAM Memory    : 50472
        Total Crypto Cores  : 3
        Enabled Crypto Cores    : 1
Done
<!--NeedCopy-->
```

Update the FIPS firmware to version 2.2 on appliances in a high availability pair

1. Log on to the secondary node and perform the update as described in “Update the FIPS firmware to version 2.2 on a standalone appliance”.

   Force the secondary node to become primary. At the prompt, type:

   ```
   force failover
   <!--NeedCopy-->
   ```

   Press **Y** at the confirmation prompt.

2. Log on to the new secondary node (old primary) and perform the update as described in “Update the FIPS firmware to version 2.2 on a standalone appliance”.

3. Force the new secondary node to become primary again. At the prompt, type:

   ```
   force failover
   <!--NeedCopy-->
   ```

   Press **Y** at the confirmation prompt.

Update the FIPS firmware to version 1.1 on a standalone appliance

1. Download the nfb_firmware-r1235_100604 and nfb_firmware-r1235_100604.sign files, to the same directory on the appliance, from the download page on www.citrix.com.
2. Log on to the appliance by using the administrator credentials.
3. At the prompt, type:

```
update ssl fips -fipsFW /<full path to the file>/nfb_firmware-
   r1235_100604
<!--NeedCopy-->
```

Reset a locked HSM

September 14, 2021

Important! The MPX 9700/10500/12500/15500 FIPS platform has reached end of life.

The HSM becomes locked (no longer operational) if you change the SO password, restart the appliance without saving the configuration, and make three unsuccessful attempts to change the password. The locking is a security measure for preventing unauthorized access attempts and changes to the HSM settings.

Important: To avoid this situation, save the configuration after initializing the HSM.
If the HSM is locked, you must reset the HSM and restart the appliance to restore the default passwords. You can then use the default passwords to access the HSM and configure it with new passwords. When finished, you must save the configuration and restart the appliance.

Caution: Reset the HSM only if it is locked.

Reset a locked HSM by using the CLI

At the command prompt, type the following commands to reset and reinitialize a locked HSM:

```
reset ssl fips
reboot -warm
set ssl fips -initHSM Level-2 <new SO password> <old SO password> <user
    password> [-hsmLabel <string>]
save ns config
reboot -warm
<!--NeedCopy-->
```

Example:


```
reset fips

reboot -warm

set fips -initHSM Level-2 newsopin123 sopin123 userpin123 -hsmLabel
   NSFIPS

saveconfig

reboot -warm

Note: By default the HSM passwords are preconfigured. The <
   Old_SO_Password> = so12345, <User_Password> = user123, <
   New_SO_Password> = sopin12345, <New_User_Password> = userpin123.
<!--NeedCopy-->
```

Reset a locked HSM by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**
2. In the details pane, on the FIPS Info tab, click Reset FIPS.
3. Configure the HSM, as described in Configuring the HSM.
4. In the details pane, click Save.

MPX 14000 FIPS appliances

October 14, 2021

> **Important:**
>
> - The MPX 9700/10500/12500/15500 FIPS platform has reached end of life.
> - Configuration steps for NetScaler MPX 14000 FIPS and NetScaler MPX 9700/10500/12500/15500 FIPS appliances are different. MPX 14000 FIPS appliances do not use firmware v2.2. A FIPS key created on the Hardware Security Module (HSM) of the MPX 9700 platform cannot be transferred to the HSM of the MPX 14000 platform. The other way round is also not supported. However, if you have imported an RSA key as a FIPS key, you can copy the RSA key to the MPX 14000 platform. Then import it as a FIPS key. Only 2048-bit and 3072-bit keys are supported.

A FIPS appliance is equipped with a tamper-proof (tamper-evident) cryptographic module—a Cavium CNN3560-NFBE-G—designed to comply with the FIPS 140-2 Level-3 specifications (from release 12.0

build 56.x). The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the HSM. The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (`nsroot`) can perform operations on the keys stored inside the HSM.

Before configuring a FIPS appliance, you must check the state of the FIPS card and then initialize the card. Create a FIPS key and server certificate, and add any additional SSL configuration.

For information about the FIPS ciphers supported, see FIPS Approved Algorithms and Ciphers.

For information about configuring FIPS appliances in an HA setup, see Configure FIPS on appliances in an HA setup.

Limitations

1. SSL renegotiation using the SSLv3 protocol is not supported on the back end of an MPX FIPS appliance.
2. 1024-bit and 4096-bit keys and exponent value of 3 aren't supported.
3. 4096-bit server certificate isn't supported.
4. 4096-bit client certificate isn't supported (if client authentication is enabled on the back-end server).

Configure the HSM

Before configuring the HSM on an MPX 14000 FIPS appliance, check the state of your FIPS card to verify that the driver loaded correctly. Then initialize the card.

At the command prompt, type:

```
1 show fips
2
3 FIPS Card is not configured
4
5 <!--NeedCopy-->
```

The message "ERROR: Operation not permitted - no FIPS card present in the system" appears if the driver is not loaded correctly.

Initialize the FIPS card

The appliance must be restarted three times for proper initialization of the FIPS card.

> **Important**
>
> - Verify that the /nsconfig/fips directory has been successfully created on the appliance.
> - Do not save the configuration before you restart the appliance for the third time.

Perform the following steps to initialize the FIPS card:

1. Reset the FIPS card (reset fips).
2. Restart the appliance (reboot).
3. Set the security officer password for partitions 0 and 1, and the user password for partition (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS
).)

   Note: The set or reset command takes more than 60 seconds to run.
4. Save the configuration (saveconfig).
5. Verify that the password encrypted key for the main partition (master_pek.key) has been created in the /nsconfig/fips/ directory.
6. Restart the appliance (reboot).
7. Verify that the password encrypted key for the default partition (default_pek.key) has been created in the /nsconfig/fips/ directory.
8. Restart the appliance (reboot).
9. Verify that the FIPS card is UP (show fips).

Initialize the FIPS card by using the CLI

The set fips command initializes the Hardware Security Module (HSM) on the FIPS card and sets a new security officer password and user password.

Caution: This command erases all data on the FIPS card. You are prompted before proceeding with the command execution. A restart is required before and after running this command for the changes to apply. Save the configuration after running this command and before restarting the appliance.

At the command prompt, type the following commands:

```
reset fips


reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

```



```
This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. Do you want
     to continue?(Y/N)y

<!--NeedCopy-->
```

Note: The following message appears when you run the `set fips` command:

```
This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. [Note: On
    MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
    default, and the -initHSM Level-2 option is internally converted to
    Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

reboot

show fips

        FIPS HSM Info:
                HSM Label               : NetScaler FIPS
                Initialization          : FIPS-140-2 Level-3
                HSM Serial Number       : 3.1G1836-ICM000136
                HSM State               : 2
                HSM Model               : NITROX-III CNN35XX-NFBE
                Hardware Version        : 0.0-G
                Firmware Version        : 1.0
                Firmware Build          : NFBE-FW-1.0-48
                Max FIPS Key Memory     : 102235
                Free FIPS Key Memory    : 102231
                Total SRAM Memory       : 557396
                Free SRAM Memory        : 262780
                Total Crypto Cores      : 63
                Enabled Crypto Cores    : 63

<!--NeedCopy-->
```

Create a FIPS key

You can create a FIPS key on your MPX 14000 FIPS appliance or import an existing FIPS key to the appliance. The MPX 14000 FIPS appliance supports only 2048-bit and 3072-bit keys and an exponent

value of F4 (whose value is 65537). For PEM keys, an exponent is not required. Verify that the FIPS key is created correctly. Create a certificate signing request and a server certificate. Finally, add the certificate-key pair to your appliance.

Specify the key type (RSA or ECDSA). For ECDSA keys, specify only the curve. ECDSA key creation with curve P_256 and P_384 are supported.

> **Note:**
>
> 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.

Create a FIPS key by using the CLI

At the command prompt, type:

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent ( 3 | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

Example1:

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4

show ssl fipskey f1

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (Hex: 0x10001)

<!--NeedCopy-->
```

Example2:

```
> create fipskey f2 -keytype ECDSA -curve P_256

> sh fipskey f2
    FIPS Key Name: f2   Key Type: ECDSA Curve: P_256

<!--NeedCopy-->
```

Create a FIPS key by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Keys tab, click **Add**.
3. In the Create FIPS Key dialog box, specify values for the following parameters:

- FIPS Key Name*—fipsKeyName
- Modulus*—modulus
- Exponent*—exponent

*A required parameter

4. Click **Create**, and then click **Close**.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you created are correct.

Import a FIPS key

To use an existing FIPS key with your FIPS appliance, you need to transfer the FIPS key from the hard disk of the appliance into its HSM.

Note: To avoid errors when importing a FIPS key, ensure that the name of the key imported is the same as the original key name when it was created.

Import a FIPS key by using the CLI

At the command prompt, type:

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
   wrapKeyName <string>] [-iv<string>] -exponent  F4 ]
<!--NeedCopy-->
```

Example:

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4

import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM

<!--NeedCopy-->
```

Verify that the FIPS key is created or imported correctly by running the `show fipskey` command.

```
show fipskey
1)      FIPS Key Name: Key-FIPS-2

<!--NeedCopy-->
```

Import a FIPS key by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.

2. In the details pane, on the FIPS Keys tab, click **Import**.
3. In the Import as a FIPS Key dialog box, select FIPS key file and set values for the following parameters:
   - FIPS Key Name*
   - Key File Name* — To put the file in a location other than the default, specify the complete path or click **Browse** and navigate to a location.
   - Exponent*

   *A required parameter
4. Click **Import**, and then click **Close**.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you imported are correct.

Export a FIPS key

Citrix recommends that you create a backup of any key created in the FIPS HSM. If a key in the HSM is deleted, you cannot create the same key again, and all the certificates associated with it are rendered useless.

In addition to exporting a key as a backup, you might need to export a key for transfer to another appliance.

The following procedure provides instructions on exporting a FIPS key to the `/nsconfig/ssl` folder on the appliance's CompactFlash and securing the exported key by using a strong asymmetric key encryption method.

Export a FIPS key by using the CLI

At the command prompt, type:

```
export ssl fipsKey <fipsKeyName> -key <string>
<!--NeedCopy-->
```

Example:

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
<!--NeedCopy-->
```

Export a FIPS key by using the GUI

1. Navigate to **Traffic Management > SSL > FIPS**.

2. In the details pane, on the FIPS Keys tab, click **Export**.
3. In the Export FIPS key to a file dialog box, specify values for the following parameters:
   - FIPS Key Name*—fipsKeyName
   - File Name*—key (To put the file in a location other than the default, you can either specify the complete path or click the Browse button and navigate to a location.)

   *A required parameter
4. Click **Export**, and then click **Close**.

Import an external key

You can transfer FIPS keys that are created within the Citrix ADC appliance's HSM. You can also transfer external private keys (such as keys created on a standard Citrix ADC, Apache, or IIS) to a Citrix ADC FIPS appliance. External keys are created outside the HSM, by using a tool such as OpenSSL. Before importing an external key into the HSM, copy it to the appliance's flash drive under `/nsconfig/ssl`.

On the MPX 14000 FIPS appliances, the -exponent parameter in the **`import`** `ssl fipskey` command is not required while importing an external key. The correct public exponent is detected automatically when the key is imported, and the value of the -exponent parameter is ignored.

The Citrix ADC FIPS appliance does not support external keys with a public exponent other than 3 or F4.

You do not need a wrap key on the MPX 14000 FIPS appliances.

You cannot import an external, encrypted FIPS key directly to an MPX 14000 FIPS appliance. To import the key you need to first decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

```
openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>
<!--NeedCopy-->
```

Note: If you import an RSA key as a FIPS key, Citrix recommends that you delete the RSA key from the appliance for security purposes.

Import an external key as a FIPS key by using the CLI

1. Copy the external key to the appliance's flash drive.
2. If the key is in .pfx format, you must first convert it to PEM format. At the command prompt, type:

```
convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx
    file name> -password <password>
```



```
<!--NeedCopy-->
```

3. At the command prompt, type the following commands to import the external key as a FIPS key and verify the settings:

```
import ssl fipsKey <fipsKeyName> -key <string> -informPEM
show ssl fipskey<fipsKeyName>
<!--NeedCopy-->
```

Example:

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx

import fipskey Key-FIPS-2 -key iis.pem -inform PEM

show ssl fipskey key-FIPS-2

FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0
   x10001)
<!--NeedCopy-->
```

Import an external key as a FIPS key by using the GUI

1. If the key is in .pfx format, you must first convert it to PEM format.
   a) Navigate to **Traffic Management > SSL**.
   b) In the details pane, under Tools, click **Import PKCS#12**.
   c) In the Import PKCS12 File dialog box, set the following parameters:
      - Output File Name*
      - PKCS12 File Name*—Specify the .pfx file name.
      - Import Password*
      - Encoding Format

      *A required parameter
2. Navigate to **Traffic Management > SSL > FIPS**.
3. In the details pane, on the FIPS Keys tab, click **Import**.
4. In the Import as a FIPS Key dialog box, select PEM file, and set values for the following parameters:
   - FIPS Key Name*
   - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.

   *A required parameter

5. Click **Import**, and then click **Close**.
6. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you imported are correct.

Configure FIPS on appliances in an HA setup

You can configure two appliances in an HA pair as FIPS appliances.

Prerequisites

- The Hardware Security Module (HSM) must be configured on both the appliances. For more information, see Configure the HSM.
- When using the GUI, ensure that the appliances are already in an HA setup. For more information about configuring an HA setup, see High availability.

> **Note:**
>
> Citrix recommends that you use the configuration utility (GUI) for this procedure. If you use the command line (CLI), make sure that you carefully follow the steps as listed in the procedure. Changing the order of steps or specifying an incorrect input file might cause an inconsistency that requires an appliance restart. In addition, if you use the CLI, the `create ssl fipskey` command is not propagated to the secondary node. When you run the command with the same input values for modulus size and exponent on two different FIPS appliances, the keys generated are not the same. Create the FIPS key on one of the nodes and then transfer it to the other node. But if you use the configuration utility to configure FIPS appliances in an HA setup, the FIPS key that you create is automatically transferred to the secondary node. The process of managing and transferring the FIPS keys is known as secure information management (SIM).

Important: The HA setup must be completed within six minutes. If the procedure fails at any step, do the following:

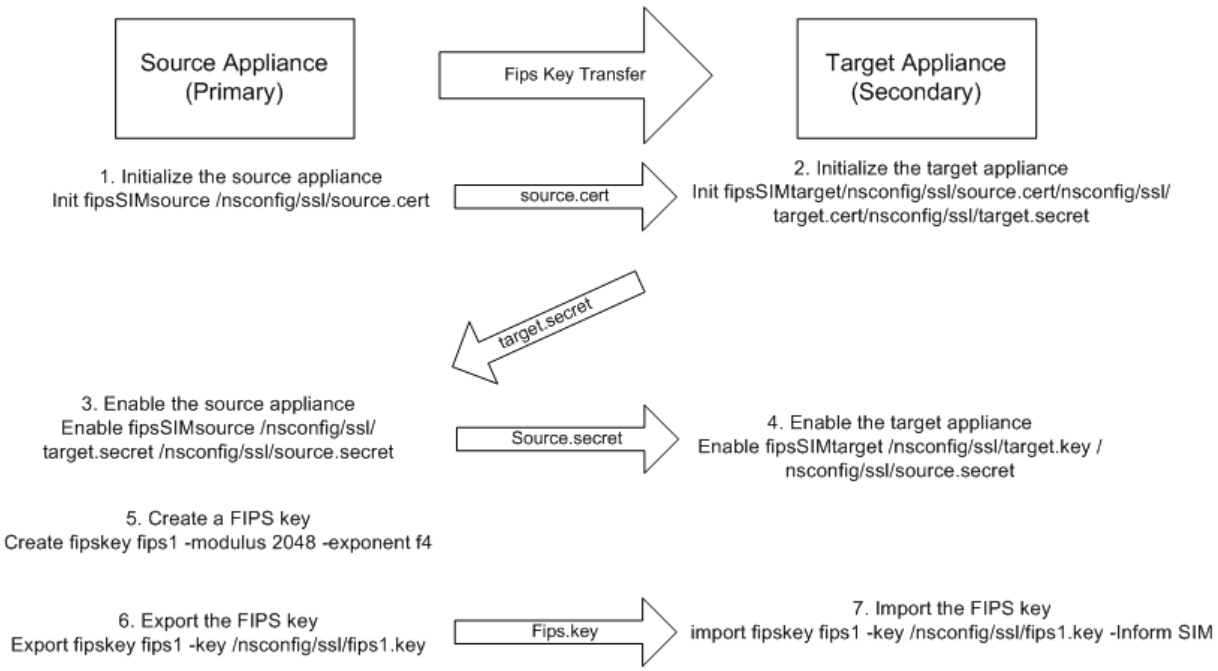
> 1. Restart the appliance or wait for 10 minutes.
> 2. Remove all the files created by the procedure.
> 3. Repeat the HA setup procedure.
>
> Do not reuse existing file names.

In the following procedure, appliance A is the primary node and appliance B is the secondary node.

Configure FIPS on appliances in an HA setup by using the CLI

The following diagram summarizes the transfer process on the CLI.

Figure 1. Transfer the FIPS key-summary



1. **On appliance A**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance, using the administrator credentials.
3. Initialize appliance A as the source appliance. At the command prompt, type:

```
1 init ssl fipsSIMsource <certFile>
2 <!--NeedCopy-->
```

Example:

```
init fipsSIMsource /nsconfig/ssl/nodeA.cert
```

4. Copy this <certFile> file to appliance B, in the /nconfig/ssl folder.

Example:

```
scp /nsconfig/ssl/nodeA.cert nsroot@198.51.100.10:/nsconfig/ssl
```

5. **On appliance B**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
6. Log on to the appliance, using the administrator credentials.
7. Initialize appliance B as the target appliance. At the command prompt, type:

```
1 init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
2 <!--NeedCopy-->
```

Example:

```
init fipsSIMtarget /nsconfig/ssl/nodeA.cert /nsconfig/ssl/nodeB.key /
nsconfig/ssl/nodeB.secret
```


8. Copy this <targetSecret> file to appliance A.

   Example:

   ```
   scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.20:/nsconfig/
   ssl
   ```

9. **On appliance A**, enable appliance A as the source appliance. At the command prompt, type:

   ```
   1 enable ssl fipsSIMSource <targetSecret> <sourceSecret>
   2 <!--NeedCopy-->
   ```

   Example:

   ```
   enable fipsSIMsource /nsconfig/ssl/nodeB.secret /nsconfig/ssl/nodeA.
   secret
   ```

10. Copy this <sourceSecret> file to appliance B.

    Example:

    ```
    scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.10:/nsconfig/
    ssl
    ```

11. **On appliance B**, enable appliance B as the target appliance. At the command prompt, type:

    ```
    1 enable ssl fipsSIMtarget <keyVector> <sourceSecret>
    2 <!--NeedCopy-->
    ```

    Example:

    ```
    enable fipsSIMtarget /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeA.secret
    ```

12. **On appliance A**, create a FIPS key, as described in Create a FIPS key.
13. Export the FIPS key to the appliance's hard disk, as described in Export a FIPS key.
14. Copy the FIPS key to the hard disk of the secondary appliance by using a secure file transfer utility, such as SCP.
15. **On appliance B**, import the FIPS key from the hard disk into the HSM of the appliance, as described in Import a FIPS key.

Configure FIPS on appliances in an HA setup by using the GUI

1. On the appliance to be configured as the source (primary) appliance, navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Info tab, click **Enable SIM**.
3. In the **Enable SIM for HA Pair** dialog box, in the **Certificate File Name** text box, type the file name. The file name must contain the path to the location at which the FIPS certificate must be stored on the source appliance.

4. In the **Key Vector File Name** text box, type the file name. The file name must contain the path to the location at which the FIPS key vector must be stored on the source appliance.
5. In the **Target Secret File Name** text box, type the location for storing the secret data on the target appliance.
6. In the **Source Secret File Name** text box, type the location for storing the secret data on the source appliance.
7. Under **Secondary System Login Credential**, enter the values for **User Name** and **Password**.
8. Click **OK**. The FIPS appliances are now configured in HA mode.

Note: After configuring the appliances in HA, create a FIPS key, as described in Create a FIPS key. The FIPS key is automatically transferred from the primary to the secondary appliance.

Create a certificate signing request by using the CLI

At the command prompt, type:

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

Example:

```
>create certreq f1.req  - fipsKeyName  f1 -countryName US  -stateName CA
     -organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com

<!--NeedCopy-->
```

Create a server certificate by using the CLI

At the command prompt, type:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
     input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
```



```
        input_filename>][-CAkeyForm ( DER | PEM )] [-CAserial <
        output_filename>]
4 <!--NeedCopy-->
```

Example:

```
1 create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
     root.key -CAserial ns-root.srl -days 1000
2
3 <!--NeedCopy-->
```

The preceding example creates a server certificate using a local root CA on the appliance.

Add a certificate-key pair by using the CLI

At the command prompt, type:

```
1 add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
     string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>][-
     expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
     positive_integer>]] [-bundle ( YES | NO )]
2 <!--NeedCopy-->
```

Example:

```
1 add certkey cert1 -cert f1.cert -fipsKey f1
2
3 <!--NeedCopy-->
```

After creating the FIPS key and server certificate, you can add the generic SSL configuration. Enable the features that are required for your deployment. Add servers, services, and SSL virtual servers. Bind the certificate-key pair and the service to the SSL virtual server. Save the configuration.

```
1  enable ns feature SSL LB
2
3  add server s1 10.217.2.5
4
5  add service sr1 s1 HTTP 80
6
7  add lb vserver v1 SSL 10.217.2.172 443
8
9  bind ssl vserver v1  - certkeyName cert1
10
11 bind lb vserver v1 sr1
```



```
12
13 saveconfig
14 <!--NeedCopy-->
```

The basic configuration of your MPX 14000 FIPS appliance is now complete.

For information about configuring secure HTTPS, click Configure FIPS.

For information about configuring secure RPC, click Configure FIPS for the first time.

Update the license on an MPX 14000 FIPS appliance

Any update to the license on this platform requires two reboots.

1. Update the license in the `/nsconfig/license` folder.
2. Restart the appliance.
3. Log on to the appliance.
4. Restart the appliance again.
   Note: Do not add new commands, save the config, or check the system state before the second reboot.
5. Log on to the appliance and ensure that FIPS is initialized by running the `show ssl fips` command.

Support for hybrid FIPS mode on the MPX 14000 FIPS and SDX 14000 FIPS platforms

> **Note:**
>
> This feature is supported only on the new MPX/SDX 14000 FIPS platform containing one primary FIPS card and one or more secondary cards. It is not supported on a VPX platform or a platform containing only one type of hardware card.

On a FIPS platform, the asymmetric and symmetric encryption and decryption are performed on the FIPS card for security reasons. However, you can perform part of this activity (asymmetric) on a FIPS card and offload the bulk encryption and decryption (symmetric) to another card without compromising the security of your keys.

The new MPX/SDX 14000 FIPS platform contains one primary card and one or more secondary cards. If you enable the hybrid FIPS mode, the pre-master secret decryption commands are run on the primary card because the private key is stored on this card. However, the bulk encryption and decryption is offloaded to the secondary card. This offload significantly increases the bulk encryption throughput on an MPX/SDX 14000 FIPS platform as compared to non-hybrid FIPS mode and the existing MPX 9700/10500/12500/15000 FIPS platform. Enabling the hybrid FIPS mode also improves the SSL transaction per second on this platform.

> **Notes:**
>
> - The hybrid FIPS mode is disabled by default to meet the strict certification requirements where all the crypto computation must be done inside a FIPS certified module. Enable the hybrid mode to offload the bulk encryption and decryption to the secondary card.
> - On an SDX 14000 FIPS platform, you must first assign an SSL chip to the VPX instance before you enable the hybrid mode.

Enable hybrid FIPS mode by using the CLI

At the command prompt, type:

```
set SSL parameter -hybridFIPSMode {
 ENABLED|DISABLED }


Arguments

hybridFIPSMode

When this mode is enabled, system will use additional crypto hardware
   to accelerate symmetric crypto operations.

Possible values: ENABLED, DISABLED

Default value: DISABLED
<!--NeedCopy-->
```

Example:

```
    set SSL parameter -hybridFIPSMode ENABLED
    show SSL parameter
    Advanced SSL Parameters
    -----------------------
    . . . . . .
    Hybrid FIPS Mode    : ENABLED
    . . . . . .

<!--NeedCopy-->
```

Enable hybrid FIPS mode by using the GUI

1. Navigate to **Traffic Management > SSL**.

2. In the details pane, under **Settings**, click **Change advanced SSL settings**.
3. In the **Change Advanced SSL Settings** dialog box, select **Hybrid FIPS Mode**.

Limitations:

1. Renegotiation is not supported.
2. The `stat ssl parameter` command on an SDX 14000 platform does not display the correct secondary card utilization percentage. It always displays 0.00% utilization.

```
stat ssl

SSL Summary
# SSL cards present 1
# SSL cards UP  1
# Secondary SSL cards present   4
# Secondary SSL cards UP    4
SSL engine status      1
SSL sessions (Rate)     963
Secondary card utilization (%)     0.00
<!--NeedCopy-->
```

SDX 14000 FIPS appliances

September 14, 2021

A Citrix ADC SDX appliance is a multitenant platform on which you can provision and manage multiple virtual Citrix ADC instances. The SDX appliance addresses cloud computing and multitenancy requirements by allowing a single administrator to configure and manage the appliance and delegate the administration of each hosted instance to tenants.

A Citrix ADC SDX 14030/14060/14080 FIPS appliance provides the capabilities of an SDX appliance with FIPS functionality. It is equipped with a tamper-proof (tamper-evident) cryptographic module—a Cavium CNN3560-NFBE-G—designed to comply with the FIPS 140-2 Level-3 specifications (from release 12.0 build 56.x). The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module. This module is also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (`nsroot`) can perform operations on the keys stored inside the HSM.

A Citrix ADC SDX 14030/14060/14080 FIPS appliance contains one FIPS HSM module with 63 cores. The FIPS HSM module can be partitioned up to a maximum of 32 partitions. The SDX administrator can assign dedicated key storage, cryptographic resources, and number of crypto SSL FIPS cores to

each partition. Keys and resources allocated to a partition are dedicated and secure and any other partition cannot access or share them.

The FIPS HSM partition that you create can be assigned or attached to a VPX instance at the time of provisioning the instance, or later by editing the instance. The FIPS partition created and attached to an instance acts like a virtual HSM module for that instance.

The VPX instances on an SDX 14030/14060/14080 FIPS appliance are assigned a FIPS virtual function (VF) partition, which is treated as an isolated FIPS virtual card or HSM. Therefore, the steps to configure a FIPS partition inside a VPX instance are similar to the steps to configure an MPX FIPS appliance. For compliance details, see the security policy details on the U.S. National Institute of Standards and Technology (NIST) website.

For information about configuring FIPS appliances in a high availability setup, see FIPS appliances in a high availability setup.

> **Important**
>
> Each key includes a private and a public key. As a result, it occupies two key spaces. Therefore, the maximum number of keys is limited to one less than half the key store size.

The SDX 14000 FIPS platform supports a hybrid FIPS mode. This mode allows you to offload part of the encryption and decryption activity to a non-FIPS card. For more information, see Hybrid FIPS mode.

Limitations

September 14, 2021

1. SSL renegotiation using the SSLv3 protocol is not supported on the back end of an SDX FIPS appliance.
2. 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.
3. Backup and restore are not supported.
4. Cluster and administrative domains are not supported.
5. You can attach only one FIPS partition to an instance.
6. An instance with a FIPS partition can be assigned only one CPU core.
7. You can assign either a FIPS partition or an SSL core to an instance, but not both.
8. 4096-bit server certificate is not supported.
9. 4096-bit client certificate is not supported (if client authentication is enabled on the back-end server).

Terminology

September 14, 2021

Zeroize: Reset the HSM. All the data on the HSM is deleted. This step is mandatory before the HSM is initialized.

Initialize: Set the HSM capabilities. The Citrix ADC SDX FIPS appliance complies with FIPS-140-2 level 2. You can create partitions after you initialize the chip.

Key store size: Number of keys that can be stored on a partition. A maximum of 102235 keys can be specified. The maximum number of keys that can be stored is one less than half the number specified. For example, if you specify 100, you can create only 49 keys because one of the keys is the RSA key pair that consumes 2 key stores.

Crypto Core Capacity: Number of crypto cores assigned to a partition. A maximum of 63 cores are available.

SSL Context: Number of concurrent SSL connections that can be created on a partition.

Initialize the HSM

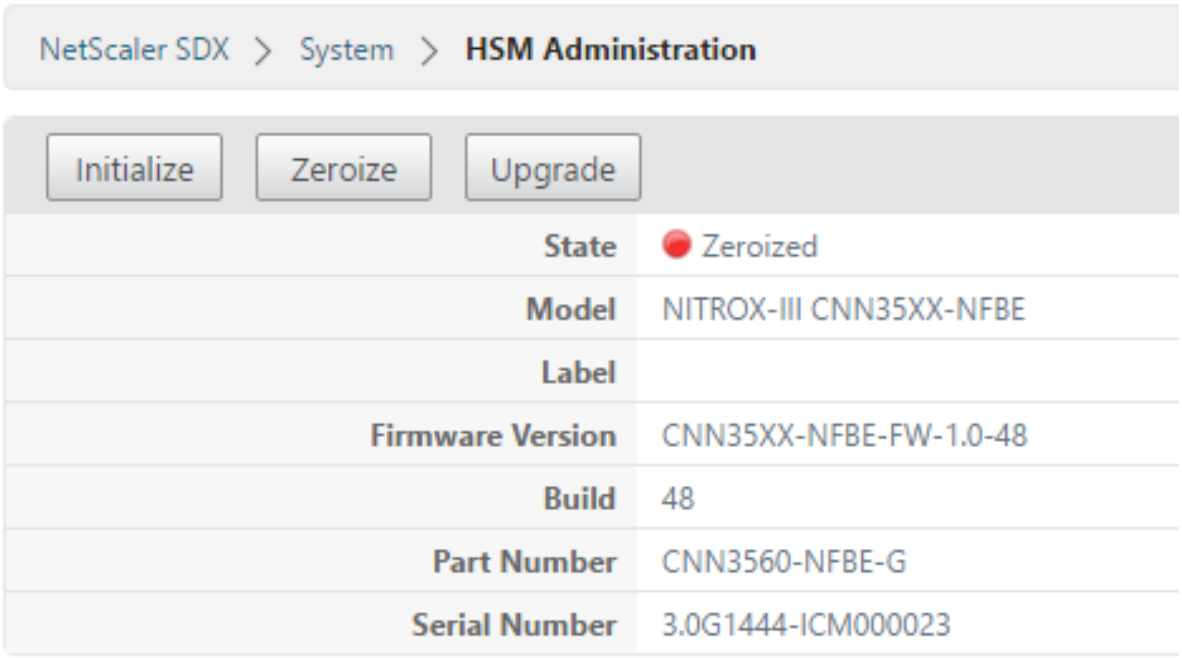
September 14, 2021

Before initializing the HSM, you must first zeroize it.

Zeroize the HSM by using the Management Service

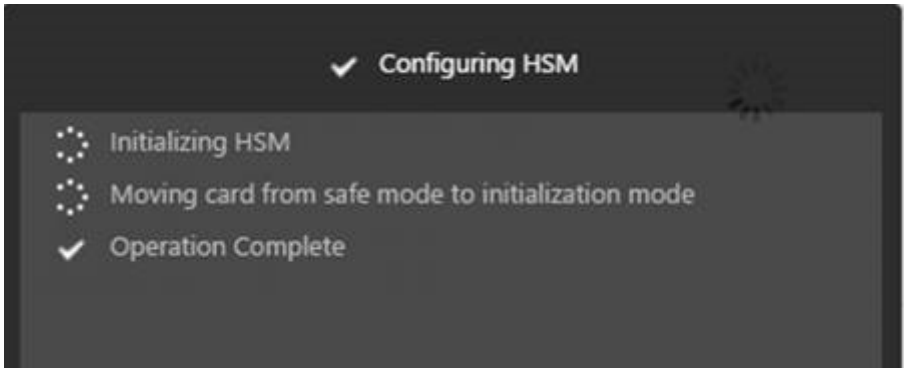
1. Open a browser and log on to the appliance.
2. On the **Configuration** tab, navigate to **System > HSM Administration**, and in the details plane, click **Zeroize**.

All data is wiped from the FIPS chip, and the state appears as “Zeroized.” Any HSM partitions created earlier are deleted.

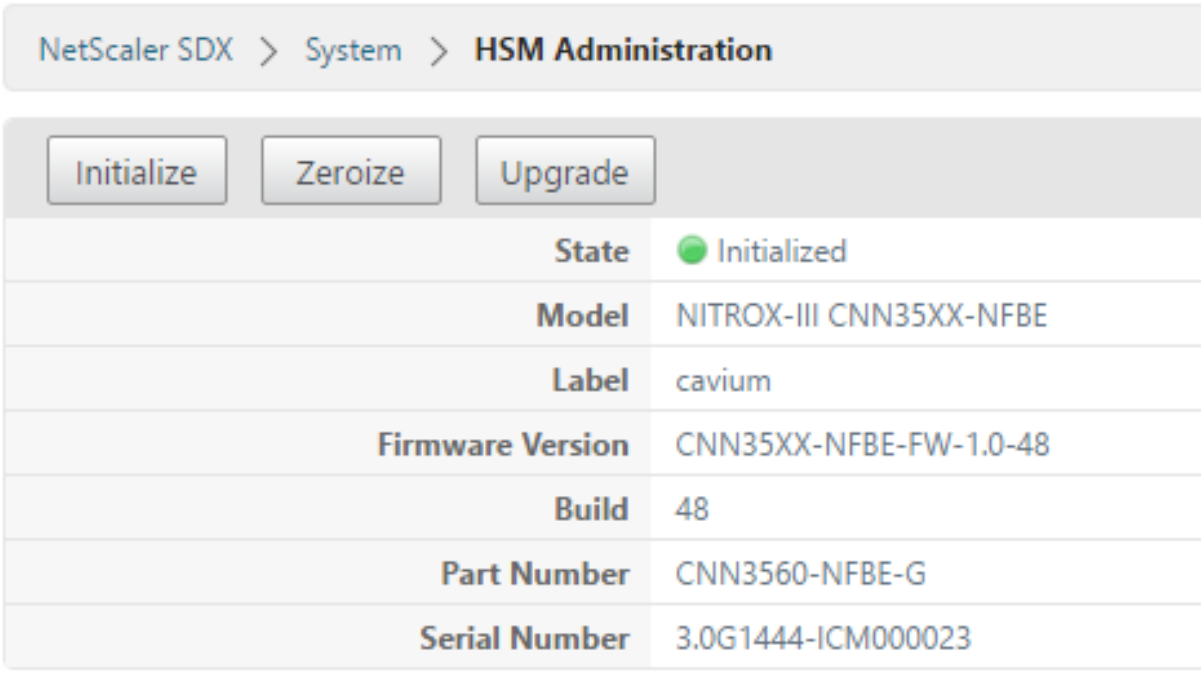


Initialize the HSM by using the Management Service

1. On the **Configuration** tab, navigate to **System > HSM Administration**, and in the details plane, click **Initialize**.
2. Type a new user name, specify a password, and click **OK**.



The card state appears as “Initialized.”



Create partitions

September 14, 2021

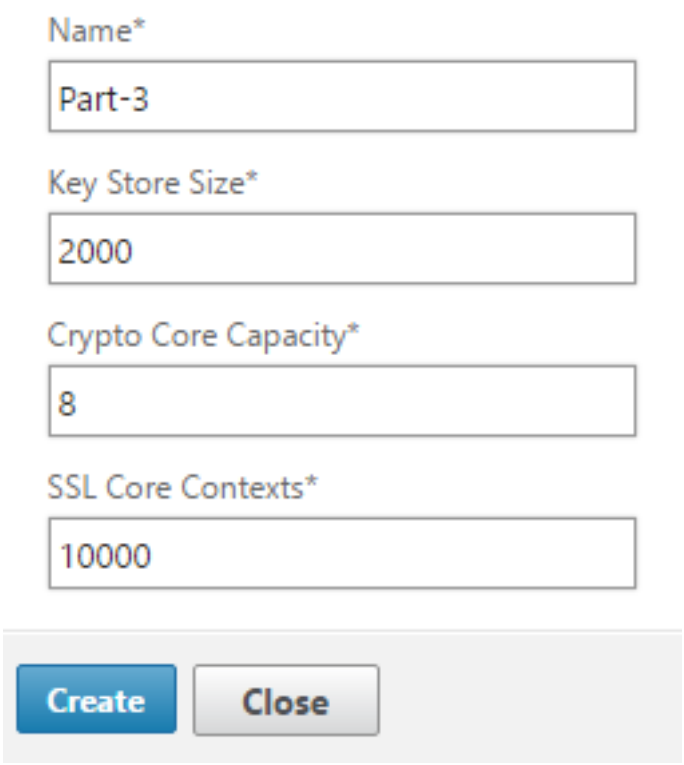
Create partitions for different tenants and specify the cryptographic resources for each partition. Each instance is assigned one partition, and a partition can be assigned to only one instance. Deleting an instance deletes the partition assigned to the instance. As a result, the partition data is also deleted and not left unsecured or accessible later. Number of keys and SSL context assignment depends on your application. For information about the number of cores to assign, see the Citrix ADC data sheet.

> **Important**
>
> After you assign a key store size and cores to an HSM partition, you cannot change them at run time. First detach the partition from the instance.

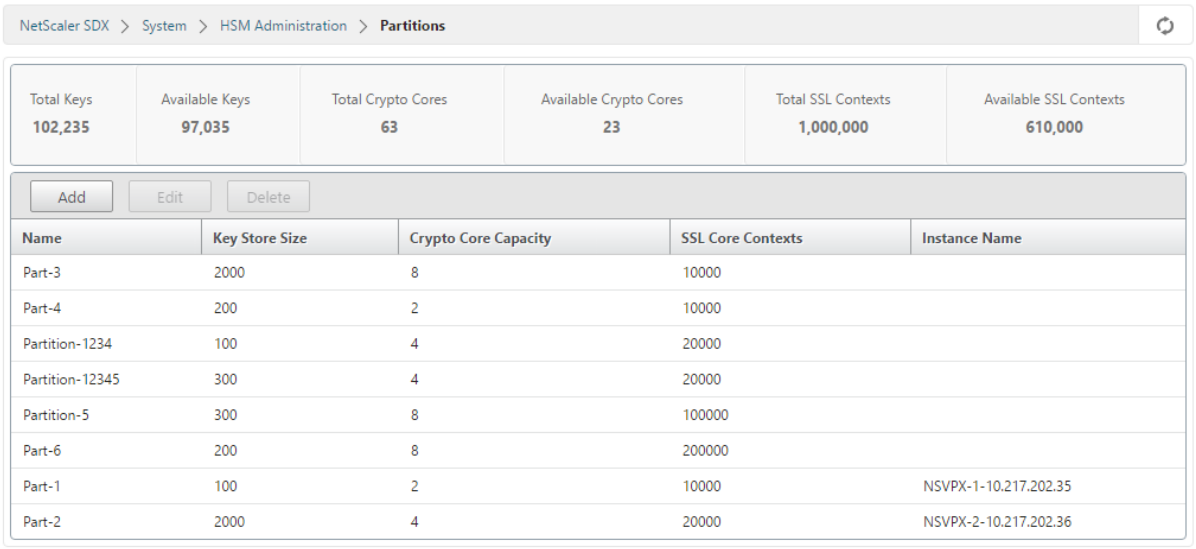
Create a partition by using the Management Service

1. On the Configuration tab, navigate to **System** > **HSM Administration** > **Partitions**, and in the details plane, click **Add**.
2. Specify a name for the partition, and the resources to be assigned to this partition.

3. Click **OK**.



The summary page displays all the partitions that were created. Some partitions are assigned an instance while some are free partitions.



Name	Key Store Size	Crypto Core Capacity	SSL Core Contexts	Instance Name
Part-3	2000	8	10000	
Part-4	200	2	10000	
Partition-1234	100	4	20000	
Partition-12345	300	4	20000	
Partition-5	300	8	100000	
Part-6	200	8	200000	
Part-1	100	2	10000	NSVPX-1-10.217.202.35
Part-2	2000	4	20000	NSVPX-2-10.217.202.36

Provision a new instance or modify an existing instance and assign a partition

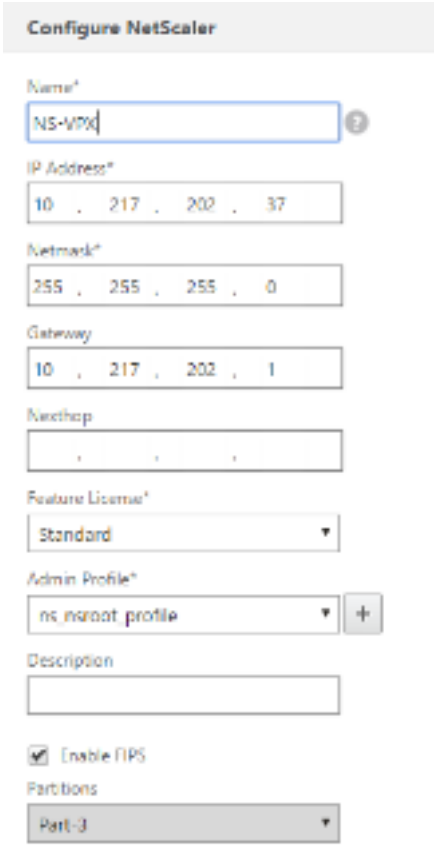
September 14, 2021

After creating the partitions, you must assign them to instances.

> **Important:**
>
> - You can attach only one FIPS partition to an instance.
> - An instance with a FIPS partition can be assigned only one CPU core.

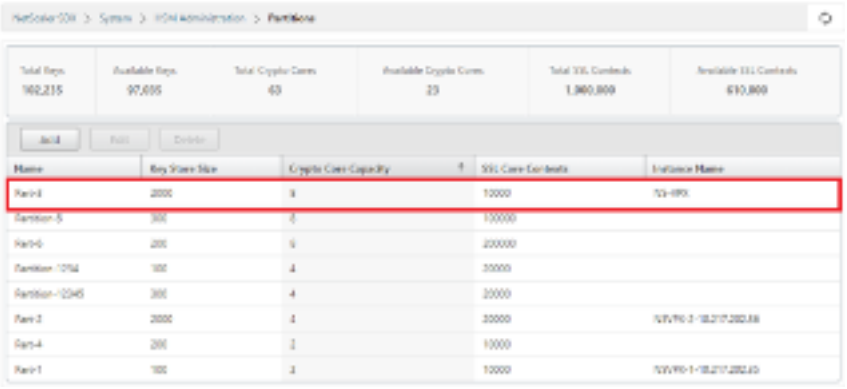
Provision a new instance or modify an existing instance

1. On the Configuration tab, navigate to **NetScaler** > **Instances**, and add or modify an instance.
2. Select **Enable FIPS**, and from the **Partitions** list, select a partition to attach to this instance.



You can verify that the partition is attached to an instance by using either the GUI or the CLI.

In the GUI, navigate to **System** > **HSM Administration** > **Partitions**. The instance name attached to the partition is displayed.



To unassign a FIPS partition , navigate to **NetScaler** > **Instances**. Edit the instance and clear the **Enable FIPS** check box.

In the CLI, at the command prompt, type the following commands:

```
show fips

FIPS Card is not configured
 Done
<!--NeedCopy-->
```

If you see the following output, see the troubleshooting section for debugging.

ERROR: Operation not permitted - no FIPS card present in the system

Configure the HSM for an instance on an SDX 14030/14060/14080 FIPS appliance

October 22, 2021

First check the state of your FIPS card to verify that the driver loaded correctly, and then initialize the card.

At the command prompt, type:

```
show fips

FIPS Card is not configured

Done
<!--NeedCopy-->
```


If the driver is not loaded correctly, the message “ERROR: Operation not permitted - no FIPS card present in the system” appears.

Initialize the FIPS card

> **Important:**
>
> Verify that the `/nsconfig/fips` directory has successfully been created on the appliance.

Do not save the configuration before you restart the appliance for the third time.

Perform the following steps to initialize the FIPS card:

1. Reset the FIPS card (`reset fips`).
2. Restart the appliance (`reboot`).
3. Set the security officer password for partitions 0 and 1, and the user password for partition (`set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS`).

   Note: The set or reset command takes more than 60 seconds to run.
4. Save the configuration (`saveconfig`).
5. Verify that the password encrypted key for the main partition (master_pek.key) has been created in the /nsconfig/fips/ directory.
6. Restart the appliance (`reboot`).
7. Verify that the FIPS card is UP (`show fips`).

Initialize the FIPS card by using the CLI

At the command prompt, type the following commands:

```
reset fips

reboot

set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -
    hsmLabel <string>
<!--NeedCopy-->
```

> **Note**: The following message appears when you run the **set fips** command:


```
This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. [Note: On
    MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
    default, and the -initHSM Level-2 option is internally converted to
    Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

show fips
<!--NeedCopy-->
```

Example:

```
reset fips

Done

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. [Note: On
    MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
    default, and the -initHSM Level-2 option is internally converted to
    Level-3]  Do you want to continue?(Y/N)y

Done

saveconfig

Done

reboot

show fips

    FIPS HSM Info:
    HSM Label : NSFIPS
    Initialization : FIPS-140-2 Level-2
    HSM Serial Number : 3.0G1532-ICM000228
    HSM State : 2
```



```
26      HSM Model : NITROX-III CNN35XX-NFBE
27      Hardware Version : 0.0-G
28      Firmware Version : 1.0
29      Firmware Build : NFBE-FW-1.0-48
30      Max FIPS Key Memory : 1000
31      Free FIPS Key Memory : 1000
32      Total SRAM Memory : 557396
33      Free SRAM Memory : 238088
34      Total Crypto Cores : 4
35      Enabled Crypto Cores : 4
36  Done
37  <!--NeedCopy-->
```

Create a FIPS key for an instance on an SDX 14030/14060/14080 FIPS appliance

September 14, 2021

You can create a FIPS key on your instance or import an existing FIPS key into the instance. An SDX 14030/14060/14080 FIPS appliance supports only 2048-bit and 3072-bit keys and an exponent value of F4. For PEM keys, an exponent is not required. Verify that the FIPS key is created correctly. Create a certificate signing request and a server certificate. Finally, add the certificate-key pair to your instance.

> **Note:**
>
> 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.

Create a FIPS key by using the CLI

At the command prompt, type:

```
1  create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (3
       | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
2  <!--NeedCopy-->
```

Example:

```
1  create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4
2
3  Done
4
5  show ssl fipskey ddvws
```



```
6
7 FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
     Hex: 0x10001)
8
9 Done
10 <!--NeedCopy-->
```

Import a FIPS key by using the CLI

At the command prompt, type:

```
1 import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
     wrapKeyName <string>] [-iv<string>] [-exponent F4 ]
2 <!--NeedCopy-->
```

Example:

```
1 import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
2 Done
3 import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
4 Done
5 <!--NeedCopy-->
```

Verify that the FIPS key is created or imported correctly by running the **show fipskey** command.

```
1 show fipskey
2 1)      FIPS Key Name: Key-FIPS-2
3 Done
4 <!--NeedCopy-->
```

Create a certificate signing request by using the CLI

At the command prompt, type:

```
1 create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
     <string>) [-keyform ( DER | PEM )  {
2  -PEMPassPhrase  }
3  ] -countryName <string> -stateName <string> -organizationName<string>
      [-organizationUnitName <string>] [-localityName <string>] [-
      commonName <string>] [-emailAddress <string>] {
4  -challengePassword  }
5   [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
6 <!--NeedCopy-->
```


Example:

```
create certreq f1.req  - fipsKeyName f1 -countryName US -stateName CA -
    organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com`
`Done
<!--NeedCopy-->
```

Create a server certificate by using the CLI

At the command prompt, type:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
     input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
     input_filename>] [-CAkeyForm ( DER | PEM )] [-CAserial <
     output_filename>]
<!--NeedCopy-->
```

Example:

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
    root.key -CAserial ns-root.srl -days 1000
Done
<!--NeedCopy-->
```

The preceding example creates a server certificate using a local root CA on the appliance.

Add a certificate-key pair by using the CLI

At the command prompt, type:

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod   <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

Example:

```
add certkey cert1 -cert f1.cert -fipsKey f1
Done
```



```
<!--NeedCopy-->
```

After creating the FIPS key and server certificate, you can add the generic SSL configuration. Enable the features that are required for your deployment. Add servers, services, and SSL virtual servers. Bind the certificate-key pair and the service to the SSL virtual server, and save the configuration.

```
enable ns feature SSL LB
Done
add server s1 10.217.2.5
Done
add service sr1 s1 HTTP 80
Done
add lb vserver v1 SSL 10.217.2.172 443
Done
bind ssl vserver v1  -certkeyName cert1
Done
bind lb vserver v1 sr1
Done
saveconfig
Done
<!--NeedCopy-->
```

For information about configuring secure HTTPS and secure RPC, click here.

Upgrade the FIPS firmware on a VPX instance

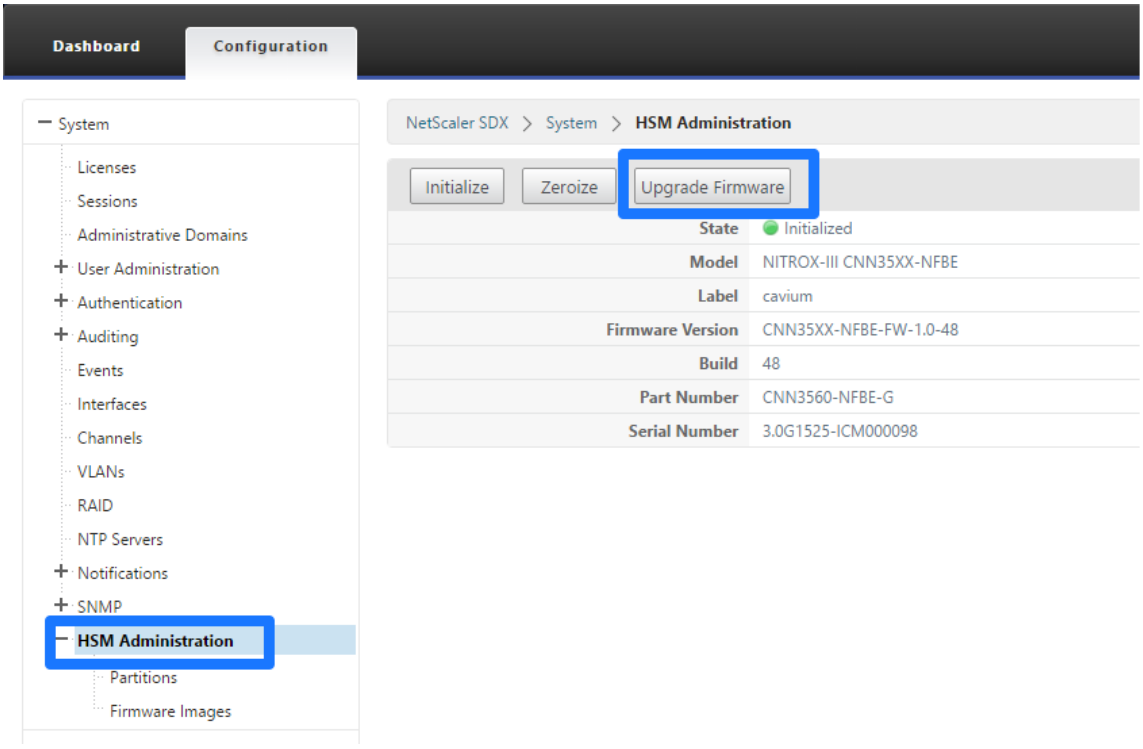
September 14, 2021

FIPS firmware updates are released from time to time. Download the latest firmware from the Citrix download page and upload it to the appliance. The upgrade process might take up to 10 minutes to complete. The instance is restarted after the upgrade.

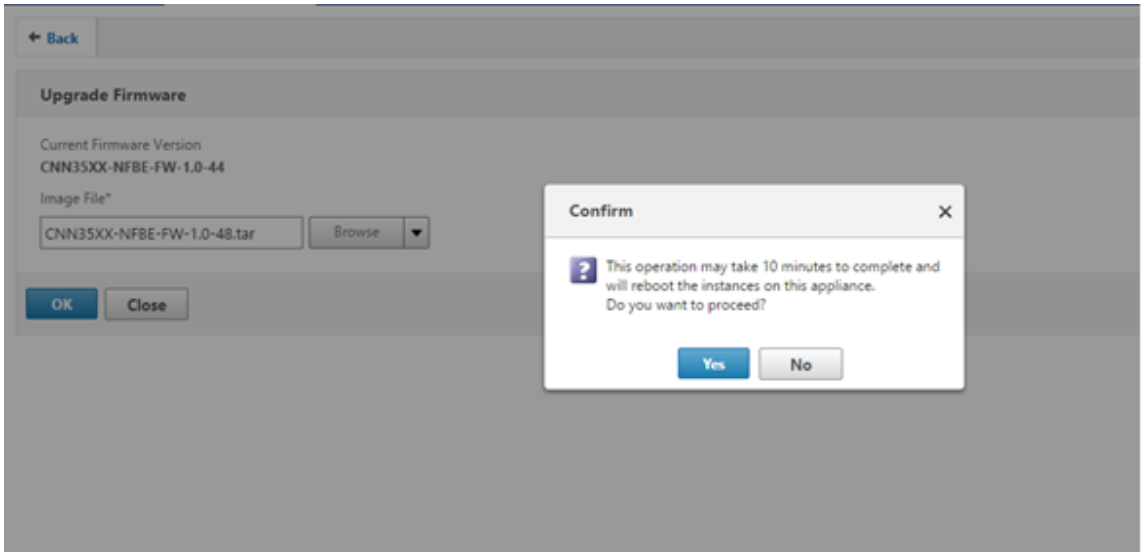
Upgrade the FIPS firmware

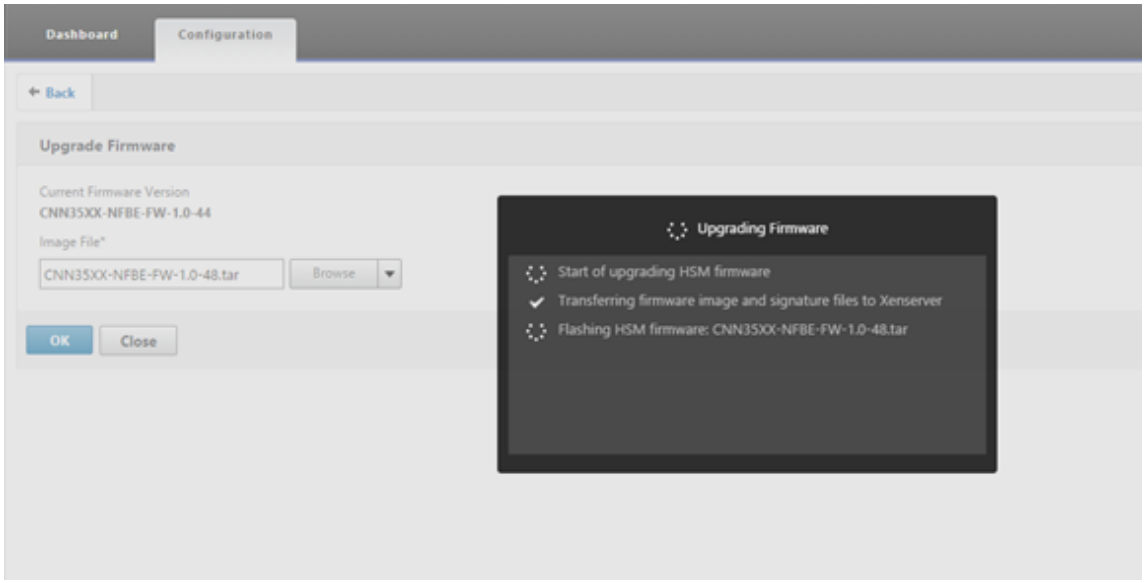
1. Navigate to **System > HSM Administration > Firmware Images**.
2. Select **Upload**.

3. Navigate to the folder that contains the firmware image and select the file.
4. Navigate to **System > HSM Administration**, and select **Upgrade Firmware**.



5. Select the firmware image to upgrade to, and click **OK**.





Support for nShield Connect hardware security module (HSM)

September 14, 2021

A non-FIPS Citrix ADC appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as the HSM. Storing a key in the HSM protects it from physical and software attacks. In addition, the keys are encrypted by using special FIPS approved ciphers.

Only the Citrix ADC MPX 9700/10500/12500/15500 FIPS appliances support a FIPS card. Support for FIPS is not available on other MPX appliances, or on the SDX and VPX appliances. This limitation is addressed by supporting a nShield Connect external HSM on all Citrix ADC MPX, SDX, and VPX appliances

except the MPX 9700/10500/12500/15500 FIPS appliances.

nShield® Connect is an external FIPS-certified network-attached HSM. With an nShield HSM, the keys are securely stored as application key tokens on a remote file server (RFS) and can be reconstituted inside the nShield HSM only.

If you are already using a nShield HSM, you can now use a Citrix ADC to optimize, secure, and control the delivery of all enterprise and cloud services.

Note:

- nShield HSMs comply with FIPS 140-2 Level 3 specifications, while the MPX FIPS appliances comply with level 2 specifications.
- You cannot decrypt the trace while using the nShield HSM. Only the `hardserver` can read the response from the HSM to the Citrix ADC appliance, because it is encrypted.

Supported versions matrix

Citrix ADC Version	nShield Client Version	`Hardserver` Version	nShield Firmware Version
10.5e, 11.0, 11.1, 12.0, 12.1	11.70, 11.72	2.71.2	2.50.16, 2.51.10

Architecture overview

September 14, 2021

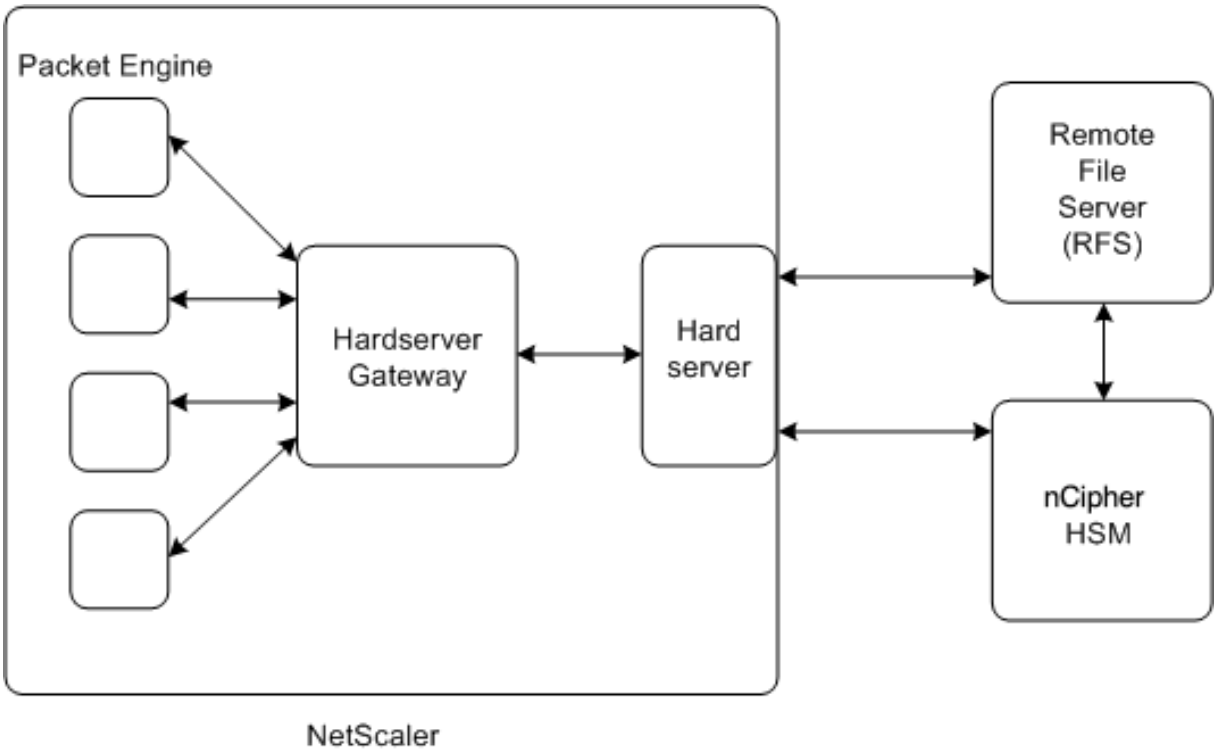
The three entities that are part of a Citrix ADC-Entrust deployment are an Entrust nShield Connect module, a remote file server (RFS), and a Citrix ADC.

The Entrust nShield Connect is a network-attached hardware security module. The RFS is used to configure the HSM and to store the encrypted key files.

`Hardserver`, a proprietary daemon provided by Entrust, is used for communication between the client (ADC), the Entrust HSM, and the RFS. It uses the IMPATH secure communication protocol. A gateway daemon, called the `Hardserver Gateway`, is used to communicate between the Citrix ADC packet engine and the `Hardserver`.

Note: The terms Entrust nShield Connect, Entrust HSM, and HSM are used interchangeably in this documentation.

The following figure illustrates the interaction between the different components.

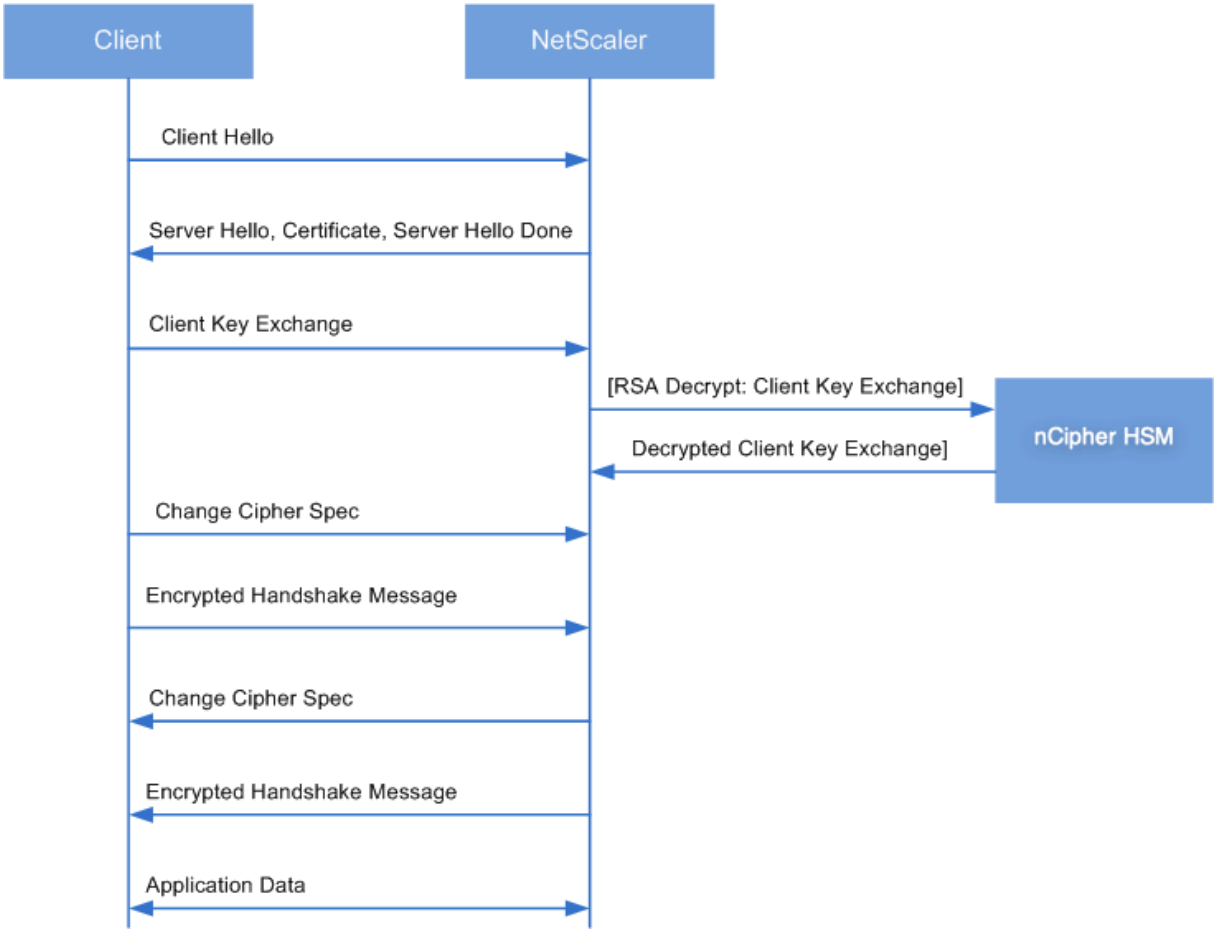


In a typical deployment, the RFS is used to securely store keys generated by the HSM. After the keys are generated, you can securely transfer them to the ADC and then use the GUI or command line to load the keys to the HSM. A virtual server on the ADC uses Entrust to decrypt the client key exchange to complete the SSL handshake. Thereafter, all the SSL operations are performed on the ADC.

Note: The terms keys and application key tokens are used interchangeably in this documentation.

The following figure illustrates the packet flow in the SSL handshake with the Entrust HSM.

Figure 1. SSL Handshake Packets Flow Diagram with Citrix ADC Using Entrust HSM



Note: The communication between the ADC and the HSM uses an Entrust proprietary communication protocol, called IMPATH.

Prerequisites

September 14, 2021

Before you can use an Entrust nShield Connect with a Citrix ADC, make sure that the following prerequisites are met:

- A Entrust nShield Connect device is installed in the network, ready to use, and accessible to the Citrix ADC. That is, the NSIP address is added as an authorized client on the HSM.
- A usable Security World exists. Security World is a unique key management architecture used by the Entrust nShield line of HSMs. It protects and manages keys as application key tokens, enabling unlimited key capacity, and automatic key backup and recovery. For more information about creating a Security World, see the nShield Connect Quick Start Guide from Entrust. You can also find the guide in the CD provided with the Entrust HSM module at CipherTools-linux-

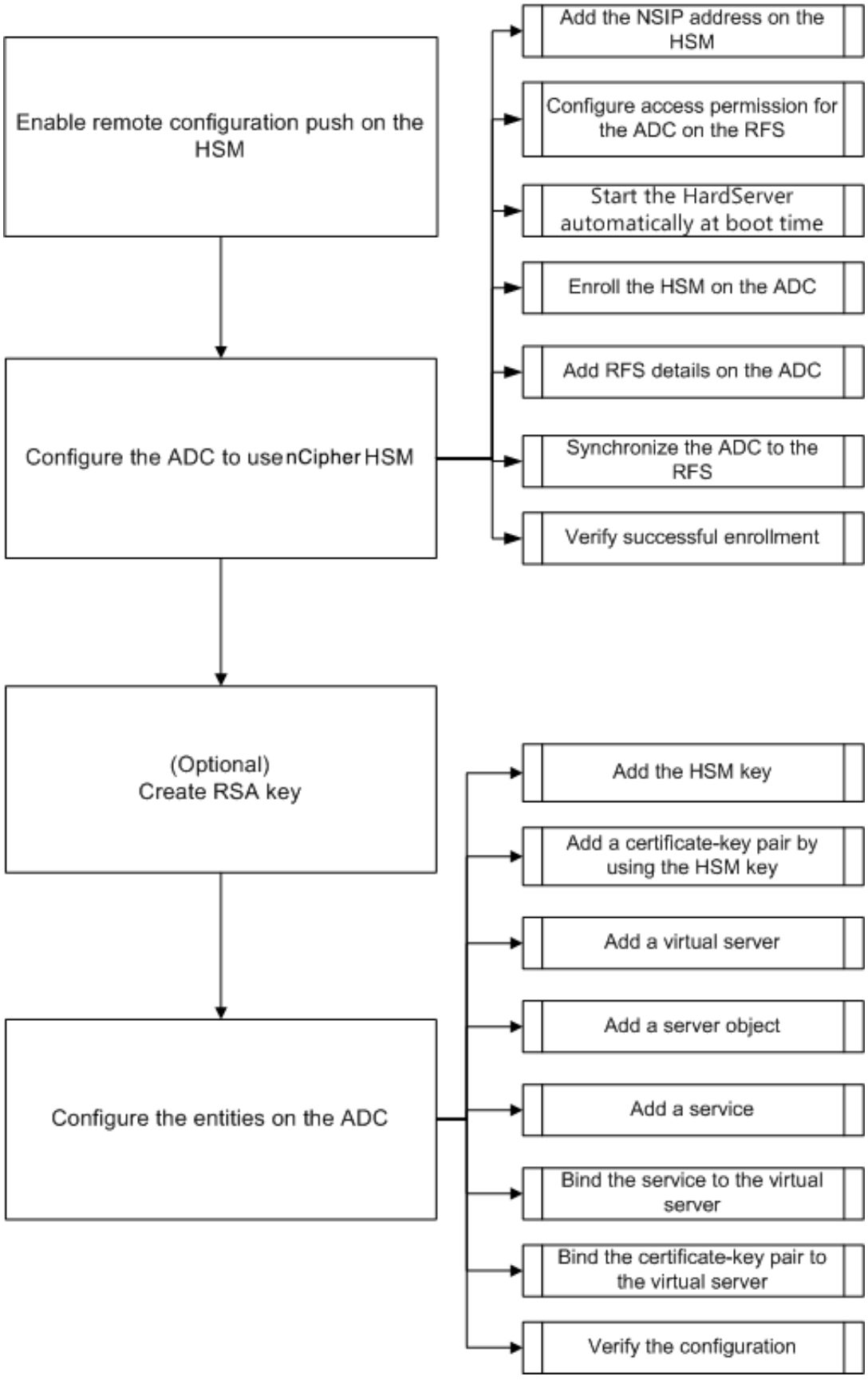
dev-xx.xx.xx/document/nShield_Connect_Quick_Start_Guide.pdf.
Note: `Softcard` or token/OCS protected keys are currently not supported on the Citrix ADC.

- Licenses are available to support the number of clients that are connected to the Entrust HSM. The ADC and remote file server (RFS) are clients of the HSM.
- An RFS is installed in the network and is accessible to the Citrix ADC.
- The Entrust nShield Connect device, the RFS, and the Citrix ADC can initiate connections with each other through port 9004.
- You are using NetScaler release 10.5 build 52.1115.e or later.
- The Citrix ADC appliance does not contain a FIPS Cavium card.
  Important: Entrust HSM is not supported on the MPX 9700/10500/12500/15500 FIPS appliances.

Configure the ADC-Entrust integration

September 14, 2021

The following flowchart depicts the tasks that you need to perform to use Entrust HSM with a Citrix ADC:



As shown in the preceding flowchart, you perform the following tasks:

1. Enable remote configuration push, on the HSM.
2. Configure the ADC to use the Entrust HSM.
    - Add the NSIP address on the HSM.
    - Configure access permission for the ADC on the RFS.
    - Configure automatic start of the `Hardserver` at boot time.
    - Enroll the HSM on the ADC.
    - Add RFS details on the ADC.
    - Synchronize the ADC to the RFS.
    - Verify that Entrust HSM is successfully enrolled on the ADC.
3. (Optional) Create an HSM RSA key.
4. Configure the entities on the Citrix ADC.
    - Add the HSM key.
    - Add a certificate-key pair by using the HSM key.
    - Add a virtual server.
    - Add a server object.
    - Add a service.
    - Bind the service to the virtual server.
    - Bind the certificate-key pair to the virtual server.
    - Verify the configuration.

Configure the Entrust HSM

Specify the IP address of the RFS on the Entrust HSM so that it accepts the configuration that the RFS pushes to it. Use the nShield Connect front panel on the Entrust HSM to perform the following procedure.

Specify the IP address of a remote computer on the Entrust HSM

1. Navigate to **System Configuration > Config file options > Allow auto push**.
2. Select **ON**, and specify the IP address of the computer (RFS) from which to accept the configuration.

Enable pushing the remote configuration on the HSM

Specify the IP address of the RFS on the Entrust HSM so that it accepts the configuration that the RFS pushes to it. Use the nShield Connect front panel on the Entrust HSM to perform the following procedure.

Specify the IP address of a remote computer on the Entrust HSM

1. Navigate to **System Configuration > Config file options > Allow auto push**.
2. Select **ON**, and specify the IP address of the computer (RFS) from which to accept the configuration.

Configure the ADC to use the Entrust HSM

Sample values used in this documentation:

NSIP address=10.217.2.43

Entrust HSM IP address=10.217.2.112

RFS IP address=10.217.2.6

Add the NSIP address on the HSM

Typically you use the nShield Connect front panel to add clients to the HSM. For more information, see the nShield Connect Quick Start Guide.

Alternately, use the RFS to add the ADC as a client to the HSM. To add the ADC, you must add the NSIP address in the HSM configuration on the RFS, and then push the configuration to the HSM. Before you can push the configuration, you must know the electronic serial number (ESN) of the HSM.

To get the ESN of your HSM, run the following command on the RFS:

```
root@ns# /opt/nfast/bin/anonkneti <Entrust HSM IP address>
<!--NeedCopy-->
```

Example:

```
root@ns# /opt/nfast/bin/anonkneti 10.217.2.112
BD17-C807-58D9 5e30a698f7bab3b2068ca90a9488dc4e6c78d822
<!--NeedCopy-->
```

The ESN number is BD17-C807-58D9.

After you have the ESN number, use an editor, such as vi, to edit the HSM configuration file on the RFS.

```
vi /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

In the `hs_clients` section, add the following entries:

```
# Amount of data in bytes to encrypt with a session key before session
    key# renegotiation, or 0 for unlimitied. (default=1024*1024*8b=8Mb).
```



```
#  datalimit=INT
addr=10.217.2.43
clientperm=unpriv
keyhash=0000000000000000000000000000000000000000
esn=
timelimit=86400
datalimit=8388608
-----
<!--NeedCopy-->
```

Note: Include one or more hyphens as delimiters to add multiple entries in the same section.

To push the configuration to the HSM, run the following command on the RFS:

```
/opt/nfast/bin/cfg-pushnethsm --address=<Entrust HSM IP address> --
   force /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

Example:

```
/opt/nfast/bin/cfg-pushnethsm --address=10.217.2.112 --force
                /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

Configure access permission for the ADC on the RFS

To configure access permission for the ADC on the RFS, run the following command on the RFS:

```
/opt/nfast/bin/rfs-setup --force -g --write-noauth <NetScaler IP
   address>
<!--NeedCopy-->
```

Example:

```
[root@localhost bin]# /opt/nfast/bin/rfs-setup --force -g --write-
   noauth 10.217.2.43
Adding read-only remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local exists
Adding new writable remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local/sync-store exists
Saving the new config file and configuring the hardserver
Done
<!--NeedCopy-->
```

Verify that the ADC can reach both the RFS and Entrust HSM by using port 9004.

Configure automatic start of the hardserver at boot time

Create a file and then restart the appliance. Now, whenever you restart the appliance, and if this file is found, the Hardserver is automatically started.

At the shell prompt, type:

```
touch /var/opt/nfast/bin/thales_hsm_is_enrolled
<!--NeedCopy-->
```

At the command prompt, type:

```
reboot
<!--NeedCopy-->
```

Enroll the HSM on the ADC

Change directory to /var/opt/nfast/bin.

To add HSM details into the ADC configuration, run the following command on the ADC:

```
nethsmenroll --force <Thales_nShield_Connect_ip_address> $(anonkneti <Thales_nShield_Connect_ip_address>)
```

Example:

```
root@ns# ./nethsmenroll --force 10.217.2.112 $(anonkneti 10.217.2.112)
OK configuring hardserver's nethsm imports
<!--NeedCopy-->
```

This step adds the following entries after the line # ntoken_esn=ESN in the nethsm_imports section of the /var/opt/nfast/kmdata/config/config file.

```
...
local_module=0
remote_ip=10.217.2.112
remote_port=9004
remote_esn=BD17-C807-58D9
keyhash=5e30a698f7bab3b2068ca90a9488dc4e6c78d822
timelimit=86400
datalimit=8388608
privileged=0
privileged_use_high_port=0
ntoken_esn=
<!--NeedCopy-->
```

Change the directory to /var/opt/nfast/bin and run the following command on the ADC:


```
touch "thales_hsm_is_enrolled"
<!--NeedCopy-->
```

Note: To remove an HSM that is enrolled on the ADC, type:

```
./nethsmenroll - -remove <NETHSM-IP>
<!--NeedCopy-->
```

Add RFS details on the ADC

To add RFS details, change the directory to /var/opt/nfast/bin/ and then run the following command:

```
./rfs-sync --no-authenticate --setup <rfs_ip_address>
<!--NeedCopy-->
```

Example:

```
./rfs-sync --no-authenticate --setup 10.217.2.6
No current RFS synchronization configuration.
Configuration successfully written; new config details:
Using RFS at 10.217.2.6:9004: not authenticating.
<!--NeedCopy-->
```

This step adds the following entries after the # local_esn=ESN line in the `rfs_sync_client` section of the /var/opt/nfast/kmdata/config/config file.

```
... ...
remote_ip=10.217.2.6
remote_port=9004
use_kneti=no
local_esn=
<!--NeedCopy-->
```

Note: To remove an RFS that is enrolled on the ADC, type:

```
./rfs_sync - remove
<!--NeedCopy-->
```

Synchronize the ADC to the RFS

To synchronize all the files, change the directory to /var/opt/nfast/bin and then run the following command on the ADC:


```
./rfs-sync - -update
<!--NeedCopy-->
```

This command fetches all the World files, module files, and key files from the /opt/nfast/kmdata/local directory on the RFS and puts them into the /var/opt/nfast/kmdata/local directory on the ADC. Citrix recommends that you manually copy the World files, the module_XXXX_XXXX_XXXX files, where XXXX_XXXX_XXXX is the ESN of the enrolled HSM, and only the required RSA key and certificate files.

Verify the Entrust HSM is successfully enrolled on the ADC

After you synchronize the ADC to the RFS, do the following:

- Verify that the local `Hardserver` is UP and running. (Entrust server running).
- Get the state of the configured HSMs, and verify that the values for the n_modules (number of modules) field and the km info fields are non-zero.
- Verify that the HSM is enrolled correctly and is usable (state 0x2 Usable) by the ADC.
- Load tests using `sigtest` run properly.

Change the directory to /var/opt/nfast/bin, and at the shell prompt, run the following commands:

```
root@ns# ./chkserv root@ns# ./nfkminfo root@ns# ./sigtest
<!--NeedCopy-->
```

See Appendix for an example.

Create an HSM RSA Key

Only RSA keys are supported as HSM keys.

Note: Skip this step if keys are already present in the `/opt/nfast/kmdata/local` folder on the RFS.

Create an RSA key, a self-signed certificate, and a Certificate Signing Request (CSR). Send the CSR to a certificate authority to get a server certificate.

The following files are created in the following example:

- Embed RSA key: key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
- Self-Signed Certificate: example_selfcert
- Certificate Signing Request: example_req

Note: The `generatekey` command is supported in strict FIPS 140-2 Level 3 Security World. An administrator card set (ACS) or an operator card set (OCS) is needed to control many operations, including the creation of keys and OCSs. When you run the `generatekey` command, you are prompted to insert an ACS or OCS card. For more information about strict FIPS 140-2 Level 3 Security World, see the nShield Connect User Guide.

The following example uses Level-2 Security World. In the example, the commands are in boldface type.

Example:

```
[root@localhost bin]# ./generatekey embed
size: Key size? (bits, minimum 1024) [1024] > 2048
OPTIONAL: pubexp: Public exponent for RSA key (hex)? []
>
embedsavefile: Filename to write key to? []
> example
plainname: Key name? [] > example
x509country: Country code? [] > US
x509province: State or province? [] > CA
x509locality: City or locality? [] > Santa Clara
x509org: Organisation? [] > Citrix
x509orgunit: Organisation unit? [] > NS
x509dnscommon: Domain name? [] > www.citrix.com
x509email: Email address? [] > example@citrix.com
nvram: Blob in NVRAM (needs ACS)? (yes/no) [no] >
digest: Digest to sign cert req with? (md5, sha1, sha256, sha384,
    sha512)
  [default sha1] > sha512
key generation parameters:
 operation      Operation to perform              generate
 application    Application                       embed
 verify         Verify security of key            yes
 type           Key type                          RSA
 size           Key size                          2048
 pubexp         Public exponent for RSA key (hex)
 embedsavefile  Filename to write key to          example
 plainname      Key name                          example
 x509country    Country code                      US
 x509province   State or province                 CA
 x509locality   City or locality                  Santa Clara
 x509org        Organisation                      Citrix
 x509orgunit    Organisation unit                 NS
 x509dnscommon  Domain name                       www.citrix.com
 x509email      Email address                     example@citrix.com
 nvram          Blob in NVRAM (needs ACS)         no
 digest         Digest to sign cert req with      sha512
Key successfully generated.
Path to key: /opt/nfast/kmdata/local/
    key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
You have new mail in /var/spool/mail/root
```



```
<!--NeedCopy-->
```

Result:

You have created a CSR (example_req), a self-signed certificate (example_selfcert), and an application key token file in embed format (/opt/nfast/kmdata/local/key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2

Because the ADC supports keys in simple format only, you must convert the embed key to a simple key.

To convert the embed key to a simple key, run the following command on the RFS:

```
[root@localhost bin]# ./generatekey -r simple
from-application: Source application? (embed, simple) [embed] > embed
from-ident: Source key identifier? (
   c6410ca00af7e394157518cb53b2db46ff18ce29,
                                    2
                                       ed5428aaeae1e159bdbd63f25292c7113ec2c78
                                       )
  [default c6410ca00af7e394157518cb53b2db46ff18ce29]
> 2ed5428aaeae1e159bdbd63f25292c7113ec2c78
ident: Key identifier? [] > examplersa2048key
plainname: Key name? [] > examplersa2048key
key generation parameters:
 operation         Operation to perform    retarget
 application       Application             simple
 verify            Verify security of key  yes
 from-application  Source application      embed
 from-ident        Source key identifier   2
    ed5428aaeae1e159bdbd63f25292c7113ec2c78
 ident             Key identifier          examplersa2048key
 plainname         Key name                examplersa2048key
Key successfully retargetted.
Path to key: /opt/nfast/kmdata/local/key_simple_examplersa2048key
<!--NeedCopy-->
```

> **Important:**
>
> When prompted for the source key identifier, enter **2ed5428aaeae1e159bdbd63f25292c7113ec2c78** as the embed key.

Result:

A key with the prefix key_simple (for example key_simple_examplersa2048key) is created.

Note: examplersa2048key is the key identifier (ident) and is referred to as the HSM key name on the ADC. A key identifier is unique. All the simple files have the prefix key_simple.

Configure the entities on the ADC

Before the ADC can process traffic, you must do the following:

1. Enable features.
2. Add a subnet IP (SNIP) address.
3. Add the HSM key to the ADC.
4. Add a certificate-key pair by using the HSM key.
5. Add a virtual server.
6. Add a server object.
7. Add a service.
8. Bind the service to the virtual server.
9. Bind the certificate-key pair to the virtual server.
10. Verify the configuration.

Enable features on the ADC

Licenses must be present on the ADC before you can enable a feature.

Enable a feature by using the CLI

At the command prompt, run the following commands:

```
enable feature lb
enable feature ssl
<!--NeedCopy-->
```

Enable a feature by using the GUI

Navigate to **System** > **Settings** and, in the **Modes and Features** group, select **Configure basic features**, and then select **SSL Offloading**.

Add a subnet IP address

For more information about subnet IP addresses, see Configuring Subnet IP Addresses.

Add a SNIP address and verify the configuration by using the CLI

At the command prompt, run the following commands:

```
add ns ip <IPAddress> <netmask> -type SNIP
show ns ip
<!--NeedCopy-->
```


Example:

```
add ns ip 192.168.17.253 255.255.248.0 -type SNIP
Done
show ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp
                 Icmp     Vserver  State
        ---------        --------------  ----             ----     ---
                 ----     -------  ------
1)      192.168.17.251   0               NetScaler IP     Active
   Enabled  Enabled  NA       Enabled
2)      192.168.17.252   0               VIP              Active
   Enabled  Enabled  Enabled  Enabled
3)      192.168.17.253   0               SNIP              Active
   Enabled  Enabled  NA       Enabled
 Done
<!--NeedCopy-->
```

Add a SNIP address and verify the configuration by using the GUI

Navigate to **System > Network > IPs**, add an IP address, and select the **IP Type as Subnet IP**.

Copy the HSM key and certificate to the ADC

Use a secure file transfer utility to securely copy the key (key_simple_examplersa2048key) to the `/var/opt/nfast/kmdata/local` folder, and the certificate (example_selfcert) to the `/nsconfig/ssl` folder on the ADC.

Add the key on the ADC

All the keys have a key-simple prefix. When adding the key to the ADC, use the ident as the HSM key name. For example, if the key that you added is key_simple_XXXX, the HSM key name is XXXX.

Important:

- The HSM key name must be the same as the ident that you specified when you converted an embed key to a simple key format.
- The keys must be present in the `/var/opt/nfast/kmdata/local/` directory on the ADC.

Add an HSM key by using the CLI

At the shell prompt, run the following command:


```
add ssl hsmKey <hsmKeyName> -key <string>
<!--NeedCopy-->
```

Example:

```
add ssl hsmKey examplersa2048key  - key key_simple_examplersa2048key
Done
<!--NeedCopy-->
```

Add an HSM key by using the GUI

Navigate to **Traffic Management > SSL > HSM**, and add an HSM key.

Add a certificate-key pair on the ADC

For information about certificate-key pairs, see Add or update a certificate-key pair.

Add a certificate-key pair by using the CLI

At the command prompt, run the following command:

```
add ssl certKey <certkeyName> -cert <string> -hsmKey <string>
<!--NeedCopy-->
```

Example:

```
add ssl certKey key22 -cert example_selfcert -hsmKey examplersa2048key
Done
<!--NeedCopy-->
```

Add a certificate-key pair by using the GUI

Navigate to **Traffic Management > SSL > Certificates**, and add a certificate-key pair.

Add a virtual server

For information about a virtual server, see SSL virtual server configuration.

Configure an SSL-based virtual server by using the CLI

At the command prompt, run the following command:

```
add lb vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver v1 SSL 192.168.17.252 443
<!--NeedCopy-->
```

Configure an SSL-based virtual server by using the GUI

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, create a virtual server, and specify the protocol as SSL.

Add a server object

Before you can add a server object on the ADC, make sure that you have created a back-end server. The following example uses the built-in python HTTP Server module on a Linux system.

Example:

```
%python  -m SimpleHTTPServer 80
<!--NeedCopy-->
```

Add a server object by using the CLI

At the command prompt, run the following command:

```
add server <name> <IPAddress>
<!--NeedCopy-->
```

Example:

```
add server s1 192.168.17.246
<!--NeedCopy-->
```

Add a server object by using the GUI

Navigate to **Traffic Management > Load Balancing > Servers**, and add a server.

Add a service

For more information, see Configuring services.

Configure a service by using the CLI

At the command prompt, run the following command:

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Example:

```
add service sr1 s1 HTTP 80
<!--NeedCopy-->
```

Configure a service by using the GUI

Navigate to **Traffic Management > Load Balancing > Services**, and create a service.

Bind the service to the virtual server

For more information, see Bind services to the SSL virtual server.

Bind a service to a virtual server by using the CLI

At the command prompt, run the following command:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver v1 sr1
<!--NeedCopy-->
```

Bind a service to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and click in the Services pane to bind a service to the virtual server.

Bind the certificate-key pair to the virtual server on the ADC

For more information, see Bind the certificate-key pair to the SSL virtual server.

Bind a certificate-key pair to a virtual server by using the CLI

At the command prompt, run the following command:

```
bind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```

Example:

```
bind ssl vserver v1 -certkeyName key22
Warning: Current certificate replaces the previous binding
<!--NeedCopy-->
```

Bind a certificate-key pair to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open an SSL virtual server and, in **Advanced Settings**, click **SSL Certificate**.
3. Bind a server certificate to the virtual server.

Verify the configuration

To view the configuration by using the CLI:

At the command prompt, run the following commands:

```
show lb vserver <name>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
show lb vserver v1
        v1 (192.168.17.252:443) - SSL   Type: ADDRESS
        State: UP
        Last state change was at Wed Oct 29 03:11:11 2014
        Time since last state change: 0 days, 00:01:25.220
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
```



```
        Appflow logging: ENABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state
           changed to UP
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
        L2Conn: OFF
        Skip Persistency: None
        IcmpResponse: PASSIVE
        RHIstate: PASSIVE
        New Service Startup Request Rate: 0 PER_SECOND, Increment
           Interval: 0
        Mac mode Retain Vlan: DISABLED
        DBS_LB: DISABLED
        Process Local: DISABLED
        Traffic Domain: 0

1) sr1 (192.168.17.246: 80) - HTTP State: UP    Weight: 1
Done
<!--NeedCopy-->
```

```
sh ssl vserver v1
        Advanced SSL configuration for VServer v1:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        SSLv2: DISABLED  SSLv3: DISABLED  TLSv1.0: ENABLED  TLSv1.1:
           DISABLED  TLSv1.2: DISABLED
        Push Encryption Trigger: Always
        Send Close-Notify: YES

        ECC Curve: P_256, P_384, P_224, P_521
```



```
18
19 1)      CertKey Name: key22       Server Certificate
20
21 1)      Cipher Name: DEFAULT
22         Description: Predefined Cipher Alias
23  Done
24 <!--NeedCopy-->
```

To view the configuration by using the GUI:

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and double-click an SSL virtual server to open it and view the configuration.

Limitations

September 14, 2021

- SSL version 3 (SSLv3) is not supported on an MPX appliance but is supported on a VPX virtual appliance. A VPX instance provisioned on an SDX appliance supports SSLv3 only if an SSL chip is not assigned to the instance.
- Export ciphers are not supported.
- SSL server key exchange using HSM keys is not supported.
- If you have added or removed keys after you last saved the configuration, you must save the configuration before you perform a warm restart. If you do not save the configuration, there is a key mismatch between the ADC and the HSM.
- You cannot bind an HSM key to a DTLS virtual server.
- You cannot bind a certificate-key pair that is created by using an HSM key to an SSL service.
- You cannot use the GUI to enroll the ADC as a client of the HSM or check the status of the HSM from the configuration utility.
- From release 11 build 62.x, SSL renegotiation is supported.
- You cannot sign OCSP requests by using a certificate-key pair that is created by using an HSM key.
- A certificate bundle with HSM keys is not supported.
- An error does not appear if the HSM key and certificate do not match. Therefore, while adding a certificate-key pair, you must make sure that the HSM key and certificate match.
- Clustering and admin partitions are not supported.

Appendix

September 14, 2021

Example:

Note: In the following example, the commands are in boldface type.

```
root@ns# ./chkserv
nCipher server running
root@ns# ./nfkminfo
World
 generation  2
 state       0x17a70000 Initialised Usable Recovery PINRecovery !
    ExistingClient RTC NVRAM FTO !AlwaysUseStrongPrimes SEEDebug
 n_modules   1
 hknso       cbec8c0c56c6b5e76b73147ef02d34a661eaa044
 hkm     bbb8d4839da5782be4d092735a7535538834dc91 (type Rijndael)
 hkmwk       1d572201be533ebc89f30fdd8f3fac6ca3395bf0
 hkre        01f21ecf43933ffdd45e74c3883525176c5c439c
 hkra        ac8ec5ee6bce00991bd97adce2091d9739b9b452
 hkmc        cf1b509abaad91995ed202d8f36613fc99433155
 hkp         c20910b2ed1ca62d6a2b0db67052a05f7bbfeb43
 hkrtc       bd811020a7c2f8df435a481c3767a89c2e13bc4f
 hknv        278b8012e48910d518a9ee91cff57233fb0c9093
 hkdsee      12230b0e31e3cec66324c0815f782cfb9249edd5
 hkfto       89dd6250b3d6149bcd15606f4553085e2fd6271a
 hkmnull     0100000000000000000000000000000000000000
 ex.client   none
 k-out-of-n  1/2
 other quora m=1 r=1 p=1 nv=1 rtc=1 dsee=1 fto=1
 createtime  2014-02-28 21:05:32
 nso timeout 10 min
 ciphersuite DLf1024s160mRijndael

Module #1
 generation 2
 state      0x2 Usable
 flags      0x10000 ShareTarget
 n_slots    2
 esn        BD17-C807-58D9
 hkml       70289a6edba00ddc7e3f6d6f5a49edc963e822f2

Module #1 Slot #0 IC 0
 generation    1
```



```
 phystype      SmartCard
 slotlistflags 0x2 SupportsAuthentication
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

Module #1 Slot #1 IC 0
 generation    1
 phystype      SoftToken
 slotlistflags 0x0
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

No Pre-Loaded Objects

root@ns# ./sigtest
 Hardware module #1 speed index 5792 recommended minimum queue 19
Found 1 module; using 19 jobs
Making 1024-bit RSAPrivate key on module #1;
  using Mech_RSApPKCS1 and PlainTextType_Bignum.
Generated and exported key from module #1.
Imported keys on module #1
 1,       3059 1223.6, 3059 overall
 2,       8698 2989.76, 4349 overall
 3,      14396 4073.06, 4798.67 overall
 4,      20091 4721.83, 5022.75 overall
 5,      25799 5116.3, 5159.8 overall
 6,      31496 5348.58, 5249.33 overall
 7,      37192 5487.55, 5313.14 overall
 8,      42780 5527.73, 5347.5 overall
 9,      45777 4515.44, 5086.33 overall
 10,      51457 4981.26, 5145.7 overall
 11,      57151 5266.36, 5195.55 overall
 12,      62813 5424.61, 5234.42 overall
 13,      68496 5527.97, 5268.92 overall
 14,      74182 5591.18, 5298.71 overall
 15,      79832 5614.71, 5322.13 overall
 16,      85518 5643.23, 5344.88 overall
```



```
 17,      88412 4543.54, 5200.71 overall
 18,      94086 4995.72, 5227 overall
 19,      99778 5274.23, 5251.47 overall
 20,     105469 5440.94, 5273.45 overall
 21,     111133 5530.16, 5292.05 overall
 22,     116838 5600.1, 5310.82 overall
 23,     122522 5633.66, 5327.04 overall
 24,     128175 5641.4, 5340.62 overall
 25,     131072 4543.64, 5242.88 overall
 26,     136762 5002.18, 5260.08 overall
 27,     142415 5262.51, 5274.63 overall
 28,     148125 5441.51, 5290.18 overall
 29,     153816 5541.3, 5304 overall
 30,     159414 5563.98, 5313.8 overall
<!--NeedCopy-->
```

Support for Thales Luna Network hardware security module

September 14, 2021

A non-FIPS Citrix ADC appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as a hardware security module (HSM). Storing a key in the HSM protects it from physical and software attacks. In addition, the keys are encrypted with special FIPS approved ciphers.

Only the Citrix ADC MPX 9700/10500/12500/15500 FIPS appliances and the MPX/SDX 14000 FIPS appliances support a FIPS card. Support for FIPS is not available on other MPX/SDX appliances, or on the Citrix ADC VPX appliances. This limitation is addressed by supporting a Thales Luna network HSM on all Citrix ADC MPX, SDX, and VPX appliances except the MPX 9700/10500/12500/15500 FIPS and the MPX/SDX 14000 FIPS appliances.

A Thales Luna network HSM is designed to protect critical cryptographic keys and to accelerate sensitive cryptographic operations across a wide range of security applications.

Supported versions matrix

Citrix ADC Version	Software Appliance Version	Firmware Version	Client Version
11.1, 12.0, 12.1, 13.0	5.2.3-1	6.2.1	6.0.0
11.1, 12.0, 12.1, 13.0	6.2.2-5	6.10.9	6.2.2

Citrix ADC Version	Software Appliance Version	Firmware Version	Client Version
13.0	7.2.0-220	7.0.3	7.2.2 (7.2.0-220)

Prerequisites

September 14, 2021

Before you can use a Thales Luna network HSM with a Citrix ADC, make sure that the following prerequisites are met:

- A Thales Luna network HSM is installed in the network, ready to use, and accessible to the Citrix ADC. That is, the NSIP address or the SNIP address is added as an authorized client on the HSM.
- Licenses are available to support the required number of partitions on the HSM.
- The Thales Luna network HSM and the Citrix ADC can initiate connections with each other through port 1792.
- You are using NetScaler release 11.1 or later.
- The Citrix ADC appliance does not contain a FIPS Cavium card.

> **Important**
>
> Thales Luna network HSMs are not supported on the MPX 9700/10500/12500/15500 FIPS appliances.

Configure a Thales Luna client on the ADC

September 14, 2021

After you have configured the Thales Luna HSM and created the required partitions, you must create clients and assign them to partitions. Begin by configuring the Thales Luna clients on the Citrix ADC and setting up the network trust links (NTLs) between the Thales Luna clients and the Thales Luna HSM. A sample configuration is given in the Appendix.

1. Change the directory to /var/safenet and install the Thales Luna client. At the shell prompt, type:

   ```
   cd /var/safenet
   <!--NeedCopy-->
   ```

   To install Thales Luna client version 6.0.0, type:


```
install_client.sh -v 600
<!--NeedCopy-->
```

To install Thales Luna client version 6.2.2, type:

```
install_client.sh -v 622
<!--NeedCopy-->
```

To install Thales Luna client version 7.2.2, type:

```
install_client.sh -v 722
<!--NeedCopy-->
```

2. Configure the NTLs between Thales Luna client (ADC) and HSM.

   After the '/var/safenet/' directory is created, perform the following tasks on the ADC.

   a) Change the directory to '/var/safenet/config/' and run the 'safenet_config' script. At the shell prompt, type:

```
cd /var/safenet/config

sh safenet_config
<!--NeedCopy-->
```

   This script copies the "Chrystoki.conf" file into the /etc/ directory. It also generates a symbolic link 'libCryptoki2_64.so' in the '/usr/lib/' directory.

   b) Create and transfer a certificate and key between the ADC and the Thales Luna HSM.

   To communicate securely, the ADC and the HSM must exchange certificates. Create a certificate and key on the ADC and then transfer it to the HSM. Copy the HSM certificate to the ADC.

   i) Change directory to /var/safenet/safenet/lunaclient/bin.

   ii) Create a certificate on the ADC. At the shell prompt, type:

```
./vtl createCert -n <ip address of Citrix ADC>
<!--NeedCopy-->
```

   This command also adds the certificate and key path to the "/etc/Chrystoki.conf" file.

   iii) Copy this certificate to the HSM. At the shell prompt, type:

```
scp /var/safenet/safenet/lunaclient/cert/client/<ip address of NS
   >.pem <LunaSA_HSM account>@<IP address of Luna SA>
<!--NeedCopy-->
```


iv) Copy the HSM certificate to the Citrix ADC. At the shell prompt, type:

```
scp <HSM account>@<HSM IP>:server.pem  /var/safenet/safenet/
   lunaclient/server_<HSM ip>.pem
<!--NeedCopy-->
```

3. Register the Citrix ADC as a client and assign it a partition on the Thales Luna HSM.

   Log on to the HSM and create a client. Enter the NSIP as the client IP. This address must be the IP address of the ADC from which you transferred the certificate to the HSM. After the client is successfully registered, assign a partition to it. Run the following commands on the HSM.

   a) Use SSH to connect to the Thales Luna HSM and enter the password.

   b) Register the Citrix ADC on the Thales Luna HSM. The client is created on the HSM. The IP address is the client's IP address. That is, the NSIP address.

   At the prompt, type:

   ```
   client register  -client <client name> -ip <Citrix ADC ip>
   <!--NeedCopy-->
   ```

   c) Assign the client a partition from the partition list. To view the available partitions, type:

   ```
   <luna_sh> partition list
   <!--NeedCopy-->
   ```

   Assign a partition from this list. Type:

   ```
   <lunash:> client assignPartition -client <Client Name> -par <
      Partition Name>
   <!--NeedCopy-->
   ```

4. Register the HSM with its certificate on the Citrix ADC.

   On the ADC, change the directory to "/var/safenet/safenet/lunaclient/bin" and, at the shell prompt, type:

   ```
   ./vtl addserver -n <IP addr of HSM> -c /var/safenet/safenet/
      lunaclient/server_<HSM_IP>.pem
   <!--NeedCopy-->
   ```

   To remove the HSM that is enrolled on the ADC, type:

   ```
   ./vtl deleteServer -n <HSM IP> -c <cert path>
   <!--NeedCopy-->
   ```

   To list the HSM servers configured on the ADC, type:


```
./vtl listServer
<!--NeedCopy-->
```

> **Note:**
>
> Before removing the HSM by using `vtl`, make sure all the keys for that HSM are manually removed from the appliance. HSM keys cannot be deleted after the HSM server is removed.

5. Verify the network trust links (NTLs) connectivity between the ADC and HSM. At the shell prompt, type:

```
./vtl verify
<!--NeedCopy-->
```

   If verification fails, review all the steps. Errors are due to an incorrect IP address in the client certificates.

6. Save the configuration.

   The preceding steps update the "/etc/Chrystoki.conf" configuration file. This file is deleted when the ADC is started. Copy the configuration to the default configuration file, which is used when an ADC is restarted.

   At the shell prompt, type:

```
root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

   Recommended practice is to run this command every time there is a change to the Thales Luna related configuration.

7. Start the Thales Luna gateway process.

   At the shell prompt, type:

```
sh /var/safenet/gateway/start_safenet_gw
<!--NeedCopy-->
```

8. Configure automatic start of the gateway daemon at boot time.

   Create the "safenet_is_enrolled" file, which indicates that Thales Luna HSM is configured on this ADC. Whenever the ADC restarts and this file is found, the gateway is automatically started.

   At the shell prompt, type:

```
touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```


Configure Thales Luna HSMs in a high availability setup on the ADC

September 14, 2021

Configuring Thales Luna HSMs in a high availability (HA) ensures uninterrupted service even if all, but one of the devices, are unavailable. In an HA setup, each HSM joins an HA group in active-active mode. Thales Luna HSMs in an HA setup provide load balancing of all the group members to increase performance and response time while providing the assurance of high availability service. For more information, contact Thales Luna Sales and Support.

Prerequisites:

- Minimum two Thales Luna HSM devices. All the devices in an HA group must have either PED (trusted path) authentication or password authentication. A combination of trusted path authentication and password authentication in an HA group is not supported.
- Partitions on each HSM device must have the same password even if the label (name) is different.
- All partitions in HA must be assigned to the client (Citrix ADC appliance).

After configuring a Thales Luna client on the ADC as described in Configure a Thales Luna client on the ADC, perform the following steps to configure Thales Luna HSMs in HA:

1. On the Citrix ADC shell prompt, launch `lunacm` (/usr/safenet/lunaclient/bin)

   Example:

   ```
   root@ns# cd /var/safenet/safenet/lunaclient/bin/

   root@ns# ./lunacm
   <!--NeedCopy-->
   ```

2. Identify the slot IDs of the partitions. To list the available slots (partitions), type:

   ```
   lunacm:> slot list
   <!--NeedCopy-->
   ```

   Example:

   ```
       Slot Id ->              0
       HSM Label ->            trinity-p1
       HSM Serial Number ->    481681014
       HSM Model ->            LunaSA 6.2.1
       HSM Firmware Version -> 6.10.9
       HSM Configuration ->    Luna SA Slot (PED) Signing With
           Cloning Mode
       HSM Status ->           OK
   ```



```
8
9       Slot Id ->              1
10      HSM Label ->            trinity-p2
11      HSM Serial Number ->    481681018
12      HSM Model ->            LunaSA 6.2.1
13      HSM Firmware Version -> 6.10.9
14      HSM Configuration ->    Luna SA Slot (PED) Signing With
           Cloning Mode
15      HSM Status ->           OK
16
17       Slot Id ->              2
18       HSM Label ->            neo-p1
19       HSM Serial Number ->    487298014
20       HSM Model ->            LunaSA 6.2.1
21       HSM Firmware Version -> 6.10.9
22       HSM Configuration ->    Luna SA Slot (PED) Signing With
            Cloning Mode
23      HSM Status ->           OK
24
25      Slot Id ->              3
26      HSM Label ->            neo-p2
27      HSM Serial Number ->    487298018
28      HSM Model ->            LunaSA 6.2.1
29      HSM Firmware Version -> 6.10.9
30      HSM Configuration ->    Luna SA Slot (PED) Signing With
           Cloning Mode
31      HSM Status ->           OK
32
33      Slot Id ->              7
34      HSM Label ->            hsmha
35      HSM Serial Number ->    1481681014
36      HSM Model ->            LunaVirtual
37      HSM Firmware Version -> 6.10.9
38      HSM Configuration ->    Luna Virtual HSM (PED) Signing With
           Cloning Mode
39      HSM Status ->           N/A - HA Group
40
41      Slot Id ->              8
42      HSM Label ->            newha
43      HSM Serial Number ->    1481681018
44      HSM Model ->            LunaVirtual
45      HSM Firmware Version -> 6.10.9
46      HSM Configuration ->    Luna Virtual HSM (PED) Signing With
           Cloning Mode
47      HSM Status ->           N/A - HA Group
```



```
48
49      Current Slot Id: 0
50  <!--NeedCopy-->
```

3. Create the HA group. The first partition is called the primary partition. You can add more than one secondary partitions.

```
1  lunacm:> hagroup createGroup -slot <slot number of primary
       partition> -label <group name> -password <partition password >
2
3  lunacm:> hagroup createGroup -slot 1 -label gp12 -password ******
4  <!--NeedCopy-->
```

4. Add the secondary members (HSM partitions). Repeat this step for all partitions to be added to the HA group.

```
1  lunacm:> hagroup addMember -slot <slot number of secondary
       partition to be added> -group <group name> -password <partition
        password>
2  <!--NeedCopy-->
```

Code:

```
1  lunacm:> hagroup addMember -slot 2 -group gp12 -password ******
2  <!--NeedCopy-->
```

5. Enable HA only mode.

```
1  lunacm:> hagroup HAOnly  - enable
2  <!--NeedCopy-->
```

6. Enable active recovery mode.

```
1  lunacm:.>hagroup recoveryMode  - mode active
2  <!--NeedCopy-->
```

7. Set auto recovery interval time (in seconds). Default is 60 seconds.

```
1  lunacm:.>hagroup interval  - interval <value in seconds>
2  <!--NeedCopy-->
```

Example:

```
1  lunacm:.>hagroup interval  - interval 120
2  <!--NeedCopy-->
```


8. Set recovery retry count. A value of -1 allows an infinite number of retries.

```
lunacm:> hagroup retry -count <xxx>
<!--NeedCopy-->
```

Example:

```
lunacm:> hagroup retry -count 2
<!--NeedCopy-->
```

9. Copy the configuration from `Chrystoki.conf` to the SafeNet configuration directory.

```
cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

10. Restart the ADC appliance.

```
reboot
<!--NeedCopy-->
```

After configuring Thales Luna HSM in HA, see Other ADC configuration for further configuration on the ADC.

Other ADC configuration

September 14, 2021

1. Generate a key on the HSM.

   Use third party tools to create keys on the HSM.

2. Add an HSM key on the ADC.

   Important! The # character is not supported in a key name. If the key name includes this character, the load key operation fails.

   To add a Thales Luna HSM key by using the CLI:

   At the command prompt, type:

```
add ssl hsmkey <KeyName> -hsmType SAFENET -serialNum <serial #> -
    password
<!--NeedCopy-->
```

   where:

-keyName is the key created on the HSM by using third party tools.

-serialNum is the serial number of the partition on the HSM on which the keys are generated.

Note: For HSM in a high availability setup, use the serial number of the high availability group.

-password is the password of the partition on which the keys are present.

To add a Thales Luna HSM key by using the GUI:

Navigate to **Traffic Management** > **SSL** > **HSM** and add an HSM key. You must specify the HSM Type as **SAFENET**.

3. Add a certificate-key pair on the ADC. First use a third party tool to generate a certificate associated with the key. Then, copy the certificate to the /nsconfig/ssl/ directory on the ADC.

   Note: The key must be an HSM key.

   To add a certkey pair on the ADC by using the CLI:

   At the command prompt, type:

   ```
   add ssl certkey <CertkeyName> -cert <cert name> -hsmkey <KeyName>
   <!--NeedCopy-->
   ```

   To add a certkey pair on the ADC by using the GUI:

   a) Navigate to **Traffic Management** > **SSL**.
   b) In **Getting Started**, select **Install Certificate (HSM)** and create a certificate-key pair using an HSM key.

4. Create a virtual server and bind the certificate-key pair to this virtual server.

For information about creating a virtual server, click SSL virtual server configuration.

For information about adding a certificate-key pair, click Add or update a certificate-key pair.

For information about binding a certificate-key pair to an SSL virtual server, click Bind the certificate-key pair to the SSL virtual server.

Citrix ADC appliances in a high availability setup

September 14, 2021

You can configure a high availability (HA) setup on the Citrix ADC appliances with a Thales Luna HSM configuration in either of the following two ways:

- First, configure a Thales Luna HSM on the two nodes, using the same HSM and partition. Then create an HA pair. Finally, add the Citrix ADC configuration, such as keys, certificate-key pairs, and virtual servers, on the primary node.

- If a Thales Luna HSM is already configured on one node with the Citrix ADC configuration, add a similar configuration on the other node. Copy "/var/safenet/sfgw_ident_file" from the first node to the other and restart the safenet_gw binary. After the gateway is up and running, add the nodes in an HA setup.

Limitations

September 24, 2021

1. For any changes to the HSM-related configuration in an existing setup, such as adding or removing an HSM, or creating a high availability setup, copy '/etc/Chrystoki.conf' to '/var/safenet/config'.
2. After adding, removing, or restarting an HSM, you must restart the '/var/safenet/gateway/safenet_gw' binary. If you don't restart the gateway binary, the HSM will not serve any traffic after it is added back or after it restarts.
3. To reboot or stop the current '/var/safenet/gateway/safenet_gw' binary, use

   ```
   kill -SIGTERM <PID>
   kill -SIGINT <PID>
   <!--NeedCopy-->
   ```

   Important! Do not use `kill -9 <PID>` or `kill -6 <PID>`
4. Before removing an existing HSM from the ADC, remove, from the ADC, all the keys and certificate-key pairs that are associated with that HSM. You can't delete these files from the ADC after you remove the HSM.
5. On a standalone Citrix ADC appliance, Thales Luna HSMs in HA are supported for Luna version 6.2 and later.
6. EXPORT ciphers are not supported.
7. Update certificate-key pair operation is not supported.
8. When you generate an HSM key on a third-party tool, the private and public key names must be the same. When you add the HSM key on the appliance, provide this name as the key name.
9. The ## character is not supported in a key name and partition password.
10. Cluster and admin partitions are not supported.

Appendix

September 14, 2021

Sample commands with their outputs:

Run the script

```
1      root@ns# pwd
2     /var/safenet/config
3     root@ns# sh safenet_config
4 <!--NeedCopy-->
```

Create a certificate

```
1     root@ns# cd /var/safenet/safenet/lunaclient/bin
2     root@ns# ./vtl createcert -n 10.102.59.175
3     Private Key created and written to: /var/safenet/safenet/lunaclient
         /cert/client/10.102.59.175Key.pem
4     Certificate created and written to: /var/safenet/safenet/lunaclient
         /cert/client/10.102.59.175.pem
5 <!--NeedCopy-->
```

Copy the certificate to the HSM

```
1     root@ns# scp /var/safenet/safenet/lunaclient/cert/client
         /10.102.59.175.pem admin@10.217.2.7:
2     admin@10.217.2.7's password:
3
4     10.102.59.175.pem          100%  818     0.8KB/s   00:00
5 <!--NeedCopy-->
```

Copy the certificate and key from the HSM to the Citrix ADC appliance

```
1     root@ns# scp admin@10.217.2.7:server.pem /var/Thales Luna/safenet/
         lunaclient/server.2.7.pem
2     admin@10.217.2.7's password:
3
4     server.pem            100% 1164     1.1KB/s   00:01
5 <!--NeedCopy-->
```


Use SSH to connect to the Thales Luna HSM

```
    ssh admin@10.217.2.7
    Connecting to 10.217.2.7:22...
    Connection established.
    To escape to local shell, press 'Ctrl+Alt+]'.

    Last login: Thu Jun 23 02:20:29 2016 from 10.252.243.11

    Luna SA 5.2.3-1 Command Line Shell - Copyright (c) 2001-2014
       SafeNet, Inc. All rights reserved.

    [Safenet1] lunash:>hsm login


      Please enter the HSM Administrators' password:
      > *******

    'hsm login' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

Register the Citrix ADC on the Thales Luna HSM

```
    [Safenet1] lunash:>client register -client ns175 -ip 10.102.59.175

    'client register' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

Assign the client a partition from the partition list

```
    [Safenet1] lunash:>client assignPartition -client ns175 -partition
       p2

    'client assignPartition' successful.
```



```
4
5
6      Command Result : 0 (Success)
7      [Safenet1] lunash:>
8  <!--NeedCopy-->
```

Register the HSM with its certificate on the Citrix ADC

```
1      root@ns# ./vtl addserver -n 10.217.2.7 -c /var/safenet/safenet/
           lunaclient/server.2.7.pem
2
3      New server 10.217.2.7 successfully added to server list.
4  <!--NeedCopy-->
```

Verify the network trust links (NTLs) connectivity between the ADC and HSM

```
1      root@ns# ./vtl verify
2
3      The following Luna SA Slots/Partitions were found:
4
5      Slot        Serial #                  Label
6      ====      ================          =====
7         0              477877010          p2
8  <!--NeedCopy-->
```

Save the configuration

```
1      root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
2  <!--NeedCopy-->
```

Configure automatic start of the gateway daemon at boot time

```
1      touch /var/safenet/safenet_is_enrolled
2  <!--NeedCopy-->
```


FAQ

September 14, 2021

- **How do I check that the Thales Luna process is running?**

  At the Citrix ADC shell prompt, type:

```
ps - aux | grep safenet_gw
<!--NeedCopy-->
```

- **How do I verify the network trust links (NTLs) connectivity between the ADC and HSM?**

  After configuring Thales Luna, change the directory to "/var/safenet/safenet/lunaclient/bin" and type:

```
./vtl verify
<!--NeedCopy-->
```

Support for Azure Key Vault

November 12, 2021

The Citrix ADC appliance integrates with external HSMs (SafeNet and Thales) for on-premises deployments. For cloud deployments, the ADC appliance integrates with Azure Key Vault. The appliance stores its private keys in the Key Vault for ease of management and security of the private key in the public cloud domain. You no longer have to store and manage keys in different locations for ADC appliances deployed across multiple data centers and cloud providers.

Using ADC with the Azure Key Vault Premium pricing tier, which provided HSM backed keys, provides FIPS 140-2 level 2 compliance.

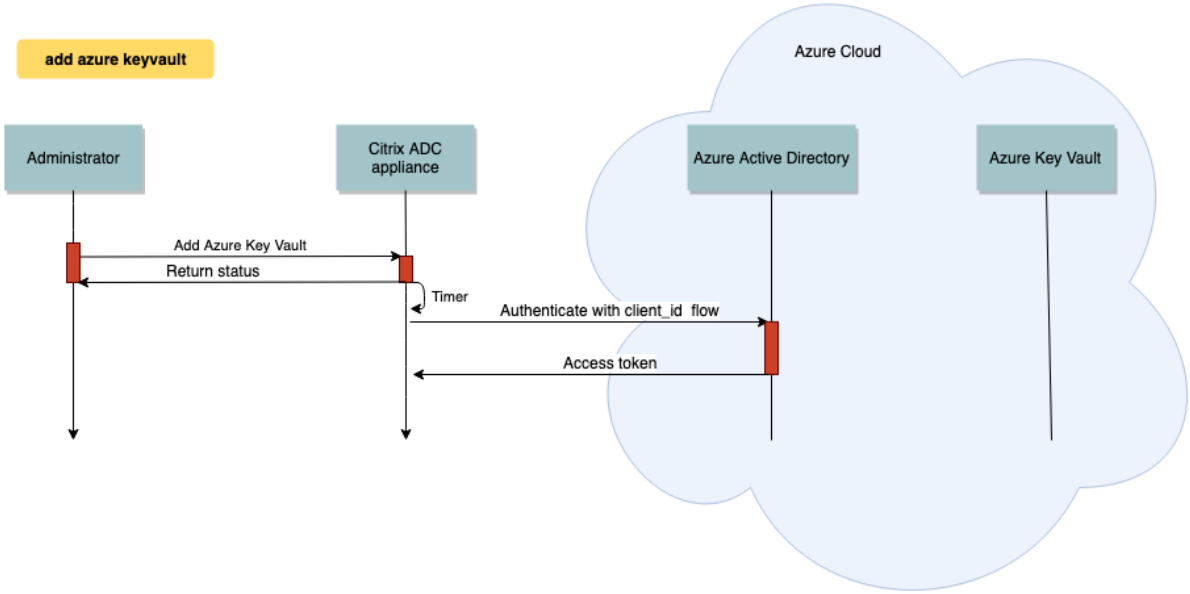
Azure Key Vault is a standard offering from Microsoft. For more information about Azure Key Vault, see the Microsoft Azure documentation.

Note: The Citrix ADC integration with Azure Key Vault is supported with the TLS 1.3 protocol.

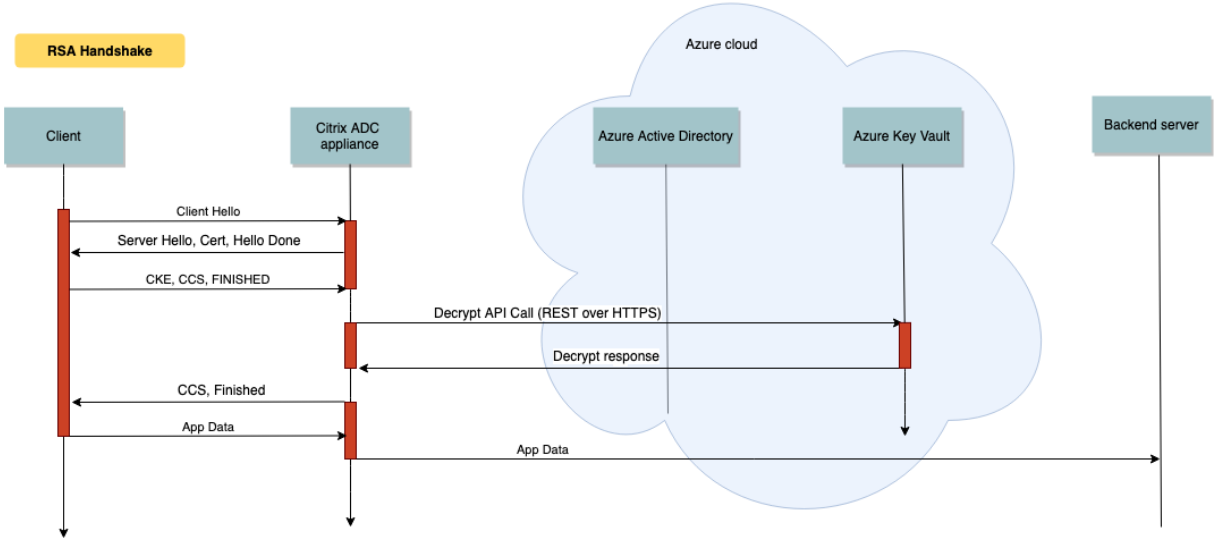
Architecture overview

Azure Key Vault is a service for storing secrets securely in the Azure cloud. By storing your keys in the Azure Key Vault, you reduce the chances of keys being stolen. Once the Key Vault is set up, you can store your keys in it. Configure virtual servers on the ADC appliance to perform private key operations in the Key Vault. The ADC appliance accesses the key for each SSL handshake.

The following diagram illustrates the process to get an access token from the Azure Active Directory after authentication. This token is used with REST API calls for crypto operations using private keys.



The following diagram shows a typical RSA handshake. The client key exchange (CKE) message that is encrypted using the public key is decrypted using the private key stored in the Key Vault.



In an ECDHE handshake, the server key exchange (SKE) message sent by the Citrix ADC appliance is signed by using the private key stored in the Key Vault.

Prerequisites

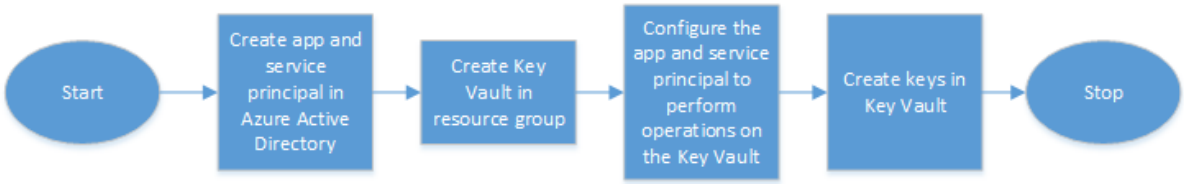
1. You must have an Azure subscription.
2. (Optional) Install Azure CLI on a Linux machine. For instructions, see the Azure documentation https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt?view=azure-cli-latest.
3. Complete the configuration on the Azure portal before configuring entities on the ADC appliance.

Configure the ADC Azure Key Vault integration

First perform the configuration on the Azure portal followed by the configuration on the ADC appliance.

Perform the following steps on the Azure portal

The following flowchart shows the high-level flow for configuration required on the Azure portal.

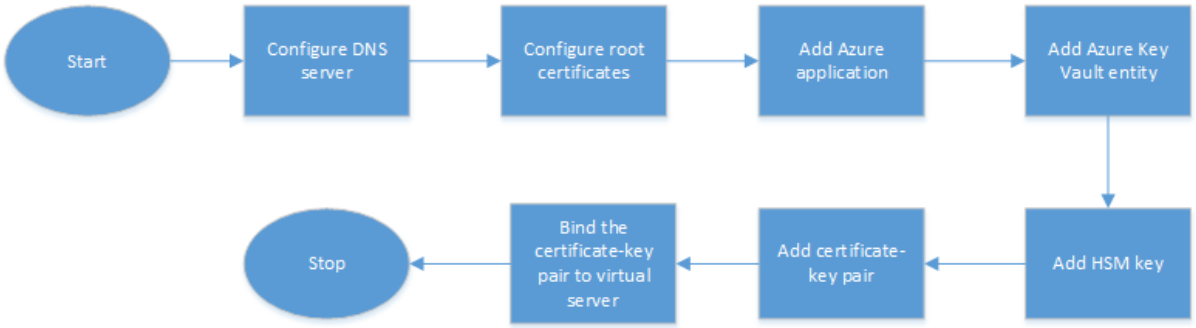


1. Create app and service principal in Azure Active Directory.
2. Create Key Vault in a resource group.
3. Configure the app and service principal to perform sign and decrypt operations on the Key Vault.
4. Create keys in the Key Vault using one of the following ways:
    a) By importing a key file.
    b) By generating a certificate.

For information about the commands to configure the preceding steps, see the Azure documentation at https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals.

Perform the following steps on the ADC appliance

The following flowchart shows the high-level flow for configuration required on the ADC appliance.



1. Configure a DNS server.
2. Configure root certificates to verify the certificates presented by Azure.
3. Create an Azure application.
4. Create an Azure Key Vault entity.
5. Create an HSM key.
6. Create a certificate-key pair.
7. Bind the certificate-key pair to a virtual server.

Configure a DNS server

A DNS server is required for the name resolution of Key Vault host and Azure Active Directory end point.

To configure a DNS server by using the CLI

At the command prompt, type:

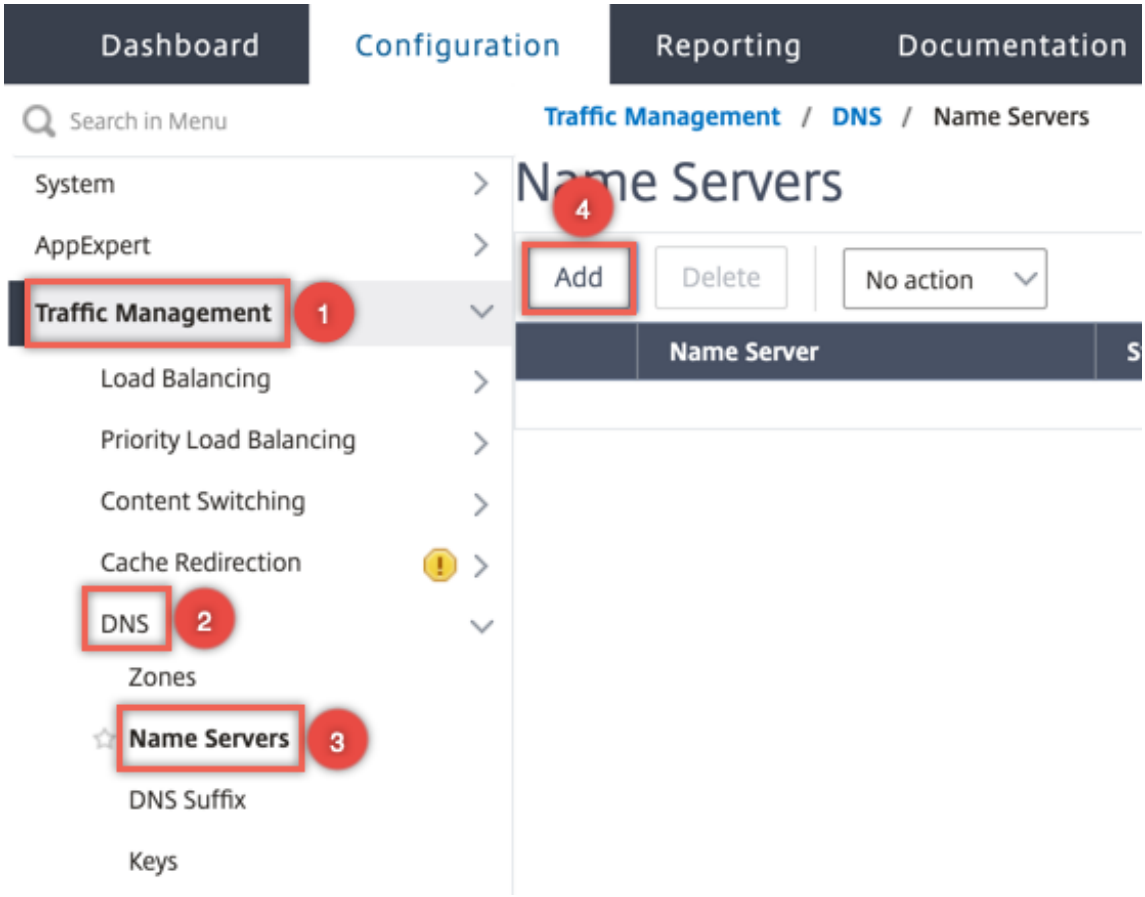
```
add dns nameserver <IP address>
<!--NeedCopy-->
```

Example:

```
add dns nameserver 192.0.2.150
<!--NeedCopy-->
```

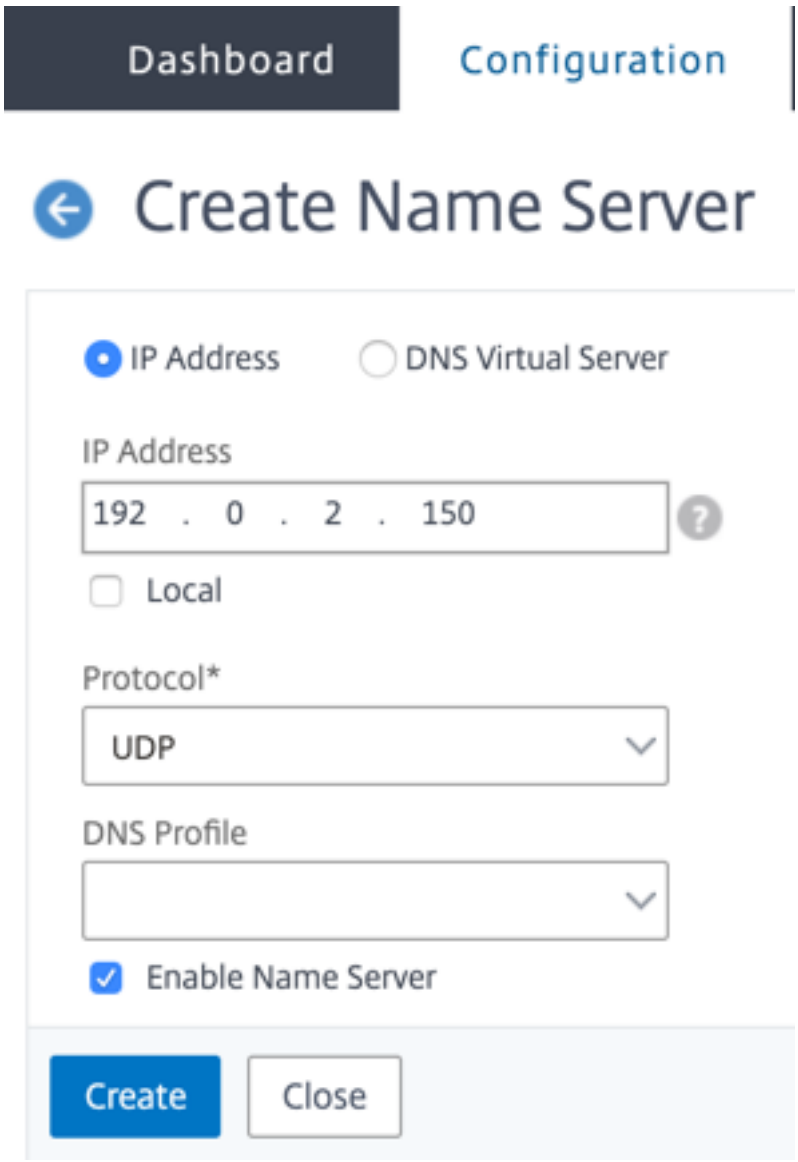
To configure a DNS server by using the GUI

1. Navigate to **Traffic Management > DNS > Name Servers**. Click **Add**.



2. Enter values for the following parameters:

   - IP Address - IP address of an external name server or, if the Local parameter is set, IP address of a local DNS server (LDNS).
   - Protocol - Protocol used by the name server. UDP_TCP is not valid if the name server is a DNS virtual server configured on the appliance.



3. Click **Create**.

Add and bind a root certificate

Download the root certificates of the certificate presented by Azure Key Vault `https://<vault_name>.vault.azure.net` and Azure Active Directory (AAD) https://login.microsoftonline.com and load it on the ADC appliance. These certificates are required to validate the certificate presented by Azure Key Vault and AAD. Bind one or more certificates to the CA certificate group `ns_callout_certs`.

To add a root certificate by using the CLI

At the command prompt, type:


```
add ssl certkey <certkeyname> -cert <certname>
bind ssl caCertGroup <caCertGroupName> <certkeyName>
<!--NeedCopy-->
```

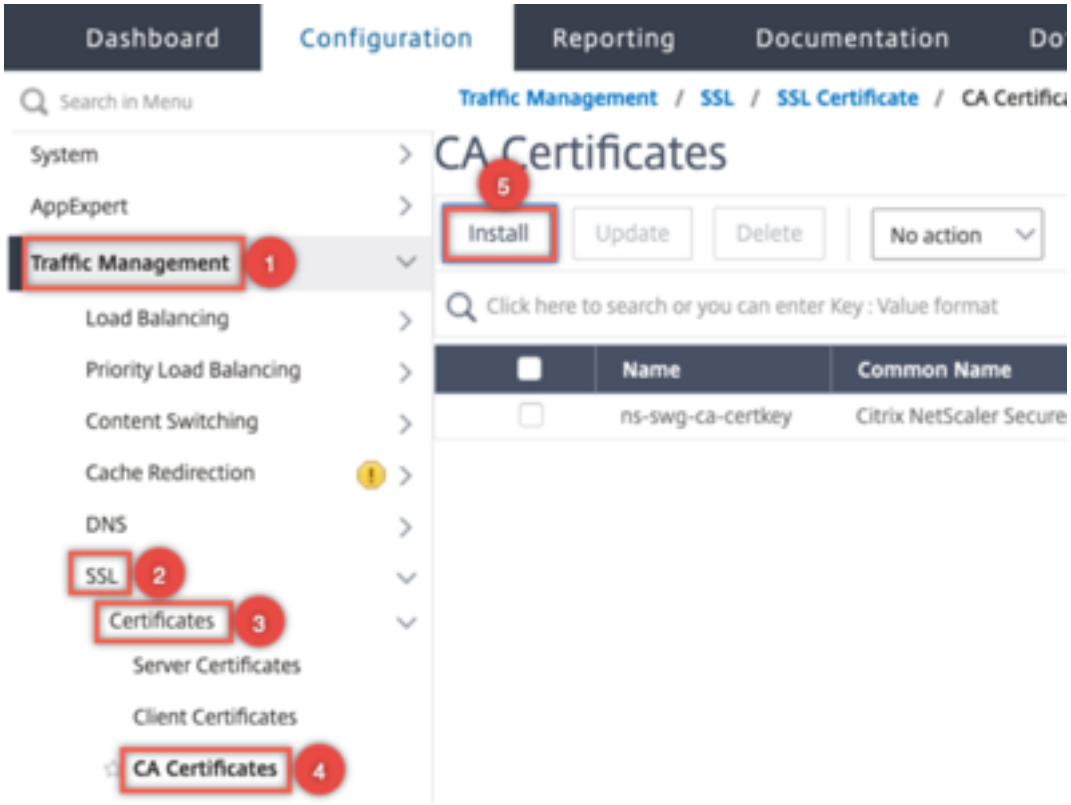
Example:

In the following example, the root certificate presented by Azure Key Vault and AAD is the same.

```
add ssl certKey rootcert -cert RootCyberTrustRoot.crt
bind ssl cacertGroup ns_callout_certs  rootcert
<!--NeedCopy-->
```

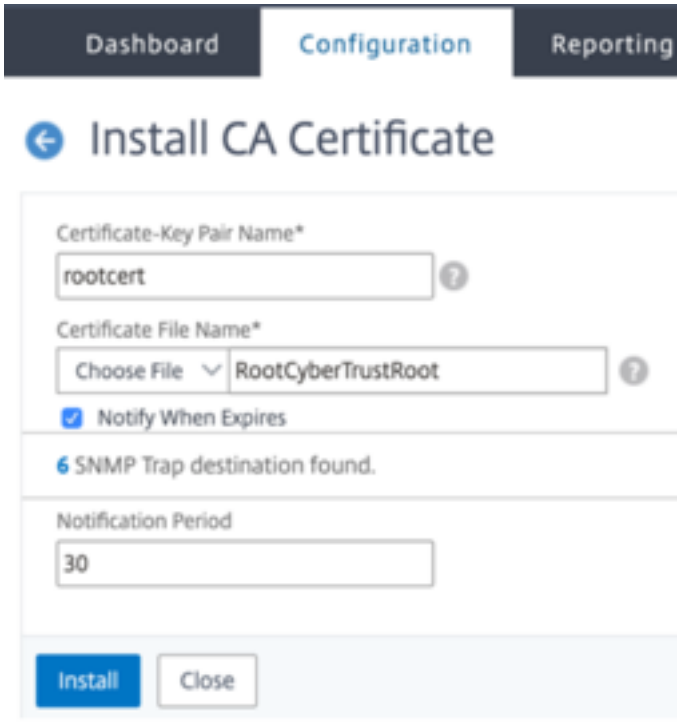
To add a root certificate by using the GUI

1. Navigate to **Traffic Management > SSL > Certificates > CA Certificates**.

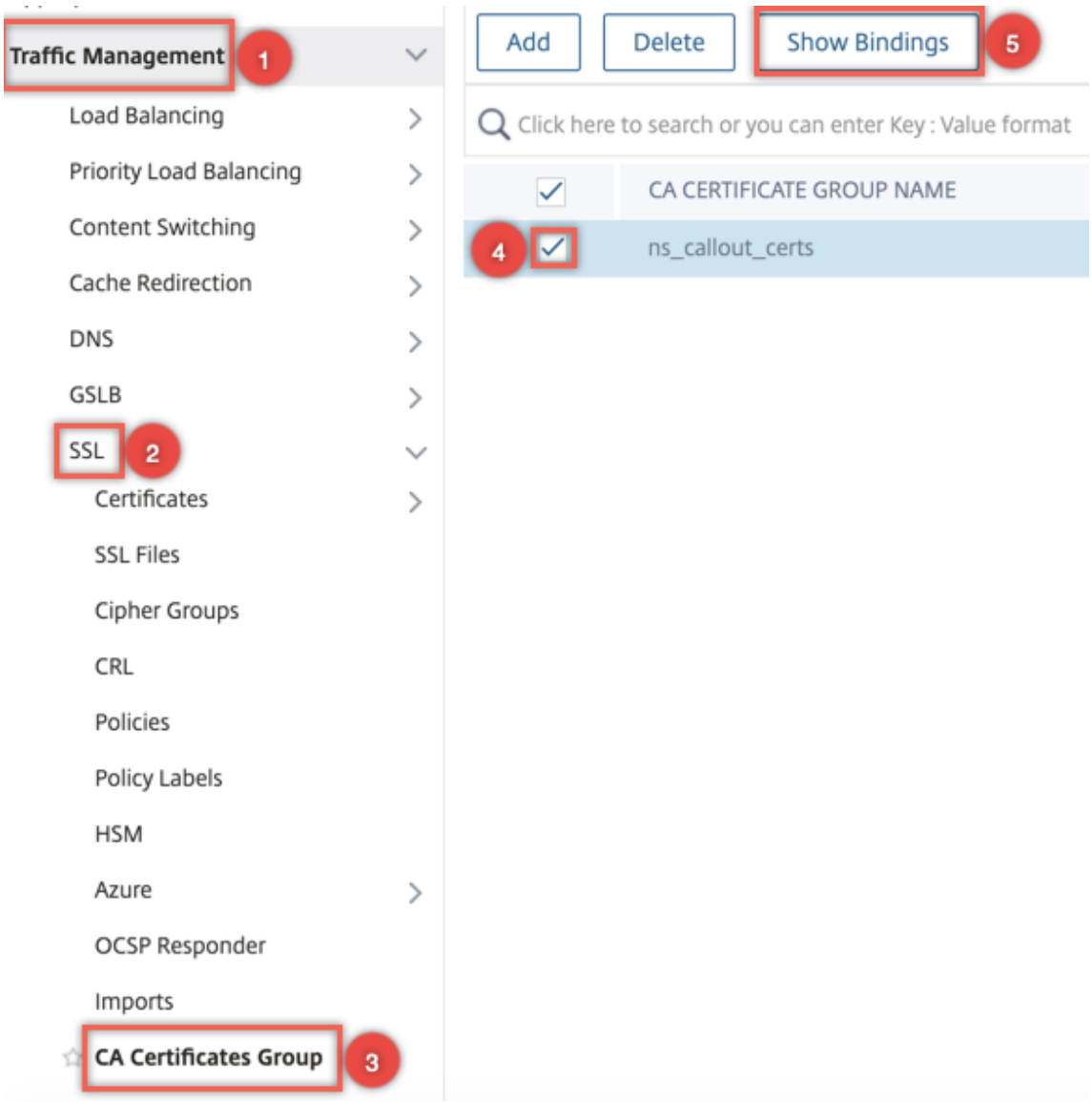


2. Enter values for the following parameters:

   - Certificate-key pair name
   - Certificate file name



3. Click **Install**.
4. Navigate to **Traffic Management > SSL > CA Certificates Group**.
5. Select **ns_callout_certs** and click **Show Bindings**.



6. Click **Bind**.
7. Select the CA certificate created earlier and click **Select**.
8. Click **Bind**, and then click **Close**.

Configure an Azure application

The Azure application entity contains the credentials required to authenticate to Azure Active Directory and obtain the access token. That is, to get authorization access to Key Vault resources and APIs, add the Azure Application ID, secret (password) and tenant ID on the ADC appliance.
When configuring the Azure Application entity using the CLI, you must enter the password. If you use the GUI, the Azure application entity contains the credentials required to authenticate to Azure Active Directory and obtain the access token.

To configure an Azure application by using the CLI

From release 13.0-61.x, a parameter, vaultResource, is added to the `add azure application` command to get the domain of the resource group before the access token is granted to the application. This parameter is added because the domain name might be different for different regions. For example, the domain might be `vault.azure.net` or `vault.usgov.net`.

At the command prompt, type:

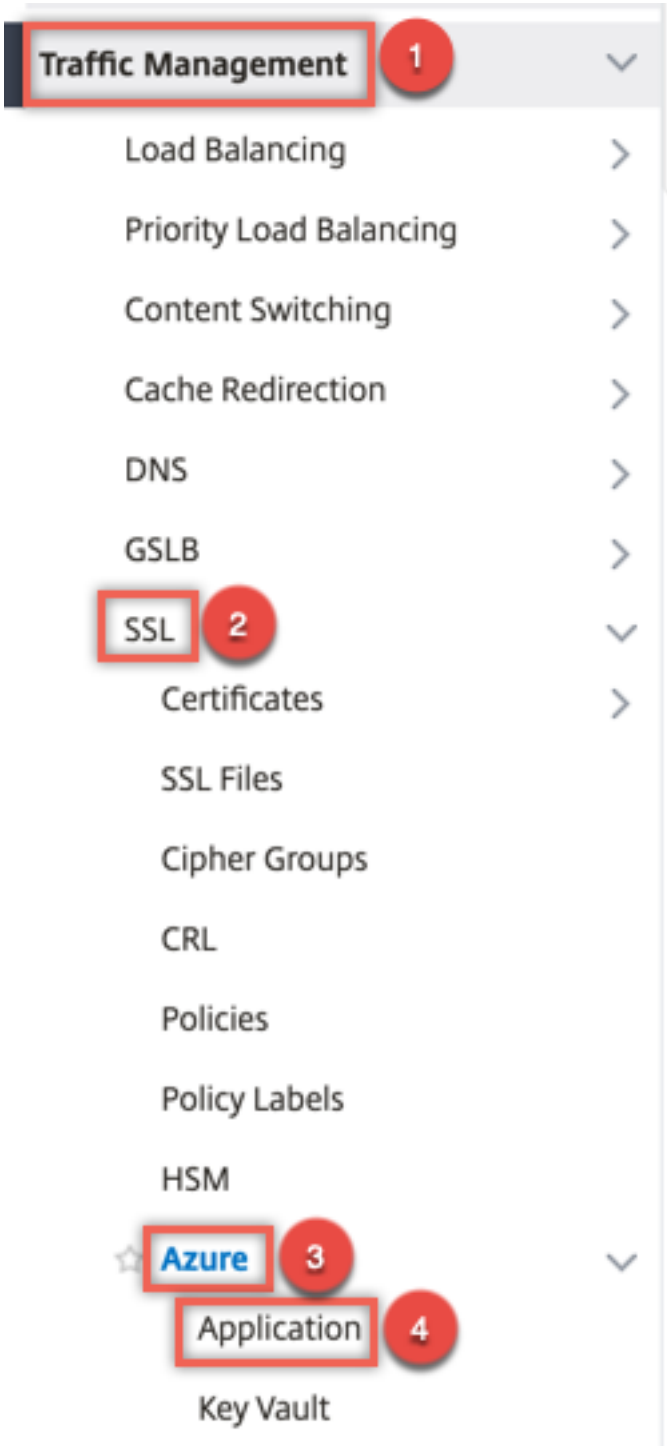
```
add azure application <name> -clientID <string> -clientSecret -tenantID
    <string> -vaultResource <string> [-tokenEndpoint <URL>]
show azure application
<!--NeedCopy-->
```

Example:

```
add azure application app10 -clientiD 12345t23aaa5 -clientsecret
   csHzOoEzmuY= -vaultResource example.vault.azure.net -tenantID 33583
   ee9ca5b
 Done
> sh azure application app10
1)  Name:  app10            ClientID:  12345t23aaa5
    TokenEndpoint:  "https://login.microsoftonline.com/33583ee9ca5b/"
    TenantID:  33583ee9ca5b  VaultResource:  example.vault.azure.net
 Done

<!--NeedCopy-->
```

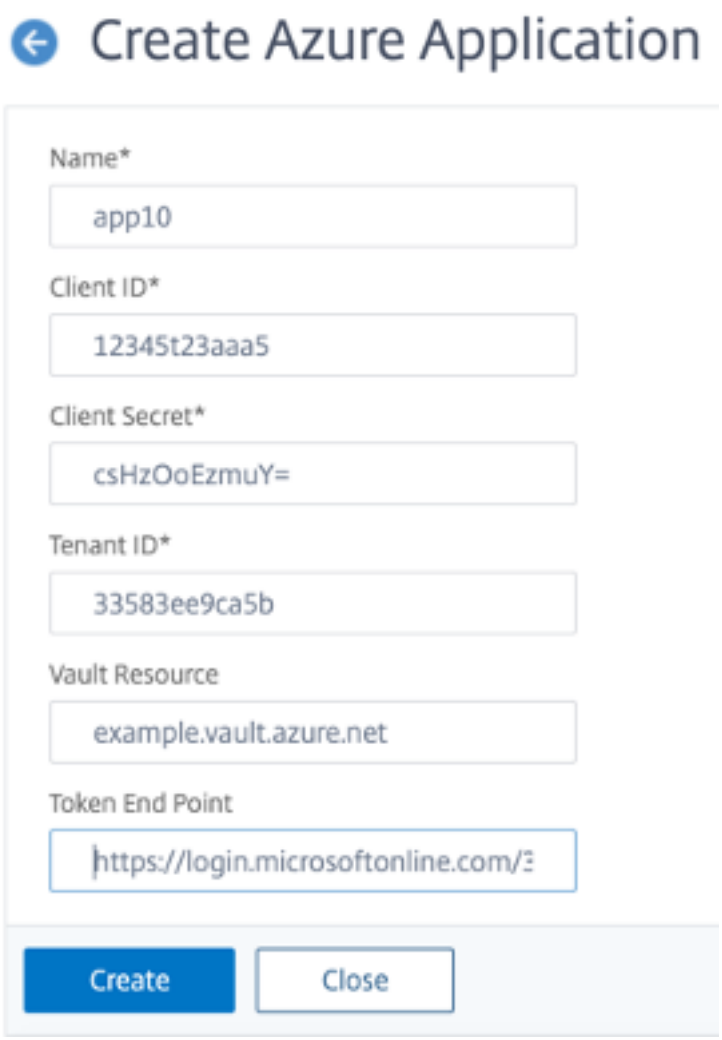
To configure an Azure application by using the GUI

1. Navigate to **Traffic Management > SSL > Azure > Application**.



2. In the details pane, click **Add**.
3. Enter values for the following parameters:
   - Name – Name for the application object on the Citrix ADC appliance.
   - Client ID – Application ID that is generated when an application is created in Azure Active Directory using either the Azure CLI or the Azure portal (GUI).

- Client Secret – Password for the application configured in Azure Active Directory. The password is specified in the Azure CLI or generated in the Azure portal (GUI).
- Tenant ID – ID of the directory inside Azure Active Directory in which the application was created.
- Vault Resource - Vault resource for which access token is granted. Example `vault.azure.net`.
- Token End point – URL from where the access token can be obtained. If the token end point is not specified, the default value is `https://login.microsoftonline.com/<tenant id>`.



Configure Azure Key Vault

Create an Azure Key Vault object on the ADC appliance.

To configure Azure Key Vault by using the CLI

At the command prompt, type:

```
add azure keyVault <name> -azureVaultName <string> -azureApplication
       <string>
show azure keyvault
<!--NeedCopy-->
```

Example:

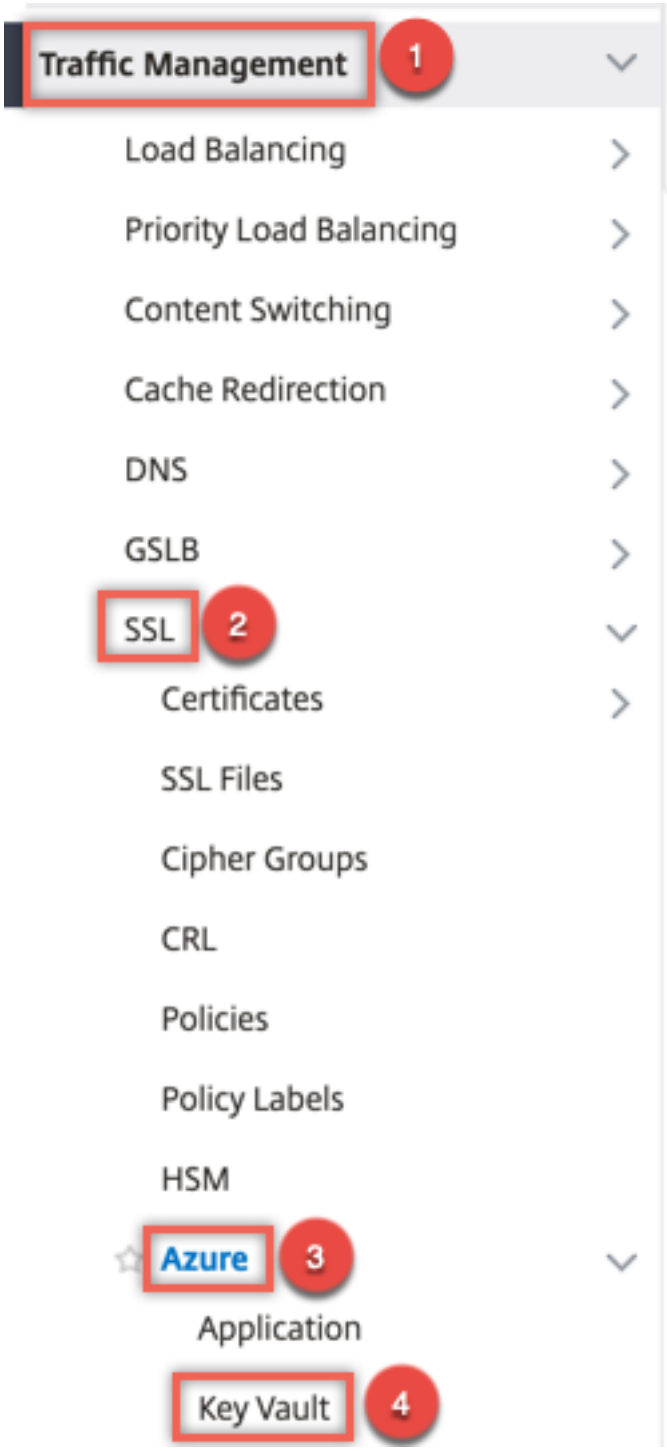
```
add azure keyvault kv1 -azureapplication app10 -azurevaultName pctest.
   vault.azure.net
> sh azure keyVault
1)  Name:  kv1               AzureVaultName:  pctest.vault.azure.net
  AzureApplication:  app10  State:  "Access token obtained"
 Done
<!--NeedCopy-->
```

The following table lists the different values that the state of the Azure Key Vault can take along with a brief description about each state.

State	Description
Created	Initial state of the Key Vault object. Authentication has not been attempted.
Could not reach token end point	Indicates one of the following: DNS server not configured, issuer certificate not bound to a CA certificate group, or network issues.
Authorization failed	Incorrect application credentials.
Token parse error	Response from Azure Active Directory is not in the expected format.
Access token obtained	Successfully authenticated by Azure Active Directory.

To configure the Azure Key Vault by using the GUI

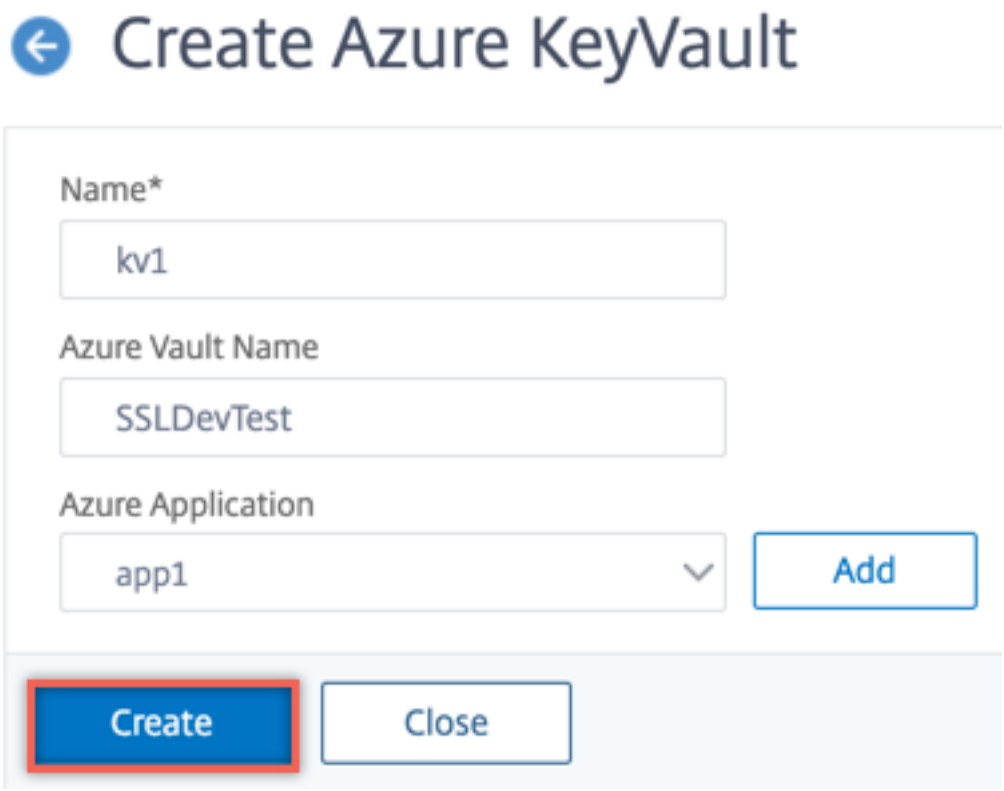
1. Navigate to **Traffic Management > SSL > Azure > Key Vault**.



2. Enter values for the following parameters:

   - Name - Name for the Key Vault.
   - Azure Key Vault Name - Name of the Key Vault configured in Azure cloud using either the Azure CLI or the Azure portal (GUI) with domain name.
   - Azure Application Name - Name of the Azure Application object created on the ADC appli-

ance. The Azure Application object with this name is used for authentication with Azure Active Directory.



Add HSM key

Storing your private key in the HSM provides FIPS 140-2 level 2 compliance.

To add an HSM key by using the CLI

At the command prompt, type:

```
add ssl hsmKey <hsmKeyName> [-hsmType <hsmType>] [-key <string> |
       -serialNum <string>] {
 -password  }
  [-keystore <string>]
<!--NeedCopy-->
```

Example:

```
add ssl hsmKey h1 -keystore kv1 -key san15key -hsmType KEYVAULT


> sh ssl hsmKey h1
```



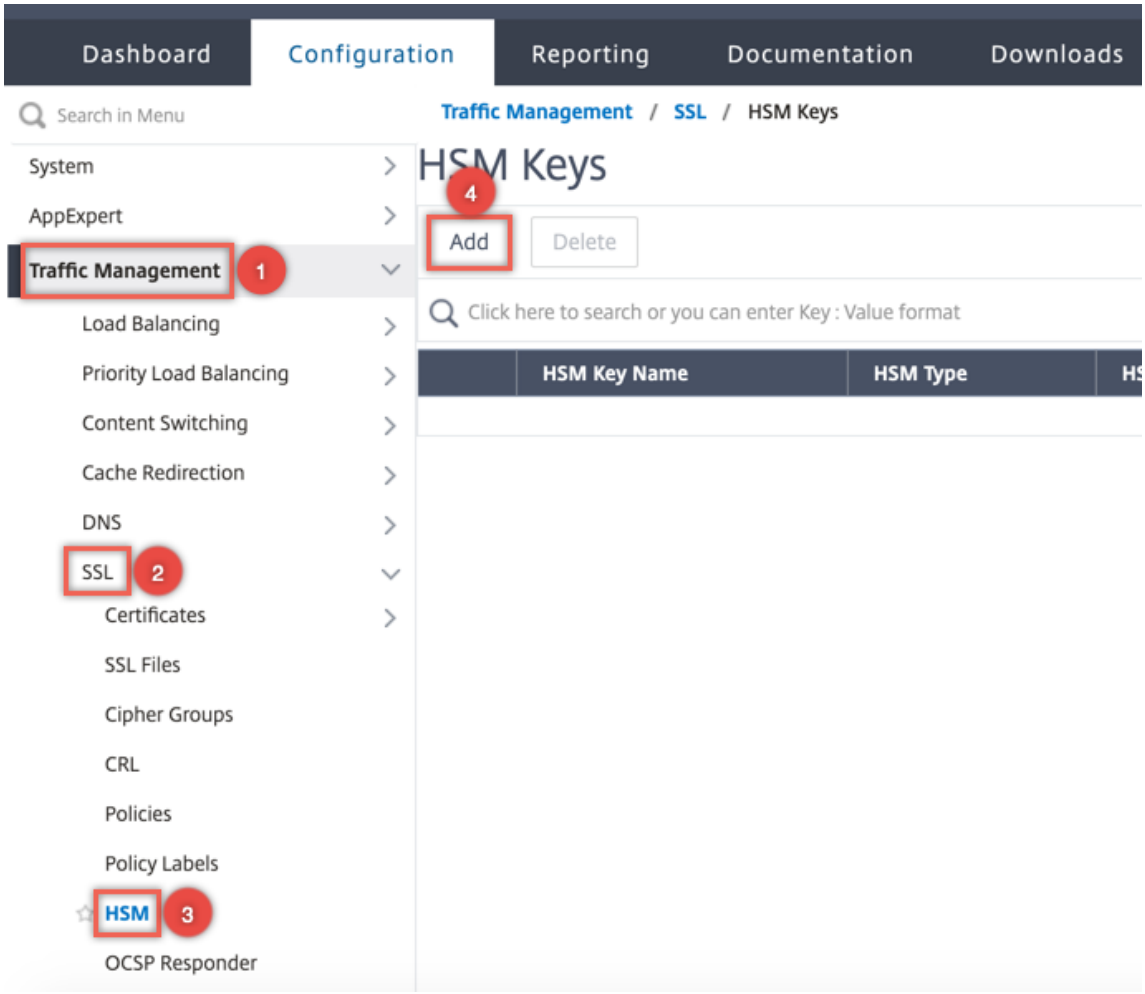
```
5      HSM Key Name: h1        Type: KEYVAULT
6      Key: san15key
7      Key store: kv1
8      State:  “Created”
9   Done
10 <!--NeedCopy-->
```

The following table lists the different values that the state of an HSM key can take along with a brief description about each state.

State	Description
Created	The HSM key is added on the ADC appliance. A key operation is not yet attempted.
Access token unavailable	Access token not available when key operation was attempted.
Unauthorized	Configured Azure application does not have permission to perform the key operation.
Does not exist	The key does not exist in the Azure Key Vault.
Unreachable	The Key Vault host is not reachable on the network.
Marked down	The HSM key is marked DOWN on the ADC appliance due to threshold errors during key operation.
Key operations successful	Success response received from the Key Vault for key operation.
Key operations failed	Failure response received from Key Vault for key operation.
Key operation throttled	The key operation request is throttled by the Key Vault.

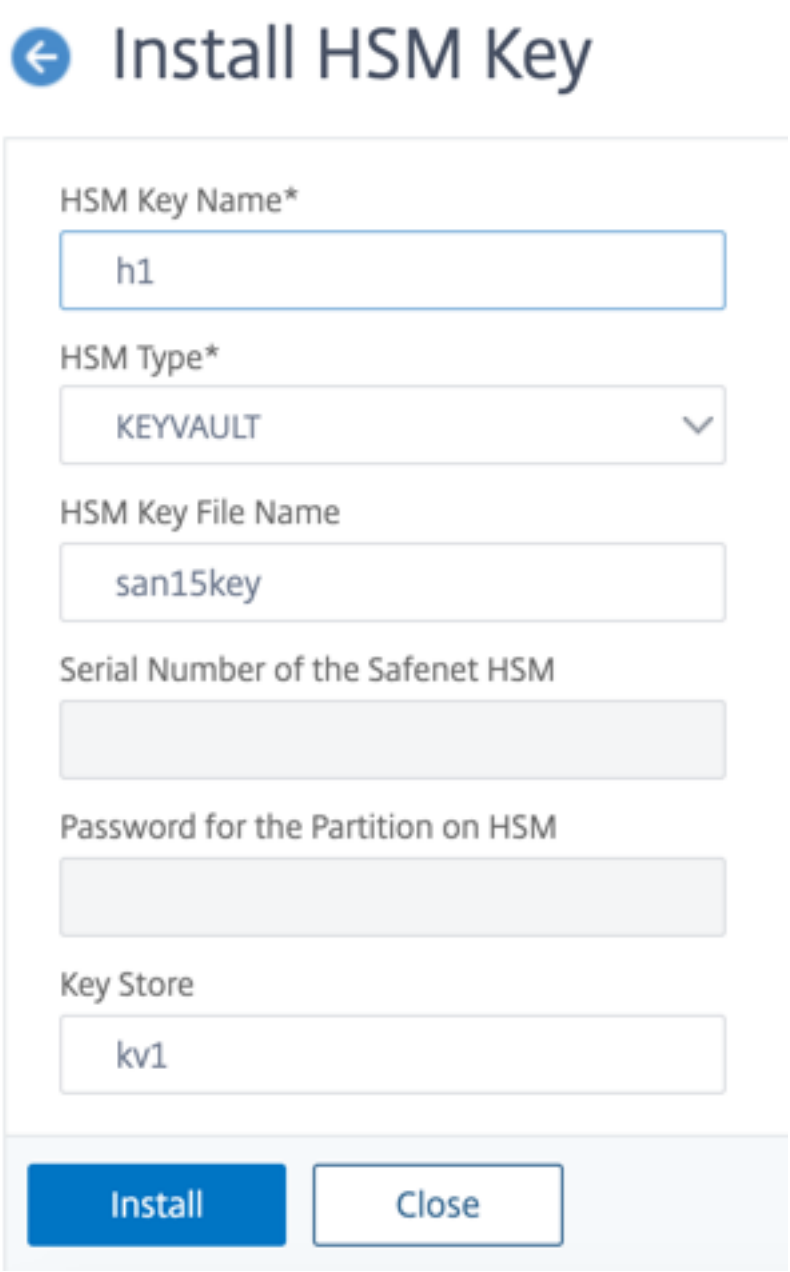
To add an HSM key by using the GUI

1. Navigate to **Traffic Management > SSL > HSM**.



2. Enter values for the following parameters.

   - HSM key name - Name of the key.
   - HSM type - Type of HSM.
   - Key store - Name of key store object representing HSM where the key is stored. For example, name of Key Vault object or Azure Key Vault authentication object. Applies only to KEYVAULT type HSM.



3. Click **Add**

Add a certificate-key pair

Add a certificate-key pair using the HSM key created earlier.

To add a certificate-key pair by using the CLI

At the command prompt, type:

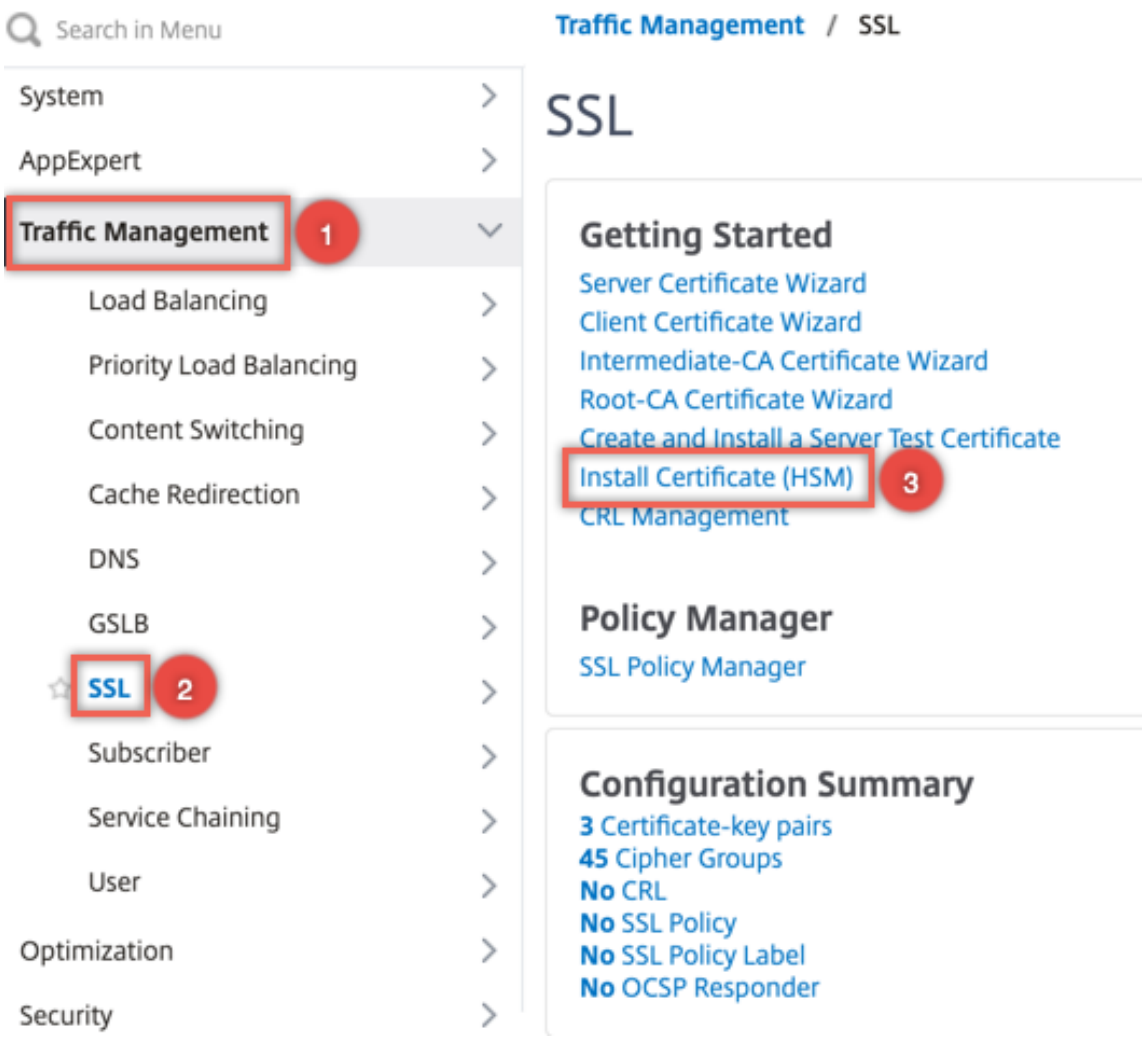

```
add ssl certKey <certkeyName> (-cert <string> [-password]) -hsmKey <
    string>]
show ssl certkey
<!--NeedCopy-->
```

Example:

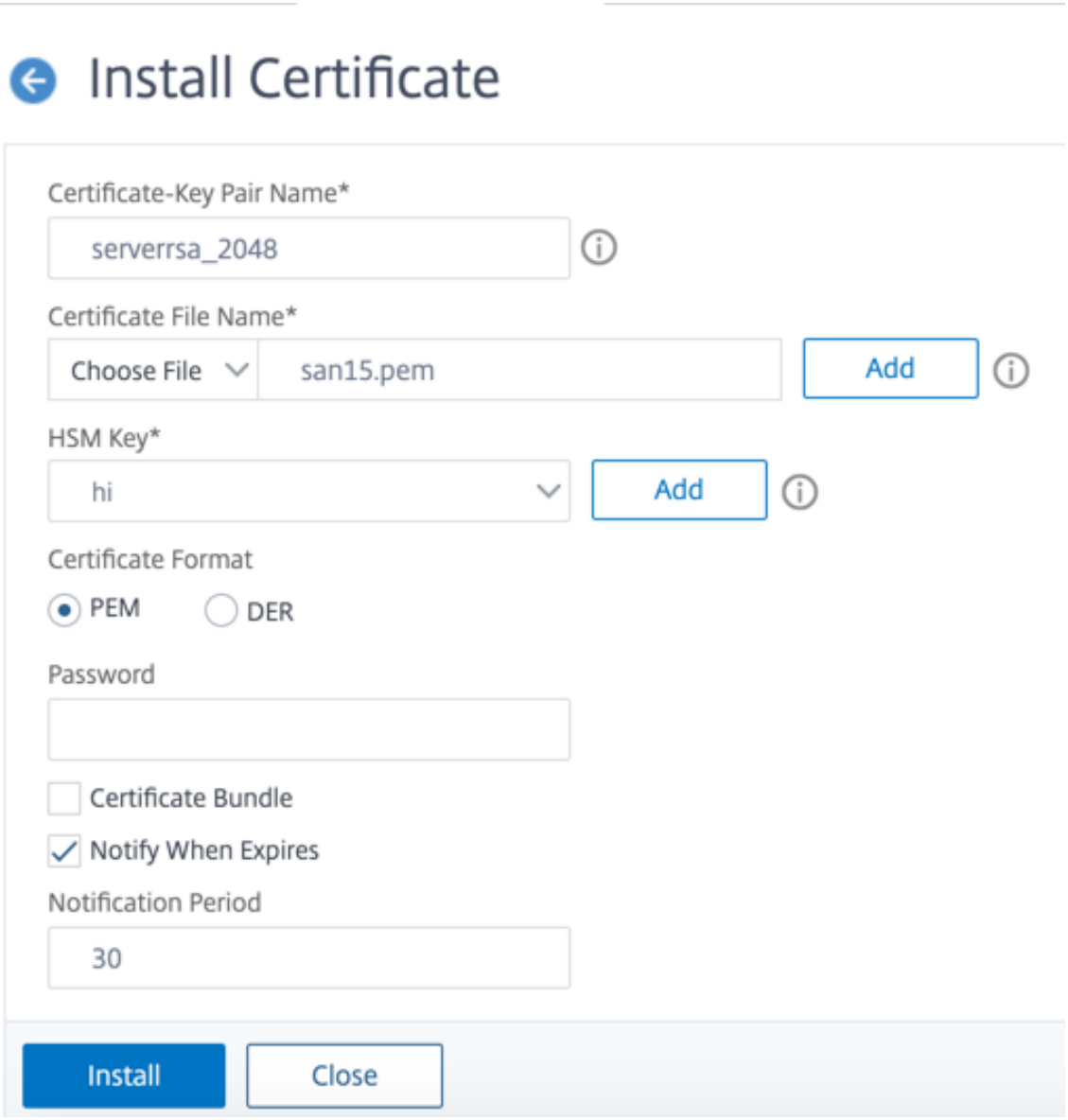
```
add ssl certKey serverrsa_2048 -cert /nsconfig/ssl/san_certs/san15.pem
    -hsmKey h1
> sh ssl certkey serverrsa_2048
    Name: serverrsa_2048        Status: Valid,   Days to expiration
        :9483
    Version: 3
    Serial Number: F5CFF9EF1E246022
    Signature Algorithm: sha256WithRSAEncryption
    Issuer:  C=in,O=citrix,CN=ca
    Validity
        Not Before: Mar 20 05:42:57 2015 GMT
        Not After : Mar 12 05:42:57 2045 GMT
    Certificate Type:   "Server Certificate"
    Subject:  C=in,O=citrix
    Public Key Algorithm: rsaEncryption
    Public Key size: 2048
    Ocsp Response Status: NONE
 Done
<!--NeedCopy-->
```

To add a certificate-key pair by using the GUI

1. Navigate to **Traffic Management > SSL > Install Certificate (HSM)**.



2. Enter values for the following parameters:
   - Certificate-Key Pair Name
   - Certificate File Name
   - HSM Key



3. Click **Install**.

Bind the certificate-key pair to a virtual server

The certificate used for processing SSL transactions must be bound to the virtual server that receives the SSL data.

To bind the SSL certificate-key pair to a virtual server by using the CLI

At the command prompt, type:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

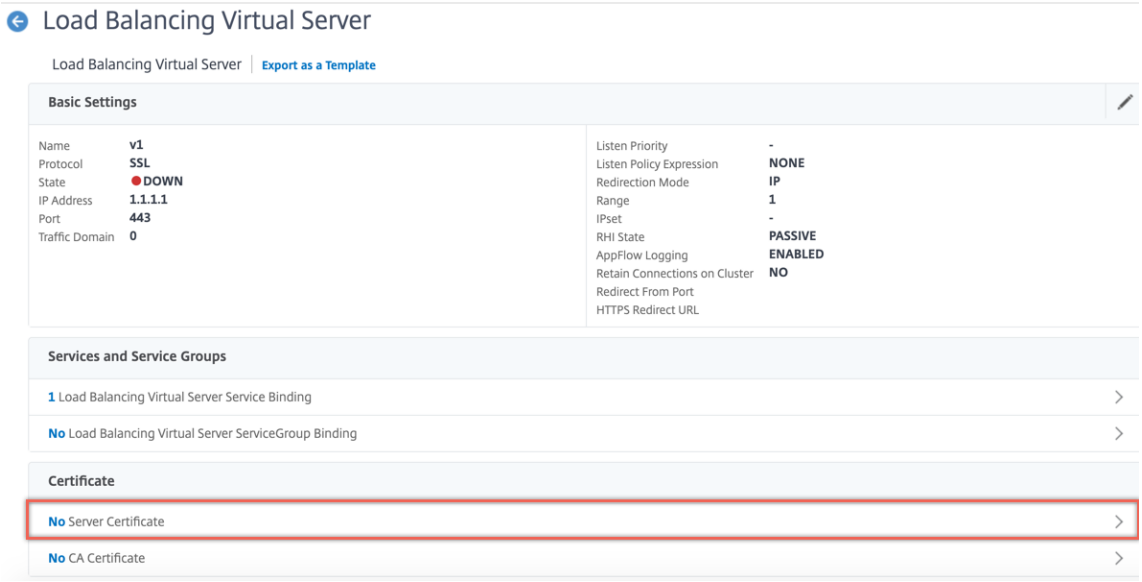

Example:

```
bind ssl vserver v1 -certkeyName serverrsa_2048

sh ssl vserver v1

    Advanced SSL configuration for VServer v1:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED      Refresh Count: 0
    Session Reuse: ENABLED      Timeout: 120 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1

    ECC Curve: P_256, P_384, P_224, P_521

1)  CertKey Name: serverrsa_2048    Server Certificate



1)  Cipher Name: DEFAULT
    Description: Default cipher list with encryption strength >= 128bit
 Done
<!--NeedCopy-->
```

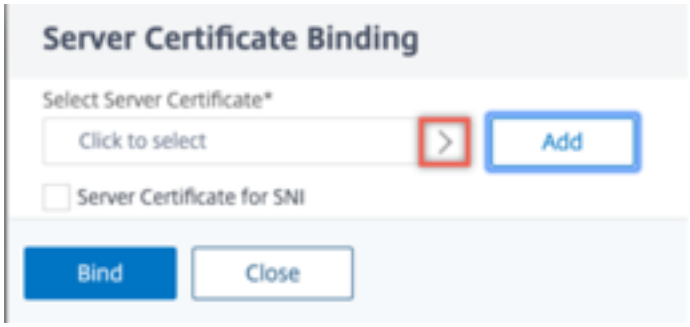
To bind an SSL certificate-key pair to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers** and open an SSL virtual

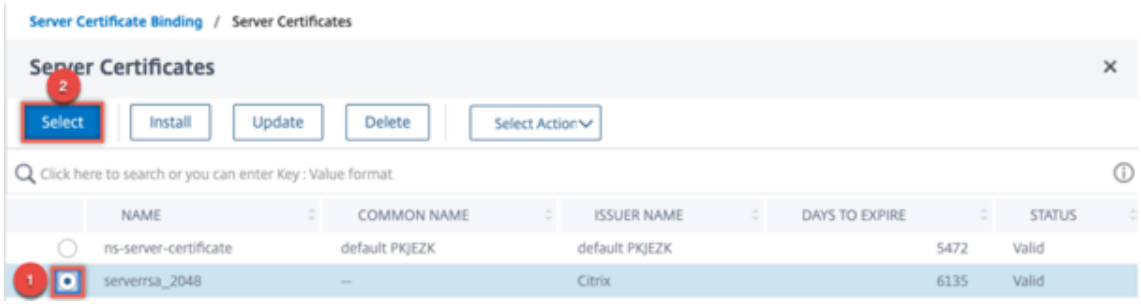
server. Click inside the Certificate section.



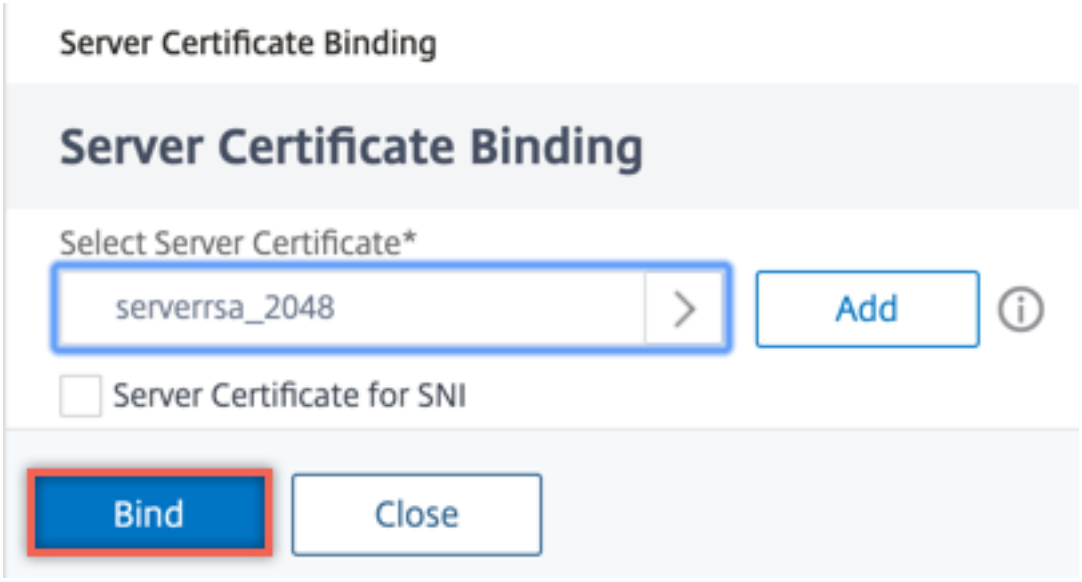
2. Click the arrow to select the certificate-key pair.



3. Select the certificate-key pair from the list.



4. Bind the certificate-key pair to the virtual server.



Limitations

- The number of concurrent calls to the Azure Key Vault for key operations is limited. Performance of the ADC appliance depends on the Key Vault limits. For more information see, Microsoft Azure Key Vault documentation.
- EC keys are not supported.
- EDT and DTLS protocols are not supported.
- ADC appliances with Intel Coleto SSL chips are not supported.
- Clustering and admin partitions are not supported.
- You cannot update the Azure Application entity, the Azure Key Vault object, and the HSM certificate-key pair after you have added them to the ADC appliance.
- A certificate bundle with HSM keys is not supported.
- An error does not appear if the HSM key and certificate do not match. While adding a certificate-key pair, ensure that the HSM key and certificate match.
- You cannot bind an HSM key to a DTLS virtual server.
- You cannot sign OCSP requests using a certificate-key pair that is created using an HSM key.
- You cannot bind a certificate-key pair to an SSL service if the certificate-key pair is created using an HSM key.

FAQ

When integrated with Azure Key Vault, are private keys stored in the ADC appliance memory?

No, private keys are not be stored in ADC appliance memory. For each SSL transaction, the appliance sends a request to Key Vault.

Is the integration FIPS 140-2 level 2 compliant?

Yes, the integrated solution provides FIPS 140-2 Level 2 support.

Which key types are supported?

Only RSA key types are supported.

What key sizes are supported?

1024-bit, 2048-bit, and 4096-bit RSA keys are supported.

Which ciphers are supported?

All ciphers supported on the ADC appliance, including TLSv1.3 ciphers with ECDHE and SHA256 are supported.

Are transactions logged?

The ADC appliance logs each transaction it makes with the Key Vault. Details such as, time, vault IP address, port, success or failure of connection, and errors are logged.
The following is a sample SSL log output.

```
1 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 896 0 :
    Backend SPCBId 30894 - ServerIP 104.211.224.186 - ServerPort 443
     - ProtocolVersion TLSv1.2 - CipherSuite "ECDHE-RSA-AES256-GCM-
    SHA384 TLSv1.2 Non-Export 256-bit" - Session New -
    SERVER_AUTHENTICATED -SerialNumber "200005
    A75B04365827852D630000000005A75B" - SignatureAlgorithm "
    sha256WithRSAEncryption" - ValidFrom "Mar 17 03:28:42 2019 GMT"
    - ValidTo "Mar 17 03:28:42 2021 GMT" - HandshakeTime 40 ms
2 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 897 0 :
    SPCBId 30894 - IssuerName " C=US,ST=Washington,L=Redmond,O=
    Microsoft Corporation,OU=Microsoft IT,CN=Microsoft IT TLS CA 2"
3 Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 898 0 :
    SPCBId 30894 - SubjectName " CN=vault.azure.net"
4 <!--NeedCopy-->
```


Troubleshooting

September 14, 2021

If the SSL feature does not work as expected after configuration, you can use some common tools to access Citrix ADC resources and diagnose the problem.

Resources for troubleshooting

For best results, use the following resources to troubleshoot an SSL issue on a Citrix ADC appliance:

- The relevant ns.log file
- The latest ns.conf file
- The messages file
- The relevant `newnslog` file
- Trace files
- A copy of the certificate files, if possible
- A copy of the key file, if possible
- The error message, if any

In addition to these resources, you can use the Wireshark application customized for the Citrix ADC trace files to expedite troubleshooting.

Troubleshooting SSL issues

To troubleshoot an SSL issue, proceed as follows:

- Verify that the Citrix ADC appliance is licensed for SSL Offloading and load balancing.
- Verify that SSL Offloading and load balancing features are enabled on the appliance.
- Verify that the status of the SSL virtual server is not displayed as DOWN.
- Verify that the status of the service bound to the virtual server is not displayed as DOWN.
- Verify that a valid certificate is bound to the virtual server.
- Verify that the service is using an appropriate port, preferably port 443.

Decrypting TLS1.3 traffic from packet trace

To troubleshoot protocols that run over TLS1.3, you must first decrypt the TLS1.3 traffic. To decrypt TLS 1.3 in Wireshark, the secrets must be exported in the `NSS` key log format. For more information about the key log format, see NSS Key Log Format.

For information about how to capture a packet trace, see Capturing SSL Session Keys During a Trace.

Note: Citrix ADC automatically logs each connection's secrets in the appropriate format for the TLS/SSL protocol version in use.

CRL refresh does not happen on the secondary node in an HA setup

The refresh does not happen because the CRL server is accessible only to the primary node through a private network.

Workaround: Add a service on the primary node with the IP address of the CRL server. This service acts as a proxy for the CRL server. When the configuration is synchronized between the nodes, CRL refresh works for both primary and secondary nodes through the service configured on the primary node.

SSL FAQs

September 14, 2021

Basic questions

HTTPS access to the GUI fails on a VPX instance. How do I gain access?

A certificate-key pair is required for HTTPS access to the GUI. On a Citrix ADC appliance, a certificate-key pair is automatically bound to the internal services. On an MPX or SDX appliance, the default key size is 1024 bytes, and on a VPX instance, the default key size is 512 bytes. However, most browsers today do not accept a key that is less than 1024 bytes. As a result, HTTPS access to the VPX configuration utility is blocked.

Citrix recommends that you install a certificate-key pair of at least 1024 bytes and bind it to the internal service for HTTPS access to the configuration utility. Alternately, update the `ns-server-certificate` to 1024 bytes. You can use HTTP access to the configuration utility or the CLI to install the certificate.

If I add a license to an MPX appliance, the certificate-key pair binding is lost. How do I resolve this problem?

If a license is not present on an MPX appliance when it starts, and you add a license later and restart the appliance, you might lose the certificate binding. Reinstall the certificate and bind it to the internal service

Citrix recommends that you install an appropriate license before starting the appliance.

What are the various steps involved in setting up a secure channel for an SSL transaction?

Setting up a secure channel for an SSL transaction involves the following steps:

1. The client sends an HTTPS request for a secure channel to the server.
2. After selecting the protocol and cipher, the server sends its certificate to the client.
3. The client checks the authenticity of the server certificate.
4. If any of the checks fail, the client displays the corresponding feedback.
5. If the checks pass or the client decides to continue even if a check fails, the client creates a temporary, disposable key. This key is called the *pre-master secret* and the client encrypts this key by using the public key of the server certificate.
6. The server, upon receiving the pre-master secret, decrypts it by using the server's private key and generates the session keys. The client also generates the session keys from the pre-master secret. Thus both client and server now have a common session key, which is used for encryption and decryption of application data.

I understand that SSL is a CPU-intensive process. What is the CPU cost associated with the SSL process?

The following two stages are associated with the SSL process:

- The initial handshake and secure channel setup by using the public and private key technology.
- Bulk data encryption by using the symmetric key technology.

Both of the preceding stages can affect server performance, and they require intensive CPU processing for of the following reasons:

1. The initial handshake involves public-private key cryptography, which is very CPU intensive because of large key sizes (1024 bit, 2048 bit, 4096 bit).
2. Encryption/decryption of data is also computationally expensive, depending on the amount of data that must be encrypted or decrypted.

What are the various entities of an SSL configuration?

An SSL configuration has the following entities:

- Server certificate
- Certificate Authority (CA) certificate
- Cipher suite that specifies the protocols for the following tasks:
  - Initial key exchange
  - Server and client authentication

    - Bulk encryption algorithm
    - Message authentication
- Client authentication
- CRL
- SSL Certificate Key Generation Tool that enables you to create the following files:
    - Certificate request
    - Self-signed certificate
    - RSA keys
    - DH parameters

I want to use the SSL offloading feature of the Citrix ADC appliance. What are the various options for receiving an SSL certificate?

You must receive an SSL certificate before you can configure the SSL setup on the Citrix ADC appliance. You can use any of the following methods to receive an SSL certificate:

- Request a certificate from an authorized certificate authority (CA).
- Use the existing server certificate.
- Create a certificate-key pair on the Citrix ADC appliance.

Note: This certificate is a test certificate signed by the test Root-CA generated by the Citrix ADC appliance. Test certificates signed by the test Root-CA are not accepted by browsers. The browser throws a warning message stating that the server's certificate cannot be authenticated.

- For anything other than test purposes, you must provide a valid CA certificate and CA key to sign the server certificate.

What are the minimum requirements for an SSL setup?

The minimum requirements for configuring an SSL setup are as follows:

- Obtain the certificates and keys.
- Create a load balancing SSL virtual server.
- Bind HTTP or SSL services to the SSL virtual server.
- Bind a certificate-key pair to the SSL virtual server.

What are the limits for the various components of SSL?

SSL components have the following limits:

- Bit size of SSL certificates: 4096.
- Number of SSL certificates: Depends on the available memory on the appliance.

- Maximum linked intermediate CA SSL certificates: 9 per chain.
- CRL revocations: Depends on the available memory on the appliance.

What are the various steps involved in the end-to-end data encryption on a Citrix ADC appliance?

The steps involved in the server-side encryption process on a Citrix ADC appliance are as follows:

1. The client connects to the SSL VIP configured on the Citrix ADC appliance at the secure site.
2. After receiving the secure request, the appliance decrypts the request and applies layer 4–7 content switching techniques and load balancing policies. Then, it selects the best available back-end web server for the request.
3. The Citrix ADC appliance creates an SSL session with the selected server.
4. After establishing the SSL session, the appliance encrypts the client request and sends it to the Web server by using the secure SSL session.
5. When the appliance receives the encrypted response from the server, it decrypts and re-encrypts the data. Then, it sends the data to the client by using the client side SSL session.

The multiplexing technique of the Citrix ADC appliance enables the appliance to reuse SSL sessions that have been established with the Web servers. Therefore, the appliance avoids the CPU intensive key exchange, known as *full handshake*. This process reduces the overall number of SSL sessions on the server and maintains end-to-end security.

Certificates and Keys

Can I place the certificate and key files at any location? Is there any recommended location to store these files?

You can store the certificate and key files on the Citrix ADC appliance or a local computer. However, Citrix recommends that you store the certificate and key files in the `/nsconfig/ssl` directory of the Citrix ADC appliance. The `/etc` directory exists in the flash memory of the Citrix ADC appliance. This action provides portability and facilitates backup and restoration of the certificate files on the appliance.

Note: Make sure that the certificate and the key files are stored in the same directory.

What is the maximum size of the certificate key supported on the Citrix ADC appliance?

A Citrix ADC appliance running a software release earlier than release 9.0 supports a maximum certificate key size of 2048 bits. Release 9.0 and later support a maximum certificate key size of 4096 bits. This limit is applicable to RSA certificates.

An MPX appliance supports certificates from 512 bits up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back end server
- 4096-bit client certificate (if client authentication is enabled on the virtual server)

A virtual appliance supports certificates from 512 bits up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back end server from release 12.0-56.x. Older releases support 2048-bit certificates.
- 2048-bit client certificate (if client authentication is enabled on the virtual server) from release 12.0-56.x.

What is the maximum size of the DH parameter supported on the Citrix ADC appliance?

The Citrix ADC appliance supports a DH parameter of maximum 2048 bits.

What is the maximum certificate-chain length, that is, the maximum number of certificates in a chain, supported on a Citrix ADC appliance?

A Citrix ADC appliance can send a maximum of 10 certificates in a chain when sending a server certificate message. A chain of the maximum length includes the server certificate and nine intermediate CA certificates.

What are the various certificate and key formats supported on the Citrix ADC appliance?

The Citrix ADC appliance supports the following certificate and key formats:

- Privacy Enhanced Mail (PEM)
- Distinguished Encoding Rule (DER)

Is there a limit for the number of certificates and keys that I can install on the Citrix ADC appliance?

No. The number of certificates and keys that can be installed is limited only by the available memory on the Citrix ADC appliance.

I have saved the certificate and key files on the local computer. I want to transfer these files to the Citrix ADC appliance by using the FTP protocol. Is there any preferred mode for transferring these files to the Citrix ADC appliance?

Yes. If using the FTP protocol, you must use binary mode to transfer the certificate and key files to the Citrix ADC appliance.

Note: By default, FTP is disabled. Citrix recommends using the SCP protocol for transferring certificate and key files. The configuration utility implicitly uses SCP to connect to the appliance.

What is the default directory path for the certificate and key?

The default directory path for the certificate and key is '/nsconfig/ssl'.

When adding a certificate and key pair, what happens if I do not specify an absolute path to the certificate and key files?

When adding a certificate-key pair, specify an absolute path to the certificate and key files. If you do not specify, the ADC appliance searches the default directory for these files and attempts to load them to the kernel. The default directory is `/nsconfig/ssl`. For example, if the cert1024.pem and rsa1024.pem files are available in the `/nsconfig/ssl` directory of the appliance, both of the following commands are successful:

```
1 add ssl certKey cert1 -cert cert1204.pem -key rsa1024.pem
2 <!--NeedCopy-->
```

```
1 add ssl certKey cert1 -cert /nsconfig/ssl/cert1204.pem -key /nsconfig/
     ssl/rsa1024.pem
2 <!--NeedCopy-->
```

I have configured a high availability setup. I want to implement the SSL feature on the setup. How must I handle the certificate and key files in a high availability setup?

In a high availability setup, you must store the certificate and key files on both the primary and the secondary Citrix ADC appliance. The directory path for the certificate and key files must be the same on both appliances before you add an SSL certificate-key pair on the primary appliance.

nCipher nShield® HSM

When integrating with nCipher nShield® HSM, do we have to keep in mind any specific configuration when adding the Citrix ADC appliance to HA?

Configure the same nCipher devices on both the nodes in HA. nCipher configuration commands don't synchronize in HA. For information about the prerequisites for nCipher nShield® HSM, see Prerequisites.

Do we have to individually integrate both the appliances with nCipher nShield® HSM and RFS? Do we need to complete this action before or after the HA setup?

You can complete the integration before or after the HA setup. If the integration is done after the HA setup, the keys imported on the primary node before configuring the secondary node are not synced to the secondary node. Therefore, Citrix recommends nCipher integration before the HA setup.

Do we need to import the key into both the primary and secondary Citrix ADC appliances, or are the keys synchronized from the primary node to the secondary node?

If nCipher is integrated on both devices before forming the HA, the keys are automatically synchronized from RFS in the process of integration.

Given that the HSM is not on the Citrix ADC appliance, but on nCipher, what happens to the keys and certificates when a node fails and is replaced?

If a node fails, you can synchronize the keys and certificates to the new node, by integrating nCipher on the new node. Then, run the following commands:

```
1 sync ha files ssl
2 force ha sync
3 <!--NeedCopy-->
```

The certificates are synchronized and added if the keys are synchronized in the process of integrating nCipher.

Ciphers

What is a NULL-Cipher?

Ciphers with no encryption are known as NULL-Ciphers. For example, NULL-MD5 is a NULL-Cipher.

Are the NULL-Ciphers enabled by default for an SSL VIP or an SSL service?

No. NULL-Ciphers are not enabled by default for an SSL VIP or an SSL service.

What is the procedure to remove NULL-Ciphers?

To remove the NULL-Ciphers from an SSL VIP, run the following command:

```
bind ssl cipher <SSL_VIP> REM NULL
<!--NeedCopy-->
```

To remove the NULL-Ciphers from an SSL Service, run the following command:

```
bind ssl cipher <SSL_Service> REM NULL -service
<!--NeedCopy-->
```

What are the various cipher aliases supported on the Citrix ADC appliance?

To list the cipher aliases supported on the appliance, at the command prompt, type:

```
sh cipher
<!--NeedCopy-->
```

What is the command to display all the predefined ciphers of the Citrix ADC appliance?

To display all the predefined ciphers of the Citrix ADC appliance, at the CLI, type:

```
show ssl cipher
<!--NeedCopy-->
```

What is the command to display the details of an individual cipher of the Citrix ADC appliance?

To display the details of an individual cipher of the Citrix ADC appliance, at the CLI, type:

```
show ssl cipher <Cipher_Name/Cipher_Alias_Name/Cipher_Group_Name>
<!--NeedCopy-->
```

Example:

```
show cipher SSL3-RC4-SHA
     1) Cipher Name: SSL3-RC4-SHA
     Description: SSLv3 Kx=RSA Au=RSA Enc=RC4(128)
    Mac=SHA1
```



```
5        Done
6  <!--NeedCopy-->
```

What is the significance of adding the predefined ciphers of the Citrix ADC appliance?

Adding the predefined ciphers of the Citrix ADC appliance causes the NULL-Ciphers to get added to an SSL VIP or an SSL service.

Is it possible to change the cipher's order without unbinding them from a cipher group on a Citrix ADC appliance?

Yes. It is possible to change the cipher's order without unbinding the ciphers from a custom cipher group. However, you cannot change the priority in inbuilt cipher groups. To change the priority of a cipher bound to an SSL entity, first unbind the cipher from the virtual server, service, or service group.

Note: If the cipher group bound to an SSL entity is empty, the SSL handshake fails because there is no negotiated cipher. The cipher group must contain at least one cipher.

Is ECDSA supported on the Citrix ADC appliance?

ECDSA is supported on the following Citrix ADC platforms. For details of supported builds, see Table 1 and Table 2 in Ciphers available on the Citrix ADC appliances.

- Citrix ADC MPX and SDX appliances with N3 chips
- Citrix ADC MPX 5900/8900/15000/26000
- Citrix ADC SDX 8900/15000
- Citrix ADC VPX appliances

Does the Citrix ADC VPX appliance support AES-GCM/SHA2 ciphers on the front-end?

Yes, AES-GCM/SHA2 ciphers are supported on the Citrix ADC VPX appliance. For details about the supported builds, see Ciphers available on the Citrix ADC appliances.

Certificates

Is the distinguished name in a client certificate available for the length of the user session?

Yes. You can access the distinguished name of the client certificate in subsequent requests during the length of the user session. That is, even after the SSL handshake is complete and the certificate is not sent again by the browser. Use a variable and an assignment as detailed in the following sample configuration:

Example:

```
add ns variable v2 -type "text(100)"

add ns assignment a1 -variable "$v2" -set       "CLIENT.SSL.CLIENT_CERT
   .SUBJECT.TYPECAST_NVLIST_T('=','/').VALUE("CN")"

add rewrite action act1 insert_http_header subject "$v2"  // example:
   to insert the distinguished name in the header

add rewrite policy pol1 true a1

add rewrite policy pol2 true act1

bind rewrite global pol1 1 next -type RES_DEFAULT

bind rewrite global pol2 2 next -type RES_DEFAULT

set rewrite param -undefAction RESET
<!--NeedCopy-->
```

Why do I need to bind the server certificate?

Binding the server certificates is the basic requirement for enabling the SSL configuration to process SSL transactions.

To bind the server certificate to an SSL VIP, at the CLI, type:

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

To bind the server certificate to an SSL service, at the CLI, type:

```
bind ssl service <serviceName> -certkeyName <cert_name>
<!--NeedCopy-->
```

How many certificates can I bind to an SSL VIP or an SSL service?

On a Citrix ADC VPX, MPX/SDX (N3), and MPX/SDX 14000 FIPS appliance, you can bind two certificates to an SSL virtual server or an SSL service if SNI is disabled. The certificates must be one each of type RSA and ECDSA. If SNI is enabled, you can bind multiple server certificates of type RSA or ECDSA. On a Citrix ADC MPX (N2) or MPX 9700 FIPS appliance, if SNI is disabled, you can bind only one certificate of type RSA. If SNI is enabled, you can bind multiple server certificates of type RSA only.

What happens if I unbind or overwrite a server certificate?

When you unbind or overwrite a server certificate, all the connections and SSL sessions created by using the existing certificate are terminated. When you overwrite an existing certificate, the following message appears:

```
ERROR:

Warning: Current certificate replaces the previous binding.
<!--NeedCopy-->
```

How do I install an intermediate certificate on a Citrix ADC appliance and link to a server certificate?

See the article at http://support.citrix.com/article/ctx114146 for information about installing an intermediate certificate.

Why am I getting a "resource already exists" error when I try to install a certificate on the Citrix ADC?

See the article at http://support.citrix.com/article/CTX117284 for instructions for resolving the "resource already exists" error.

I want to create a server certificate on a Citrix ADC appliance to test and evaluate the product. What is the procedure to create a server certificate?

Perform the following procedure to create a test certificate.

Note: A certificate created with this procedure cannot be used to authenticate all the users and browsers. After using the certificate for testing, you must obtain a server certificate signed by an authorized Root certificate authority.

To create a self-signed server certificate:

1. To create a Root CA certificate, at the CLI, type:

```
create ssl rsakey /nsconfig/ssl/test-ca.key 1024

create ssl certreq /nsconfig/ssl/test-ca.csr -keyfile /nsconfig/
   ssl/test-ca.key

Enter the required information when prompted, and then type the
   following command:

```



```
create ssl cert /nsconfig/ssl/test-ca.cer /nsconfig/ssl/test-ca.
    csr ROOT_CERT -keyfile /nsconfig/ssl/test-ca.key
<!--NeedCopy-->
```

2. Perform the following procedure to create a server certificate and sign it with the root CA certificate that you just created

   a) To create the request and the key, at the CLI, type:

```
create ssl rsakey /nsconfig/ssl/test-server.key 1024

    create ssl certreq /nsconfig/ssl/test-server.csr -keyfile
       /nsconfig/ssl/test-server.key
<!--NeedCopy-->
```

   b) Enter the required information when prompted.

   c) To create a serial-number file, at the CLI, type:

```
shell
 # echo '01' >
/nsconfig/ssl/serial.txt
 # exit
<!--NeedCopy-->
```

   d) To create a server certificate signed by the root CA certificate created in step 1, at the CLI, type:

```
create ssl cert /nsconfig/ssl/test-server.cer /nsconfig/ssl/
    test-server.csr SRVR_CERT -CAcert /nsconfig/ssl/test-ca.cer
     -CAkey /nsconfig/ssl/test-ca.key -CAserial /nsconfig/ssl/
    serial.txt
<!--NeedCopy-->
```

   e) To create a Citrix ADC cert-key pair, which is the in-memory object that holds the server certificate information for SSL handshakes and bulk encryption, at the CLI, type:

```
add ssl certkey test-certkey -cert /nsconfig/ssl/test-server.
    cer -key /nsconfig/ssl/test-server.key
<!--NeedCopy-->
```

   f) To bind the cert-key pair to the SSL virtual server, at the CLI, type:

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```


I have received a Citrix ADC appliance on which NetScaler software release 9.0 is installed. I have noticed an extra license file on the appliance. Is there any change in the licensing policy starting with NetScaler software release 9.0?

Yes. Starting with Citrix NetScaler software release 9.0, the appliance might not have a single license file. The number of license files depends on the Citrix ADC software release edition. For example, if you have installed the Advanced edition, you might need extra license files for the full functionality of the various features. However, if you have installed the Premium edition, the appliance has only one license file.

How do I export the certificate from the Internet Information Service (IIS)?

There are many ways, but by using the following method the appropriate certificate and private key for the website are exported. This procedure must be performed on the actual IIS server.

1. Open the Internet Information Services (IIS) Manager administration tool.
2. Expand the websites node and locate the SSL-enabled website that you want to serve through the Citrix ADC appliance.
3. Right-click this website and click Properties.
4. Click the Directory Security tab and, in the Secure Communications section of the window, select the View Certificate box.
5. Click the Details tab, and then click Copy to File.
6. On the Welcome to the Certificate Export Wizard page, click Next.
7. Select Yes, export the private key, and click Next.

   Note: The private key MUST be exported for SSL Offload to work on the Citrix ADC.
8. Make sure that the Personal Information Exchange -PKCS #12 radio button is selected, and select *only* the Include all certificates in the certification path if possible check box. Click Next.
9. Enter a password and click Next.
10. Enter a file name and location, and then click Next. Give the file an extension of .PFX.
11. Click Finish.

How do I convert the PKCS#12 certificate and install it on the Citrix ADC?

1. Move the exported .PFX certificate file to a location from where it can be copied to the Citrix ADC appliance. That is, to a machine that permits SSH access to the management interface of a Citrix ADC appliance. Copy the certificate to the appliance by using a secure copy utility such as SCP.

2. Access the BSD shell and convert the certificate (for example, cert.PFX) to .PEM format:

```
root@ns# openssl pkcs12 -in cert.PFX -out cert.PEM
<!--NeedCopy-->
```

3. To make sure that the converted certificate is in the correct x509 format, verify that the following command produces no error:

```
root@ns# openssl x509 -in cert.PEM -text
<!--NeedCopy-->
```

4. Verify that the certificate file contains a private key. Begin by issuing the following command:

```
root@ns# cat cert.PEM

Verify that the output file includes an RSA PRIVATE KEY section.

-----BEGIN RSA PRIVATE KEY-----
Mkm^s9KMs9023pz/s...
-----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

The following is another example of an RSA PRIVATE KEY section:

```
    Bag Attributes
    1.3.6.1.4.1.311.17.2: <No Values>
    localKeyID: 01 00 00 00
    Microsoft CSP Name: Microsoft RSA SChannel Cryptographic
    Provider
    friendlyName:
    4b9cef4cc8c9b849ff5c662fd3e0ef7e_76267e3e-6183-4d45-886e-6
       e067297b38f

    Key Attributes
    X509v3 Key Usage: 10
    -----BEGIN RSA PRIVATE KEY-----
    Proc-Type: 4,ENCRYPTED
    DEK-Info: DES-EDE3-CBC,43E7ACA5F4423968
    pZJ2SfsSVqMbRRf6ug37Clua5gY0Wld4frPIxFXyJquUHr31dilW5ta3hbIaQ+
       Rg

    ... (more random characters)
    v8dMugeRplkaH2Uwt/mWBk4t71Yv7GeHmcmjafK8H8iW80ooPO3D/ENV8X4U/
       tlh

```



```
5eU6ky3WYZ1BTy6thxxLlwAullynVXZEflNLxq1oX+ZYl6djgjE3qg==
-----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

The following is a SERVER CERTIFICATE section:

```
Bag Attributes
localKeyID: 01 00 00 00
friendlyName: AG Certificate
subject=/C=AU/ST=NSW/L=Wanniassa/O=Dave Mother
Asiapacific/OU=Support/CN=davemother.food.lan
issuer=/DC=lan/DC=food/CN=hotdog
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIKCGryDgAAAAAHzANBgkqhkiG9w0BAQUFADA8MRMwEQYK


... (more random characters) 5
   pLDWYVHhLkA1pSxvFjNJHRSIydWHc5ltGyKqIUcBezVaXyel94pNSUYx07NpPV
   /

MY2ovQyQZM8gGe3+lGFum0VHbv/y/gB9HhFesog=
-----END CERTIFICATE-----
<!--NeedCopy-->
```

The following is an INTERMEDIATE CA CERTIFICATE section:

```
Bag Attributes: <Empty Attributes>
subject=/DC=lan/DC=food/CN=hotdog
issuer=/DC=lan/DC=food/CN=hotdog
-----BEGIN CERTIFICATE-----
MIIESDCCAzCgAwIBAgIQah20fCRYTY9LRXYMIRaKGjANBgkqhkiG9w0BAQUFADA8


... (more random characters)
   Nt0nksawDnbKo86rQcNnY5xUs7c7pj2zxj/IOsgNHUp5W6dDI9pQoqFFaDk
   =

-----END CERTIFICATE-----
<!--NeedCopy-->
```

Further Intermediate CA certificates might follow, depending on the certification path of the exported certificate.

5. Open the .PEM file in a text editor

6. Locate the first line of the .PEM file and the first instance of the following line, and copy those two lines and all the lines between them:

```
-----END CERTIFICATE-----

Note: Make sure that last copied line is the first
-----END CERTIFICATE----- line in the .PEM file.

<!--NeedCopy-->
```

7. Paste the copied lines into a new file. Call the new file something intuitive, such as cert-key.pem. This certificate-key pair is for the server hosting the HTTPS service. This file must contain both the section labeled RSA PRIVATE KEY and the section labeled SERVER CERTIFICATE in the preceding example.

   Note: The certificate-key pair file contains the private key and must be kept secure.

8. Locate any subsequent sections beginning with —-BEGIN CERTIFICATE—- and ending with —END CERTIFICATE—-, and copy each such section to a separate new file.

   These sections correspond to certificates of trusted CAs that have been included in the certification path. These sections must be copied and pasted into new individual files for these certificates. For example, the INTERMEDIATE CA CERTIFICATE section of the preceding example must be copied and pasted into a new file).

   For multiple intermediate CA certificates in the original file, create files for each intermediate CA certificate in the order in which they appear in the file. Keep track (using appropriate file names) of the order in which the certificates appear, as they must be linked together in the correct order in a later step.

9. Copy the certificate-key file (cert-key.pem) and any additional CA certificate files into the /nsconfig/ssl directory on the Citrix ADC appliance.
10. Exit the BSD shell and access the Citrix ADC prompt.
11. Follow the steps in “Install the certificate-key files on the appliance” to install the key/certificate once uploaded on the device.

How do I convert the PKCS#7 certificate and install it on the Citrix ADC appliance?

You can use OpenSSL to convert a PKCS #7 Certificate to a format recognizable by the Citrix ADC appliance. The procedure is identical to the procedure for PKCS #12 certificates, except that you invoke OpenSSL with different parameters. The steps for converting PKCS #7 certificates are as follows:

1. Copy the certificate to the appliance by using a secure copy utility, such as SCP.
2. Convert the certificate (for example, cert.P7B) to PEM format:


```
openssl pkcs7 -inform DER -in cert.p7b -print_certs -text -out
    cert.pem
<!--NeedCopy-->
```

3. Follow steps 3 through 7 as described in the answer for PKCS #12 certificates.
   Note: Before loading the converted PKCS #7 certificate to the appliance, verify that it contains a private key, exactly as described in step 3 for the PKCS #12 procedure. PKCS #7 certificates, particularly the certificates exported from IIS, do not typically contain a private key.

When I bind a cipher to a virtual server or service by using the bind cipher command, I see the error message "Command deprecated."?

The command for binding a cipher to a virtual server or service has changed.

Use the `bind ssl vserver <vsername> -ciphername <ciphername>` command to bind an SSL cipher to an SSL virtual server.

Use the `bind ssl service <serviceName> -ciphername <ciphername>` command to bind an SSL cipher to an SSL service.

Note: New ciphers and cipher groups are added to the existing list and not replaced.

Why can't I create a cipher group and bind ciphers to it by using the add cipher command?

The add cipher command functionality has changed in release 10. The command only creates a cipher group. To add ciphers to the group, use the bind cipher command.

OpenSSL

How do I use OpenSSL to convert certificates between PEM and DER?

To use OpenSSL, you must have a working installation of the OpenSSL software and be able to run OpenSSL from the command line.

x509 certificates and RSA keys can be stored in several different formats.

Two common formats are:

- DER (a binary format used primarily by Java and Macintosh platforms)
- PEM (a base64 representation of DER with header and footer information, which is used primarily by UNIX and Linux platforms).

A key and the corresponding certificate, in addition to the root and any intermediate certificates, can also be stored in a single PKCS#12 (.P12, .PFX) file.

Procedure

Use the **OpenSSL** command to convert between formats as follows:

1. To convert a certificate from PEM to DER:

```
x509 -in input.crt -inform PEM -out output.crt -outform DER
<!--NeedCopy-->
```

2. To convert a certificate from DER to PEM:

```
x509 -in input.crt -inform DER -out output.crt -outform PEM
<!--NeedCopy-->
```

3. To convert a key from PEM to DER:

```
rsa -in input.key -inform PEM -out output.key -outform DER
<!--NeedCopy-->
```

4. To convert a key from DER to PEM:

```
rsa -in input.key -inform DER -out output.key -outform PEM
<!--NeedCopy-->
```

Note: If the key you are importing is encrypted with a supported symmetric cipher, you are prompted to enter the pass-phrase.

Note: To convert a key to or from the obsolete NET (Netscape server) format, substitute NET for PEM or DER as appropriate. The stored key is encrypted in a weak unsalted RC4 symmetric cipher, so a pass-phrase is requested. A blank pass-phrase is acceptable.

System Limits

What are the important numbers to remember?

1. Create Certificate Request:
   - Request File Name: Maximum 63 characters
   - Key File Name: Maximum 63 characters
   - PEM Passphrase (For Encrypted Key): Maximum 31 characters
   - Common Name: Maximum 63 characters
   - City: Maximum 127 characters
   - Organization Name: Maximum 63 characters
   - State/Province Name: Maximum 63 characters
   - Email Address: Maximum 39 Characters

- Organization Unit: Maximum 63 characters
- Challenge Password: Maximum 20 characters
- Company Name: Maximum 127 characters

2. Create Certificate:

   - Certificate File Name: Maximum 63 characters
   - Certificate Request File Name: Maximum 63 characters
   - Key File Name: Maximum 63 characters
   - PEM Passphrase: Maximum 31 characters
   - Validity Period: Maximum 3650 days
   - CA Certificate File Name: Maximum 63 characters
   - CA Key File Name: Maximum 63 characters
   - PEM Passphrase: Maximum 31 characters
   - CA Serial Number File: Maximum 63 characters

3. Create and Install a Server Test Certificate:

   - Certificate File Name: Maximum 31 characters
   - Fully Qualified Domain Name: Maximum 63 characters

4. Create Diffie-Hellman (DH) key:

   - DH File Name (with path): Maximum 63 characters
   - DH Parameter Size: Maximum 2048 bits

5. Import PKCS12 key:

   - Output File Name: Maximum 63 characters
   - PKCS12 File Name: Maximum 63 characters
   - Import Password: Maximum 31 characters
   - PEM Passphrase: Maximum 31 characters
   - Verify PEM Passphrase: Maximum 31 characters

6. Export PKCS12

   - PKCS12 File Name: Maximum 63 characters
   - Certificate File Name: Maximum 63 characters
   - Key File Name: Maximum 63 characters
   - Export Password: Maximum 31 characters
   - PEM Passphrase: Maximum 31 characters

7. CRL Management:

   - CA Certificate File Name: Maximum 63 characters
   - CA Key File Name: Maximum 63 characters
   - CA Key File Password: Maximum 31 characters

- Index File Name: Maximum 63 characters
- Certificate File Name: Maximum 63 characters

8. Create RSA Key:
    - Key File Name: Maximum 63 characters
    - Key Size: Maximum 4096 bits
    - PEM Passphrase: Maximum 31 characters
    - Verify Passphrase: Maximum 31 characters
9. Change advanced SSL settings:
    - Maximum CRL memory size: Maximum 1024 Mbytes
    - Encryption trigger timeout (10 mS ticks): Maximum 200
    - Encryption trigger packet count: Maximum 50
    - OCSP cache size: Maximum 512 Mbytes
10. Install Certificate:
    - Certificate-Key pair Name: Maximum 31 characters
    - Certificate File Name: Maximum 63 characters
    - Private Key File Name: Maximum 63 characters
    - Password: Maximum 31 characters
    - Notification Period: Maximum 100
11. Create Cipher Group:
    - Cipher Group Name: Maximum 39 characters
12. Create CRL:
    - CRL Name: Maximum 31 characters
    - CRL File: Maximum 63 characters
    - URL: Maximum 127 characters
    - Base DN: Maximum 127 characters
    - Bind DN: Maximum 127 characters
    - Password: Maximum 31 characters
    - Days: Maximum 31
13. Create SSL Policy:
    - Name: Maximum 127 characters
14. Create SSL Action:
    - Name: Maximum 127 characters
15. Create OCSP Responder:
    - Name: Maximum 32 characters

- URL: Maximum 128 characters
- Batching Depth: Maximum 8
- Batching Delay: Maximum 10000
- Produced At Time Skew: Maximum 86400
- Request Time-out: Maximum120000

16. Create Virtual Server:
    - Name: Maximum 127 characters
    - Redirect URL: Maximum 127 characters
    - Client Time-out: Maximum 31536000 secs
17. Create Service:
    - Name: Maximum 127 characters
    - Idle Time-out (secs):
      Client: Maximum 31536000
      Server: Maximum 31536000
18. Create Service Group:
    - Service Group Name: Maximum 127 characters
    - Server ID: Maximum 4294967295
    - Idle Time-out (secs):
      Client: Maximum value 31536000
      Server: Maximum 31536000
19. Create Monitor:
    - Name: Maximum 31 characters
20. Create Server:
    - Server Name: Maximum 127 characters
    - Domain Name: Maximum 255 characters
    - Resolve Retry: Maximum 20939 secs

Content inspection

September 14, 2021

In recent times, there is an expansion of device types to display various multimedia content. The device types can be of mobile handsets to tablets, and to desktops. Intermediate infrastructure providers need to transform the original content from a web server to a format suitable for the device that asks for the content. The external devices inspect the content that transcodes and send it back

to the client. Commonly used protocol to achieve this is ICAP. ICAP enables the Citrix ADC appliance to be put in various deployments. ICAP uses the content inspection technique that inspects data for malware and security issues.

> **Note**
>
> HTTP/2 is not compatible with content inspection. The applications using the HTTP/2 might not function properly if the traffic is sent through the content inspection.

ICAP for remote content inspection

September 14, 2021

The Internet Content Adaptation Protocol (ICAP) is a simple lightweight protocol for running the value-added transformation service on HTTP messages. In a typical scenario, an ICAP client forwards HTTP requests and responses to one or more ICAP servers for processing. The ICAP servers perform content transformation on the requests and send back responses with appropriate action to take on the request or response.

ICAP on a Citrix ADC appliance

In a Citrix ADC setup, the appliance acts as an ICAP client interoperating with third-party ICAP servers (such as antimalware and Data Loss Protection (DLP)). When the appliance receives an incoming web traffic, the appliance intercepts the traffic and uses a Content Inspection policy to evaluate if the HTTP request needs an ICAP processing. If yes, the appliance decrypts and sends the message as a plain text to the ICAP servers. The ICAP servers run the content transformation service on the request message and send back a response to the appliance. The adapted messages can either be an HTTP request or an HTTP response. If the appliance interoperates with multiple ICAP servers, the appliance performs load balancing of ICAP servers. This scenario happens when one ICAP server is not sufficient to handle all the traffic load. Once the ICAP servers return a modified message, the appliance forwards the modified message to the back-end origin server.

The Citrix ADC appliance also provides a secured ICAP service if the incoming traffic is an HTTPS type. The appliance uses an SSL based TCP service to establish a secured connection between the appliance and ICAP servers.

How ICAP request modification (REQMOD) works

In the request modification (REQMOD) mode, the Citrix ADC appliance forwards the HTTP request received from the client to the ICAP server. The ICAP server then performs one of the following:

1. Sends back a modified version of the request and the appliance in turn sends the modified request to the back-end origin server or it pipelines the modified request to another ICAP server.
2. Responds with a message indicating no adaptation is required.
3. Returns an error and the appliance in turn sends the error message back to the user.

How ICAP response modification (RESPMOD) works

In the response modification (RESPMOD) mode, the Citrix ADC appliance sends an HTTP response to the ICAP server (the response sent by the appliance is typically the response sent by the origin server). The ICAP server then performs one of the following:

1. Sends a modified version of the response and the appliance in turn sends the response to the user or pipelines the response to another ICAP server.
2. Responds with a message indicating no adaptation is required.
3. Returns an error and the appliance in turn sends the error message to the user.

ICAP license

The ICAP feature works on a Citrix ADC standalone or high availability setup with Citrix ADC Premium or Advanced license edition.

Configure ICAP for content transformation service

To use ICAP for content transformation service, you must begin by enabling the Content inspection and load balancing features. Once you enable the features, you can complete the following tasks

To enable content inspection

If you want the Citrix ADC appliance to act as an ICAP client, you must first enable the Content Inspection and load balancing features.

At the command prompt, type:

```
enable ns feature contentInspection LoadBalancing
<!--NeedCopy-->
```

Add ICAP profile

ICAP configurations for a Citrix ADC appliance are specified in an entity called the ICAP profile. The profile has a collection of the ICAP settings. The settings include parameters to dynamically generate an ICAP request, receive the ICAP response, and log content inspection data.

To dynamically generate an ICAP request to the ICAP server, a new parameter, “insertHTTPRequest” is added to the ICAP profile. If this parameter is configured, the appliance takes the configured value as a policy expression and evaluates the expression and includes the result as an encapsulated HTTP request or response and then sent it to the ICAP server. Also, a new parameter “insertICAPHeaders” is configurable to dynamically evaluate and include the ICAP headers.

When the appliance sends an ICAP request and does not receive a response the ICAP server, the connection becomes unresponsive. It occurs until the ICAP server sends a response or a session gets freed. The behavior can be handled by configuring the ICAP response time-out option. You can set a request time-out parameter for action if there is delayed ICAP response. If the Citrix ADC appliance does not receive a response within the configured request time-out, the request timeout action is performed.

ReqTimeoutAction: Possible values are BYPASS, RESET, DROP.
BYPASS: This Ignores the remote ICAP server’s response and sends the request/response to Client/Server.
RESET (default): Reset the client connection by closing it.
DROP: Drop the request without sending a response to the user

To evaluate an ICAP response, a new policy expression `ICAP.RES` is used in the content inspection callout return expression. This expression evaluates the ICAP response similar to the `HTTP.RES` expression in a `HTTP_CALLOUT`.

For example, when a Citrix ADC appliance receives an HTTP request for a service hosted behind the Citrix ADC virtual IP address, the appliance might have to check the client’s authentication with an external server and take an action.

At the command prompt, type:

```
add ns icapProfile <name> [-preview ( ENABLED | DISABLED )][-previewLength
 <positive_integer>] -uri <string> [-hostHeader <string>] [-userAgent <
string>] -Mode ( REQMOD | RESPMOD )[-queryParams <string>] [-connectionKeepAlive
 ( ENABLED | DISABLED )][-allow204 ( ENABLED | DISABLED )] [-insertICAPHeaders
 <string>][-insertHTTPRequest <string>] [-reqTimeout <positive_integer>][-
reqTimeoutAction <reqTimeoutAction>] [-logAction <string>]
```

Example:

```
add icapprofile reqmod-profile -mode RESPMOD -uri  “/req_scan”  -hostHeader
 “Webroot.reqsca”  -useragent  “NS_SWG-Proxy”
```

```
add ns icapProfile icap_prof1 -uri "/example"-Mode REQMOD -reqtimeout 4 -
reqtimeoutaction BYPASS
```

```
> add icapProfile reqmode-profile -uri '/example'-mode REQMOD -insertHTTPRequest
 q{ HTTP.REQ.METHOD + ""+ HTTP.REQ.URL + "HTTP/1.1\r\n"+ "Host: "+ HTTP.REQ
```



```
.HOSTNAME + "\r\n\r\n"}
```

Log ICAP content inspection action

To dynamically generate content inspection log stream records or SYSLOG logs, you can use the ICAP.RES based policy expression on the ICAP response. This parameter is configurable in the ICAP profile to configure the policy expression to generate the dynamic log records.

At the command prompt, type:

```
add audit messageaction icap_log_expr INFORMATIONAL icap.res.full_header
```

```
set icapProfile reqmode-profile -logAction messageaction
```

Add ICAP service as a TCP or SSL_TCP service

After you enable the Content Inspection feature, you must add an ICAP service for the ICAP servers that will be part of the load balancing setup. The service that you add provides the ICAP connection between the Citrix ADC appliance and load balancing virtual servers.

Note: As an administrator, you can add an ICAP service and directly configure the ICAP server IP address in the Content Inspection action.

At the command prompt, type the following:

```
1 add service <name> <IP> <serviceType> <port>
2 <!--NeedCopy-->
```

Example:

```
add service icapsv1 10.10.10.10 SSL_TCP 1345
```

```
add service icapsv2 10.10.10.11 SSL_TCP 1345
```

Add a TCP or SSL_TCP based load balancing virtual server

After creating an ICAP service, you must create a virtual server to accept ICAP traffic and load balance the ICAP servers.

> **Note:**
>
> You can also use an SSL based TCP service over a secured channel. You use a SSL_TCP service and bind to the Content Inspection action.

At the command prompt, type the following:

```
1 add lb vserver <name> <serviceType> <port>
2 <!--NeedCopy-->
```


Example:

```
add lb vserver vicap TCP 0.0.0.0.0 - persistenceType NONE -cltTimeout
    9000

add lb vserver vicap SSL_TCP 0.0.0.0 0 - persistenceType NONE -
    cltTimeout 9000
<!--NeedCopy-->
```

Bind ICAP service to the load balancing virtual server

After you create an ICAP service and virtual server, you must bind the ICAP service to the virtual server.

At the command prompt, type the following:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver vicap icapsv1
<!--NeedCopy-->
```

Add content inspection action

After you enable the Content Inspection feature, you must add an ICAP action for handling the ICAP request information. The ICAP profile and services, or load balancing virtual server that are created are bound to the ICAP action. If the ICAP server is down, you can configure the `ifserverdown` parameter for the appliance to perform any one of the following actions.

CONTINUE: If the User wants to bypass the content inspection when the remote server is down, you can choose the “CONTINUE” action, as default.
RESET (default): This action responds to the client by closing the connection with RST.
DROP: This action silently drops the packets without sending a response to the user.

At the command prompt, type the following:

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>

add ContentInspection action <name> -type ICAP -serverip <ip> -
    serverport <port> -icapProfileName <string>
<!--NeedCopy-->
```


> **Note:**
>
> If you can configure the ICAP service instead of a load balancing virtual server, you can mention the service name in the \<-serverip> option. When adding the Content Inspection action, the TCP service is automatically created for the given IP address with port 1344 and it is used for ICAP communication.

Example:

```
add ContentInspection action ci_act_lb -type ICAP -serverName vicap -
    icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName icapsv1
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverip 1.1.1.1 -
    serverport 1344 -icapProfileName icap_reqmod
<!--NeedCopy-->
```

Add content inspection policies

After you create a Content Inspection action, you must create content inspection policies to evaluate requests for ICAP processing and audit logging. The policy is based on a rule which consists of one or more expressions. The rule is associated to the content inspection action that is associated if a request matches the rule.

At the command prompt, type the following:

```
add contentInspection policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

Example:

```
add ContentInspection policy ci_pol_basic - rule true - action
    ci_act_svc

add ContentInspection policy ci_pol_HTTP - rule HTTP.REQ.URL.CONTAINS(
    "html") - action ci_act_svc
<!--NeedCopy-->
```

Bind content inspection policies to the content switching or load balancing virtual server

To put an ICAP policy into effect, you must bind it globally or bind it to a Content Switching or load balancing virtual server, which front end the application. When you bind the policy, you must assign

a priority to it. The priority determines the order in which the policies you define are evaluated.

> **Note:**
>
> The application virtual server must be of type - HTTP/SSL/CS-PROXY.

For information about configuring a load balancing setup for forwarding the traffic to the back-end origin server after content transformation, see Load Balancing.

Configure secured ICAP service

To establish a secured connection between the Citrix ADC appliance and the ICAP web servers, the appliance uses an SSL-based TCP service or load balancing virtual server bound to an ICAP action.

To establish a secured ICAP connection, complete the following tasks:

1. Add SSL-based TCP service.
2. Bind SSL-based TCP service to load balancing virtual server of type TCP or SSL_TCP.
3. Bind SSL-based TCP service or load balancing virtual server to Content Inspection action.

Add SSL-based TCP service to load balancing virtual server

To establish a secured connection between the Citrix ADC appliance and the ICAP web servers, the appliance uses an SSL-based TCP service or load balancing virtual server bound to an ICAP action.

To establish a secured ICAP connection, complete the following tasks:

1. Add SSL-based TCP service.
2. Bind SSL-based TCP service to load balancing virtual server of type TCP or SSL_TCP.

Bind SSL-based TCP service or load balancing virtual server to Content Inspection action

Add SSL-based TCP service to load balancing virtual server

After you enable the Content Inspection feature, you must add a secured ICAP service that will be part of the load balancing setup. The service that you add provides a secured ICAP connection between the Citrix ADC appliance and load balancing virtual servers.

At the command prompt, type the following:

```
add service <name> <IP> <serviceType> <port>
<!--NeedCopy-->
```

Example:


```
add service icapsv2 10.102.29.200 SSL_TCP 1344 -gslb NONE -maxclient
    0 -maxReq 0 -cip DISABLED -usip NO -useproxport YES -sp ON -
    cltTimeout 9000 -svrTimeout 9000 -CKA NO -TCPB NO -CMP NO
<!--NeedCopy-->
```

Bind SSL-based TCP service to SSL_TCP or TCP load balancing virtual server

After you create a secured ICAP service, you must bind the service to the load balancing virtual server. It is required if you are using a load balancing virtual server to load balance the ICAP servers.

At the command prompt, type the following:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver vicap icapsv2
<!--NeedCopy-->
```

Bind SSL-based TCP service or load balancing virtual server to the Content Inspection action

You add an ICAP action for handling the ICAP request information and also bind the SSL-based TCP service to the action.

At the command prompt, type the following:

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>
<!--NeedCopy-->
```

Example:

```
add ContentInspection action ci_act_svc -type ICAP -serverName icapsv2
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName vicap -
    icapProfileName icap_reqmod
<!--NeedCopy-->
```

Configure ICAP protocol by using the GUI

1. Navigate to **Load Balancing** > **Services** and click **Add**.

2. In the **Services** page, enter the service details.
3. Navigate to **Load Balancing** > **Virtual Servers**. Add a load balancing virtual server of type HTTP/SSL. Or, you can select a virtual server and click **Edit**.
4. After entering the server basic details, click **Continue**.
5. In the **Advanced Settings** section, click **Policies**.
6. Go the **Policies** section and click the **Pencil** icon to configure the Content Inspection policy.
7. On the **Choose Policy** page, select **Content Inspection**. Click **Continue**.
8. In the **Policy Binding** section, click **+** to add a Content Inspection policy.
9. In the **Create ICAP Policy** page, enter a name for the policy.
10. In the **Action** field, click the "+" sign to add an ICAP action.
11. In the **Create ICAP Action** page, enter a name for the action.
12. Enter a name for the action.
13. In the **Server Name** field, enter the name of the TCP service already created.
14. In **ICAP Profile** field, click the "+" sign to add an ICAP profile.
15. In the **Create ICAP Profile** page, enter a profile name, URI, and MODE.
16. Click **Create**.
17. In the **Create ICAP Action** page, click **Create**.
18. In the **Create ICAP Policy** page, enter "true" in the **Expression Editor** and then click **Create**.
19. Click **Bind**.
20. When prompted to enable the Content Inspection feature, click **Yes**.
21. Click **Done**.

For information about the Citrix ADC GUI configuration for load balancing and forwarding the traffic to the back-end origin server after content transformation, see Load Balancing.

Configure secured ICAP protocol by using the GUI

1. Navigate to **Load Balancing** > **Services** and click **Add**.
2. In the **Services** page, enter the service details.
3. Navigate to **Load Balancing** > **Virtual Servers**. Add a virtual server of type HTTP/SSL. Or, you can select a virtual server and click **Edit**.
4. After entering the server basic details, click **Continue**.
5. In the **Advanced Settings** section, click **Policies**.
6. Go the **Policies** section and click the **Pencil** icon to configure the Content Inspection policy.
7. On the **Choose Policy** page, select **Content Inspection**. Click **Continue**.
8. In the **Policy Binding** section, click **+** to add a Content Inspection policy.
9. In the **Create ICAP Policy** page, enter a name for the policy.
10. In the **Action** field, click the "+" sign to add an ICAP action.
11. In the **Create ICAP Action** page, enter a name for the action.
12. Enter a name for the action.

13. In the **Server Name** field, enter the name of the TCP_SSL service already created.
14. In **ICAP Profile** field, click the "+" sign to add an ICAP profile.
15. In the **Create ICAP Profile** page, enter a profile name, URI, and MODE.
16. Click **Create**.
17. In the **Create ICAP Action** page, click **Create**.
18. In the **Create ICAP Policy** page, enter "true" in the **Expression Editor** and then click **Create**.
19. Click **Bind**.
20. When prompted to enable the Content Inspection feature, click **Yes**.
21. Click **Done**.

Audit log support for remote content inspection

If an incoming request or outgoing response is content inspected, the Citrix ADC appliance logs the ICAP details. The appliance stores the details as a log message in the ns.log file.

Each log message typically contains the following details:

```
<Source IP> <Destination IP> <Domain> <ICAP server IP><ICAP Mode> <
    Service URI> <ICAP response> <Policy action>
<!--NeedCopy-->
```

Example for content inspected request log message:

```
Apr 18 14:45:41 <local0.info> 10.106.97.104 04/18/2018:14:45:41 GMT 0-
    PPE-0 : default CI ICAP_LOG 788 0 : Source 10.102.1.98:39048 -
    Destination 10.106.97.89:8011 - Domain 10.106.97.89 - Content-Type
    application/x-www-form-urlencoded - ICAP Server 10.106.97.99:1344 -
    Mode REQMOD - Service /example - Response 204 - Action FORWARD
<!--NeedCopy-->
```

Example for content inspected response log message:

```
Apr 18 12:34:08 <local0.info> 10.106.97.104 04/18/2018:12:34:08 GMT 0-
    PPE-0 : default CI ICAP_LOG 71 0 : Source 10.106.97.105:18552 -
    Destination 10.106.97.99:80 - Domain NA - Content-Type NA - ICAP
    Server 10.106.97.99:1344 - Mode RESPMOD - Service /example -
    Response 400 - Action Internal Error
<!--NeedCopy-->
```

Inline device integration with Citrix ADC

September 14, 2021

Security devices such as Intrusion Prevention System (IPS) and Next Generation Firewall (NGFW) protect servers from network attacks. These devices are deployed in layer 2 inline mode and their primary function is to protect servers from network attacks and report security threats on the network.

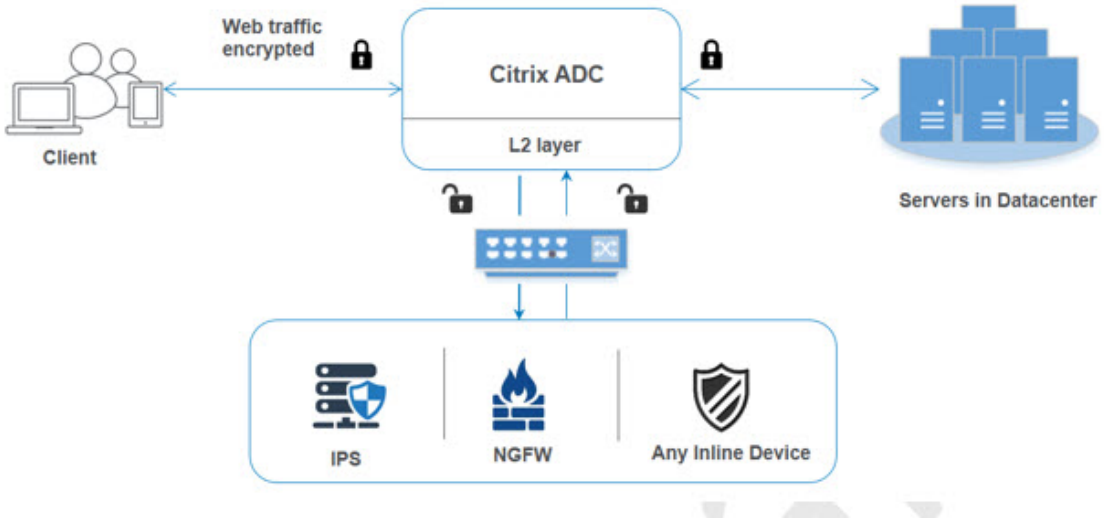
To prevent vulnerable threats and provide advanced security protection, a Citrix ADC appliance is integrated with one or more inline devices. The inline devices can be any security device such as IPS, NGFW.

Following are some of the use cases that benefit in using the inline device integration with the Citrix ADC appliance:

- **Inspecting encrypted traffic**. Most IPS and NGFW appliances bypass encrypted traffic, thereby leaving servers vulnerable to attacks. A Citrix ADC appliance can decrypt traffic and send it to inline devices for inspection. It enhances the customer's network security.
- **Offloading inline devices from TLS/SSL processing**. TLS/SSL processing is expensive and the issue can result in high system CPU in IPS or NGFW appliances if they decrypt the traffic. As encrypted traffic is growing at a fast pace, these systems fail to decrypt and inspect encrypted traffic. Citrix ADC helps in offloading inline devices from TLS/SSL processing. It results in the inline device supporting a high volume of traffic inspection.
- **Loading balancing inline devices**. The Citrix ADC appliance load balances multiple inline devices when there is a high volume of traffic.
- **Smart selection of traffic**. Every packet flowing into the appliance might be content inspected, for example download of text files. User can configure the Citrix ADC appliance to select specific traffic (for example .exe files) for inspection and send the traffic to inline devices for processing the data

How the Citrix ADC is integrated with inline devices

The following diagram shows how a Citrix ADC is integrated with inline security devices.



When you integrate inline devices with the Citrix ADC appliance, the component interacts as per the following:

1. A client sends a request to Citrix ADC appliance.
2. The appliance receives the request and sends it to an inline device based on policy evaluation.
   Note: If there are two or more inline devices, the appliance load balances the devices and sends the traffic.
   If the incoming traffic is an encrypted one, the appliance decrypts the data and sends it as a plain text to the inline device for content inspection.
3. The inline device inspects the data for threats and decides whether to drop, reset, or send the data back to the appliance.
4. If there are security threats, the device modifies the data and sends it to the appliance.
5. The Citrix ADC in turn re-encrypts the data and forwards the request to the back-end server.
6. The back-end server sends the response to the Citrix ADC appliance.
7. The appliance again decrypts the data and sends it to the inline device for inspection.
8. Appliance re-encrypts the data and sends the response to the client

Software licensing

To deploy the inline device integration, your Citrix ADC appliance must be provisioned with one of the following licenses:

1. ADC Premium
2. ADC Advanced
3. Telco Advanced

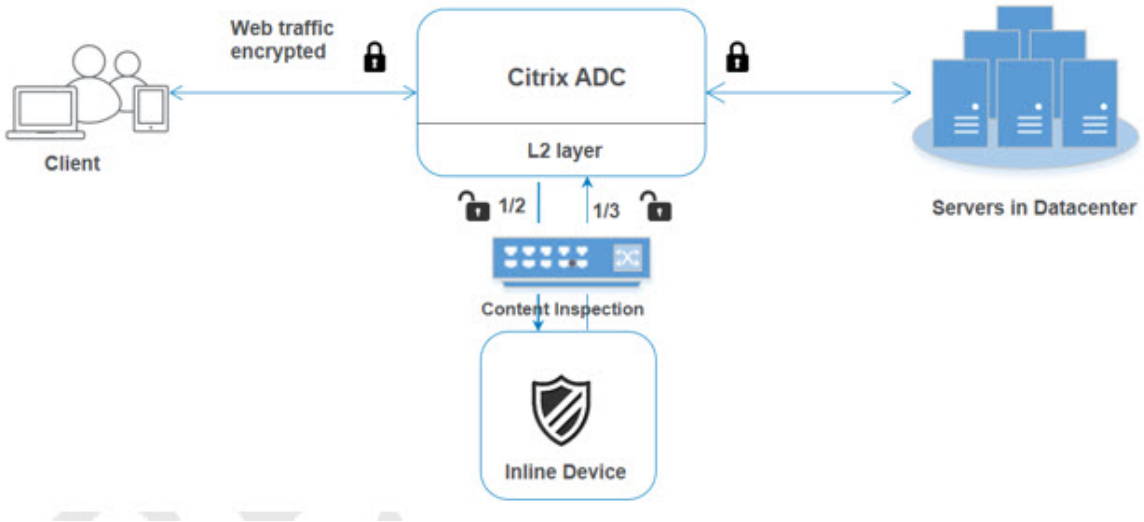
4. Telco Premium
5. SWG license

Configuring inline device integration

You can configure a Citrix ADC appliance with an inline device in three different ways. The configuration scenarios are as follows.

Scenario 1 for using a single inline device

If you want to integrate a security device (IPS or NGFW) in inline mode, you must begin by first enabling the Content Inspection feature and enabling the Citrix ADC in MBF (MAC-based forwarding) in global mode. Once you have enabled the features, you must add the Content Inspection profile, add the Content Inspection action for inline devices to reset, block, or drop the traffic based on inspection. Then, add the Content Inspection policy for the appliance to decide what subset of traffic to send to the inline devices. Then, configure the load balancing virtual server with layer 2 connection enabled on the server. Finally, bind the content inspection policy to the load balancing virtual server.



Enable MBF (MAC-based forwarding) mode

If you want the Citrix ADC appliance to be integrated to inline devices such as IPS, or firewalls, you must enable this mode. For more information about MBF, see Configure MAC-based Forwarding topic.

At the command prompt, type:

```
enable ns mode mbf
```


Enable Content Inspection

If you want the Citrix ADC appliance to decrypt and then send the content for inspection to the inline devices, you must enable the Content Inspection and load balancing features.

```
enable ns feature contentInspection LoadBalancing
```

Add Layer 2 connection method

To handle response generated by inline devices, the appliance uses the VLAN channel as a layer 2 method (L2ConnMethod) of communication with inline devices.

At the command prompt, type:

```
set l4param -l2ConnMethod <l2ConnMethod>
```

Example

```
set l4param -l2ConnMethod VlanChannel
```

Add Content Inspection profile for service

Inline device configuration for a Citrix ADC appliance can be specified in an entity called the Content Inspection profile. The profile has a collection of settings that explains how to integrate with an inline device.

At the command prompt, type:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface "1/2" -egressInterface "1/3"
```

Add IPS-TCP monitor

If you want to configure monitors, you add a user defined monitor.
Note: If you want to configure monitors, you must use a custom monitor. When adding a monitor, you must enable the transparent parameter.

At the command prompt, type:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr>] [-destPort
 <port>] [-transparent ( YES | NO )]
```

Example:


```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

Add a service

Add a service. Specify a dummy IP address that is not owned by any of the devices, including the inline devices. Set `use source IP address` (USIP) to YES. Set `useproxyport` to NO. By default, health monitoring is ON, bind the service to a health monitor, and also set the TRANSPARENT option in the monitor ON. At the command prompt, type:

```
add service <Service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor YES -usip ON -useproxyport OFF
```

Example:

```
add service ips_service 192.168.10.2 TCP * -healthMonitor YES -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof
```

Add a health monitor

By default the health monitor is turned on and you also have the option to disable it, if necessary. At the command prompt, type:

```
add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent <
YES, NO>
```

Example:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

Bind the service to the health monitor

After configuring the health monitor, you must bind the service to the health monitor. At the command prompt, type:

```
bind service <name> -monitorName <name>
```

Example:

```
bind service ips_svc -monitorName ips_tcp
```

Add content inspection action for service

After you enable the Content Inspection feature and then after you add the inline profile and service, you must add the Content Inspection action for handling the request. Based on the content inspection action, the inline device can drop, reset, or block action after it has inspected the data.

If the Inline server or service is down, you can configure the `ifserverdown` parameter in the appliance to perform any one of the following actions.

CONTINUE: If the User wants to bypass the content inspection when the remote server is down, you can choose the “CONTINUE” action, as default.
RESET (default): This action responds to the client by closing the connection with RST.
DROP: This action silently drops the packets without sending a response to the user.

At the command prompt, type:

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction
 <reqTimeoutAction>]

add ContentInspection action <action_name> -type InlineINSPECTION -serverName
 Service_name/Vserver_name>
```

Example:

```
add ContentInspection action <Inline_action> -type InlineSPECTION -serverName
 Inline_service1
```

Add content inspection policy for inspection

After you create a Content Inspection action, you must add Content Inspection policies to evaluate requests for inspection. The policy is based on a rule which consists of one or more expressions. The policy evaluates and selects the traffic for inspection based on the rule.

At the command prompt, type the following:

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

Example

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

Add content switching or load balancing virtual server of type HTTP/SSL

To receive the web traffic, you must add a load balancing virtual server. Also you must enable the layer2 connection on the virtual server.

At the command prompt, type:

```
add lb vserver <name> <vserver name> -l2Conn ON
```

Example:

```
add lb vserver HTTP_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```


Bind Content Inspection policy to content switching virtual server or load balancing virtual server of type HTTP/SSL

You bind the load balancing virtual server or content switching virtual server of type HTTP/SSL to the Content Inspection policy.
At the command prompt, type the following:

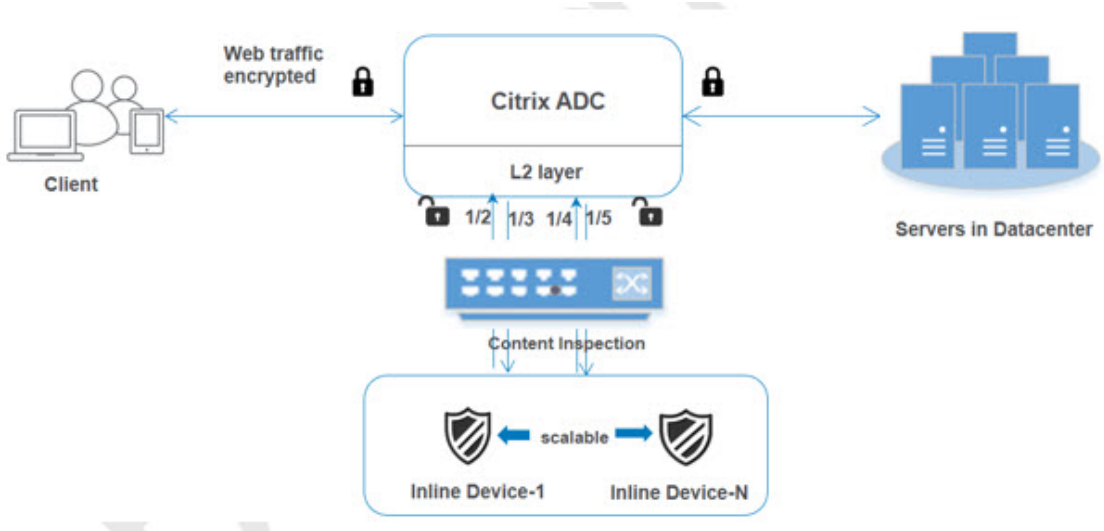
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

Example:

```
bind lb vserver HTTP_vserver -policyName Inline_pol1 -priority 100 -type
REQUEST
```

Scenario 2: Load balancing multiple inline devices using dedicated interfaces

If you are using two or more inline devices, you must load balance the devices using different content inspection services in a dedicated VLAN setup. In this case, the Citrix ADC appliance load balances the devices on top of sending a subset of traffic to each device through a dedicated interface.
For basic configuration steps, refer to scenario 1.



Add content inspection profile1 for service1

Inline configurations for a Citrix ADC appliance can be specified in an entity called the Content Inspection profile. The profile has a collection of device settings. The Content Inspection profile1 is created for inline service 1 and the communication is through 1/2 and 1/3 dedicated interfaces.

At the command prompt, type:


```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface "1/2" -egressInterface "1/3"
```

Add content inspection profile2 for service2

The Content Inspection profile2 is added for service2 and the inline device communicates with the appliance through 1/4 and 1/5 dedicated interfaces.
At the command prompt, type:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile Inline_profile2 -type InlineInspection -
ingressinterface "1/4" -egressInterface "1/5"
```

Add service 1 for inline device 1

After you enable the Content Inspection feature and add the inline profile, you must add an inline service 1 for inline device 1 to be part of the load balancing setup. The service that you add, provides all the inline configuration details.

At the command prompt, type:

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
<Inline_Profile_1> -healthmonitor OFF -usip ON -useproxyport OFF
```

Example:

```
add service Inline_service1 10.102.29.200 TCP 80 -contentInspectionProfileName
 Inline_profile1 -healthmonitor OFF -usip ON -useproxyport OFF
```

Add service 2 for inline device 2

After you enable the Content Inspection feature and add the inline profile, you must add an inline service 2 for inline device 2. The service that you add, provides all the inline configuration details.

At the command prompt, type:


```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
<Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

Example:

```
add service Inline_service1 10.29.20.205 TCP 80 -contentInspectionProfileName
 Inline_profile2 -healthmonitor OFF -usip ON -useproxyport OFF
```

Add load balancing virtual server

After you have added the inline profile and the services, you must add a load balancing virtual server for load balancing the services.

At the command prompt, type:

```
add lb vserver <vserver_name> TCP <Pvt_IP3> <port>
```

Example:

```
add lb vserver lb-Inline_vserver TCP *
```

Bind service 1 to the load balancing virtual server

After you add the load balancing virtual server, now bind the load balancing virtual server to the first service.

At the command prompt, type:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

Example:

```
bind lb vserver lb-Inline_vserver Inline_service1
```

Bind service 2 to the load balancing virtual server

After you add the load balancing virtual server, now bind the server to the second service.

At the command prompt, type:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

Example:

```
bind lb vserver lb-Inline_vserver Inline_service2
```


Add content inspection action for the service

After you enable the Content Inspection feature, you must add the Content Inspection action for handling the inline request information. Based on the action selected, the inline device drops, resets, or blocks after it has examined the given subset of traffic.

At the command prompt, type:

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction
 <reqTimeoutAction>]
```

```
add ContentInspection action < action_name > -type InlineINSPECTION -
serverName Service_name/Vserver_name>
```

Example:

```
add ContentInspection action Inline_action -type InlineINSPECTION -serverName
 lb-Inline_vserver
```

Add content inspection policy for inspection

After you create a Content Inspection action, you must add the Content Inspection policy to evaluate requests for service. The policy is based on a rule which consists of one or more expressions. The rule is associated to the Content Inspection action that is associated if a request matches the rule.

At the command prompt, type the following:

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

Example:

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

Add content switching or load balancing virtual server of type HTTP/SSL

Add a content switching or load balancing virtual server to accept web traffic. Also you must enable the layer2 connection on the virtual server.
For more information about load balancing, see How load balancing works topic.

At the command prompt, type:

```
add lb vserver <name> <vserver name> -l2Conn ON
```

Example:

```
add lb vserver http_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```


Bind Content Inspection policy to load balancing virtual server of type HTTP/SSL

You must bind the content switching or load balancing virtual server of type HTTP/SSL to the Content Inspection policy.

At the command prompt, type the following:

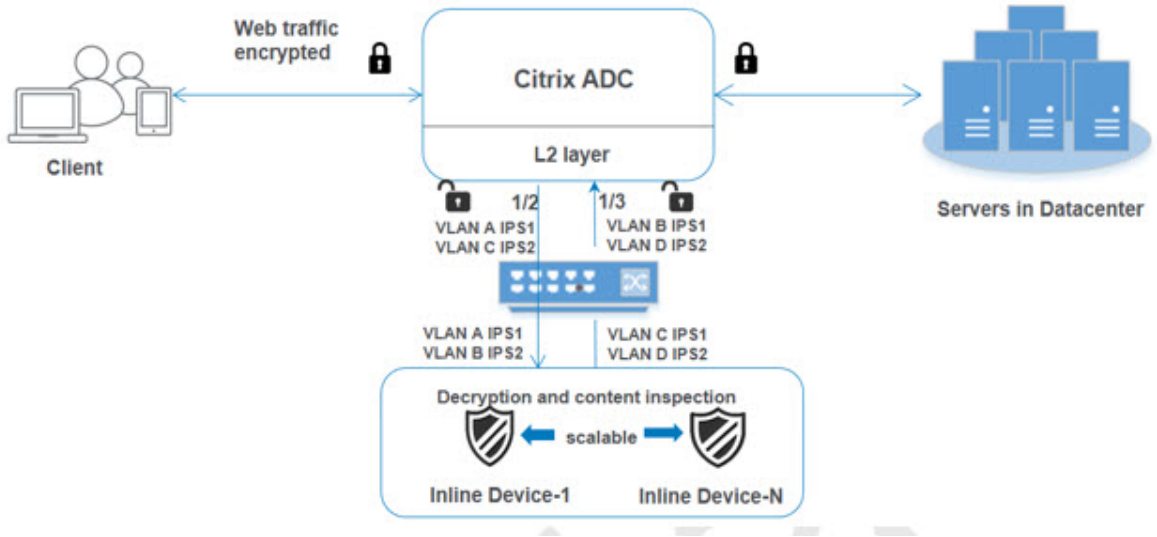
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <L7InlineREQUEST | L4Inline-REQUEST>
```

Example:

```
bind lb vserver http_vserver -policyName Inline_pol1 -priority 100 -type
REQUEST
```

Scenario 3: Load balancing multiple inline devices using shared interfaces

You can refer to this configuration, if you are using multiple inline devices and if you want to load balance the devices using different services in a shared VLAN interface. This configuration using shared VLAN interfaces is similar to use case 2. For basic configuration, refer to scenario 2.



Bind VLAN A with sharing option enabled

At the command prompt, type the following:

```
bind vlan <id> -ifnum <interface> -tagged
```

Example:

```
bind vlan 100 -ifnum 1/2 tagged
```


Bind VLAN B with sharing option enabled

At the command prompt, type the following:

```
bind vlan <id> -ifnum <interface> -tagged
```

Example:

```
bind vlan 200 -ifnum 1/3 tagged
```

Bind VLAN C with sharing option enabled

At the command prompt, type the following:

```
bind vlan <id> -ifnum <interface> -tagged
```

Example:

```
bind vlan 300 -ifnum 1/2 tagged
```

Bind VLAN D with sharing option enabled

At the command prompt, type the following:

```
bind vlan <id> -ifnum <interface> -tagged
```

Example:

```
bind vlan 400 -ifnum 1/3 tagged
```

Add content inspection profile1 for service1

Inline configurations for a Citrix ADC appliance can be specified in an entity called the Content Inspection profile. The profile has a collection of device settings. The Content Inspection profile is created for inline service 1 and the communication is through 1/2 and 1/3 dedicated interfaces.

At the command prompt, type:

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3” -egressVlan 100 -ingressVlan
 300
```


Add content inspection profile2 for service2

The Content Inspection profile2 is added for service2 and the inline device communicates with the appliance through 1/2 and 1/3 dedicated interfaces.

At the command prompt, type:

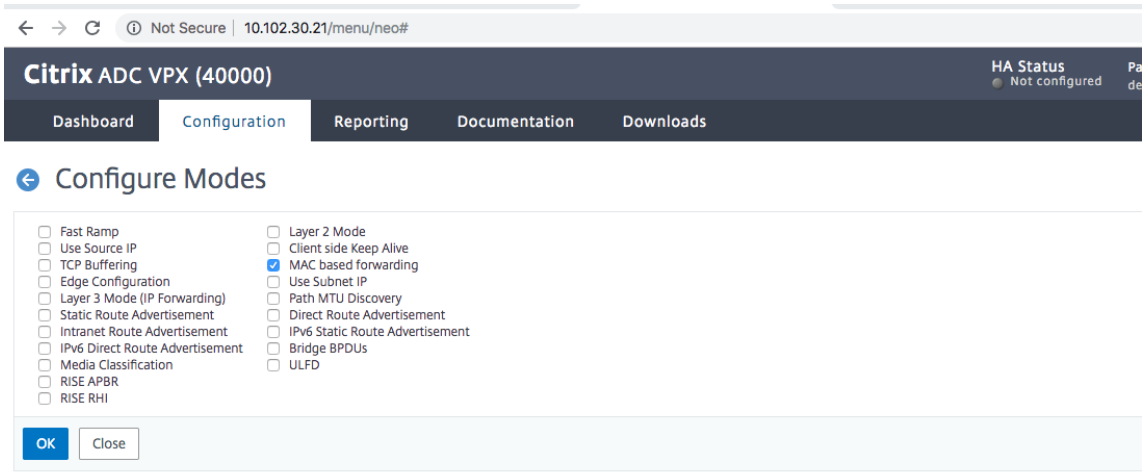
```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

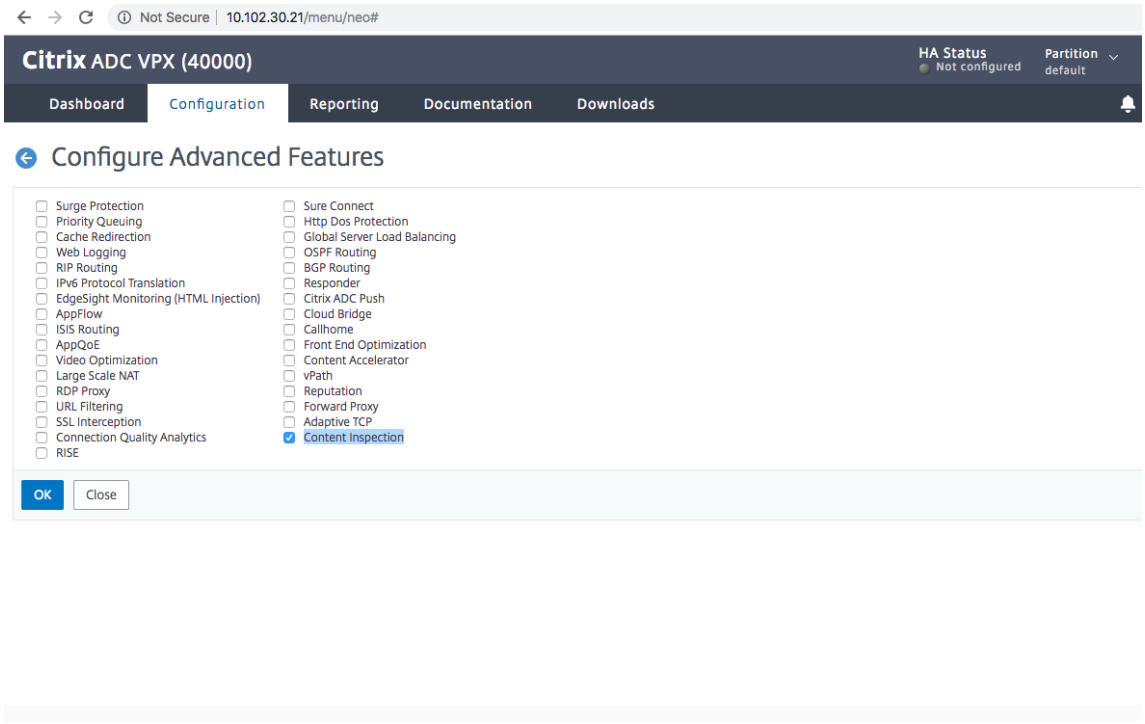
```
add contentInspection profile Inline_profile2 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3” -egressVlan 200 -ingressVlan
 400
```

Configure inline service integration using the Citrix ADC GUI

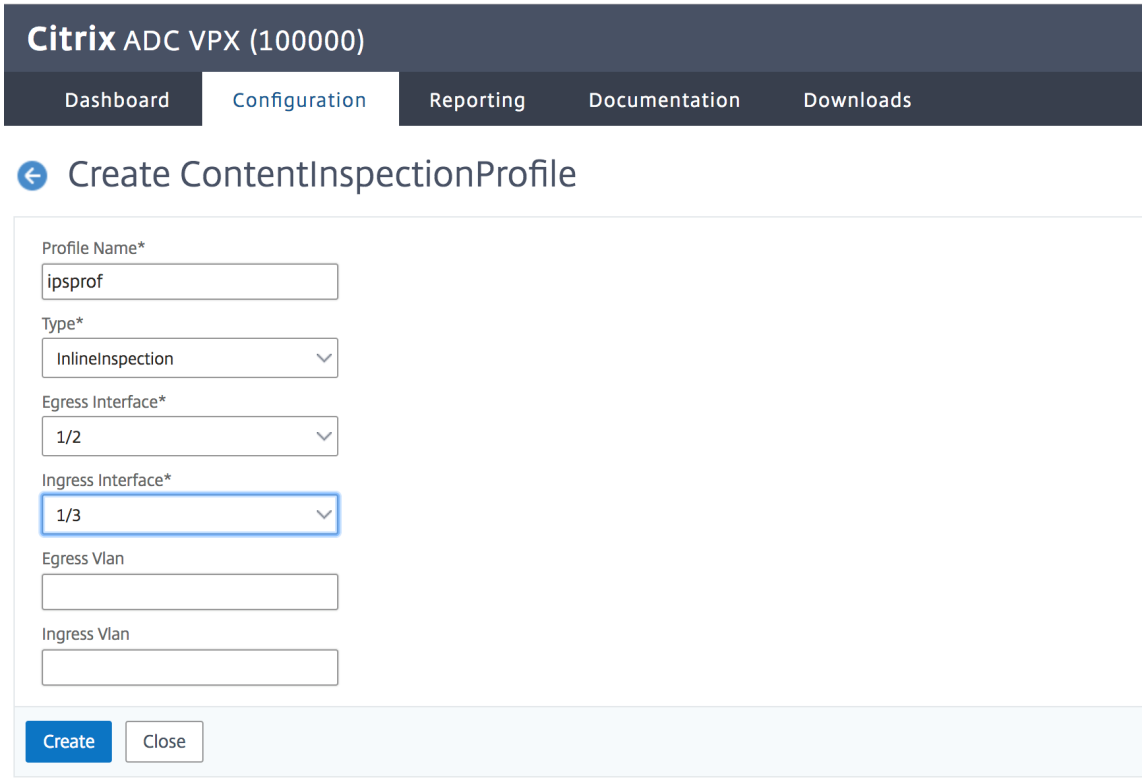
1. Log on to the Citrix ADC appliance and navigate to **Configuration** tab page.
2. Navigate to **System** > **Settings** > **Configure Modes**.
3. In the **Configure Modes** page, select **Mac Based Forwarding**.
4. Click **OK** and **Close**.



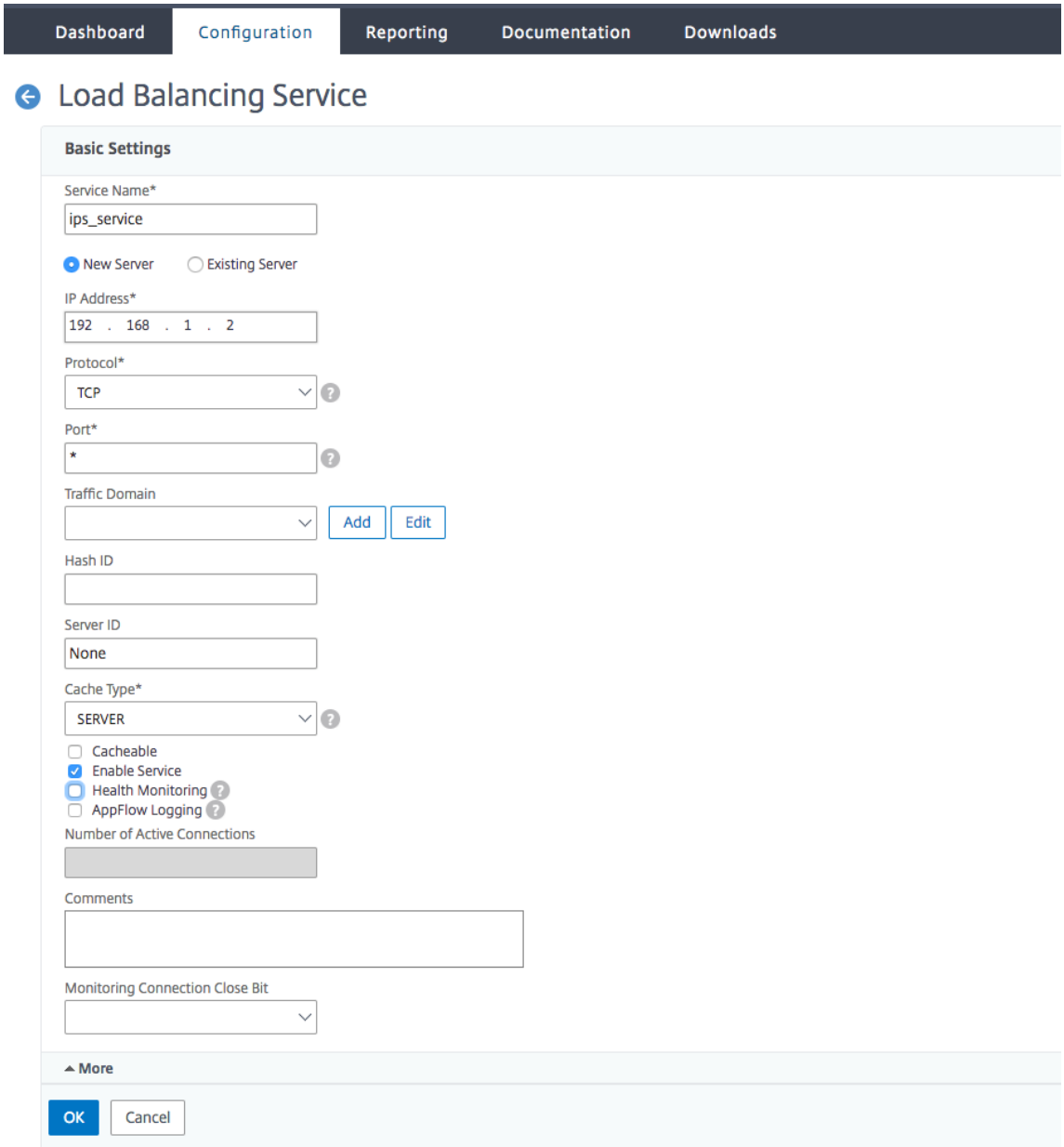
5. Navigate to **System** > **Settings** > **Configure Advanced Features**.
6. In the **Configure Advanced Feature** page, select **Content Inspection**.
7. Click **OK** and **Close**.



8. Navigate to **Security** > **Content Inspection** > **ContentInspection Profiles**.
9. In the **ContentInspection Profiles** page, click **Add**.
10. In the **Create ContentInspection Profiles** page, set the following parameters.
    a) Profile Name. Name of the content inspection profile.
    b) Type. Select the profile type as inlineInspection.
    c) Egress Interface. Interface through which the appliance sends traffic from the Citrix ADC to the Inline device.
    d) Ingress Interface. Interface through which the appliance receives traffic from the Inline device to the Citrix ADC.
    e) Egress VLAN. Interface VLAN ID through which the traffic is sent to the Inline device.
    f) Ingress VLAN. Interface VLAN ID through which the appliance receives traffic from Inline to Citrix ADC (if it is configured).

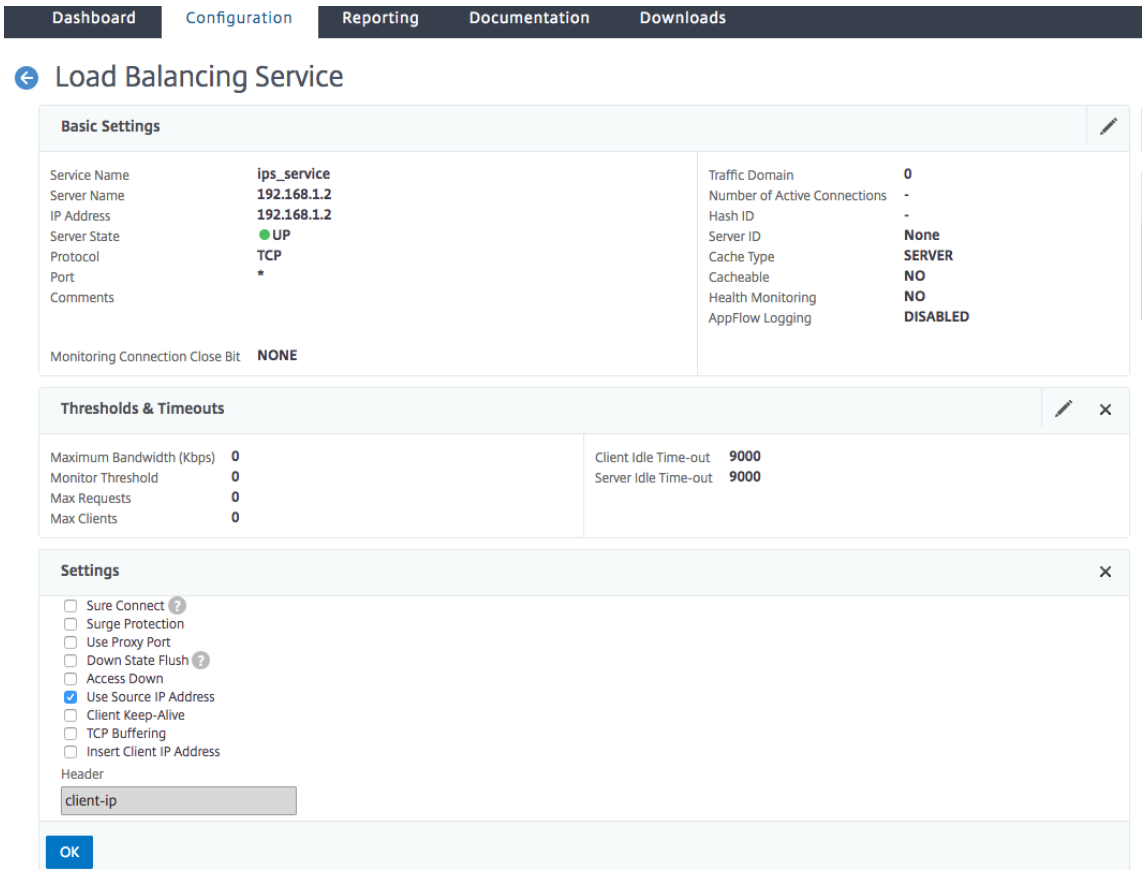


11. Click **Create** and **Close**.
12. Navigate to **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
13. In the **Services** page, set the following parameters:
    a) Service name. Name of the load balancing service.
    b) IP address. Use a dummy IP address. Note: No device must own the IP address.
    c) Protocol. Select protocol type as TCP.
    d) Port. Enter *
    e) Health Monitoring. Clear this option and enable it only if you want to bind the service to the TCP type monitor. If you want to bind a monitor to service then the TRANSPARENT option in the monitor must be ON. See step 14 on how to add monitor and how to bind it to service.
    f) Click **OK**.

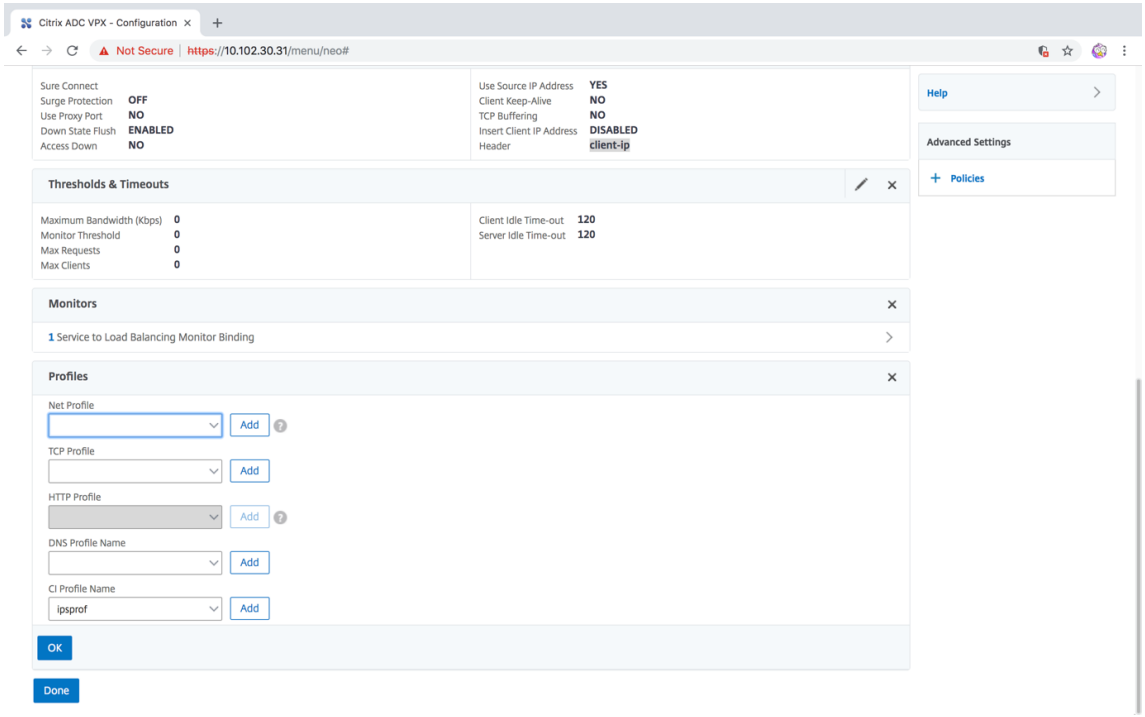


14. In the **Settings** section, edit the following and click **OK**.

    a) Use Proxy Port: Turn it OFF
    b) Use Source IP Address: Turn it ON



15. In the **Advanced Settings** section, click **Profiles**.
16. Go to **Profiles** section, and add the inline content inspection profile and click **OK**.

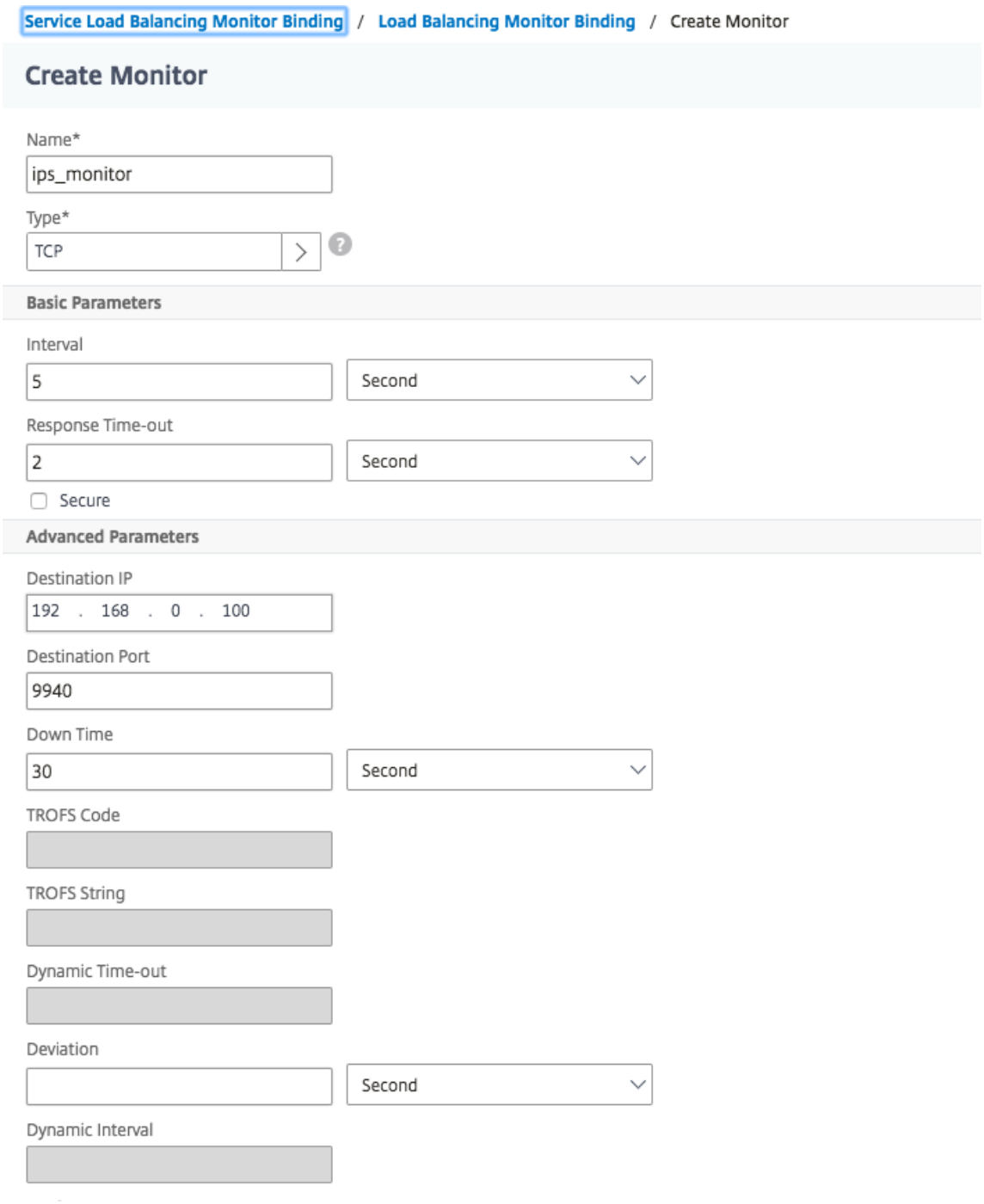


17. Go to **Monitors** section, **Add Bindings** > **Select Monitor** > **Add**.

    a) Name: Name of monitor
    b) Type: Select TCP type
    c) Destination IP, PORT: Destination IP address and Port.
    d) Transparent: Turn ON

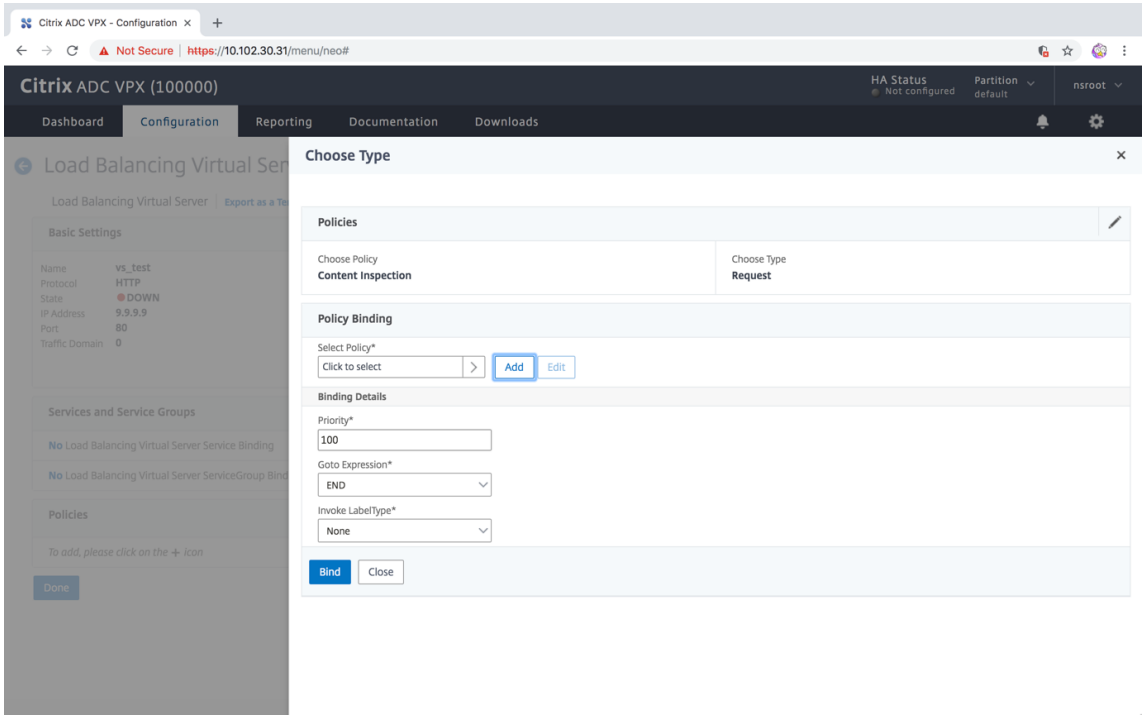
    Note: Monitor packets must flow through the inline device to monitor inline device status.

18. Click **Create**.

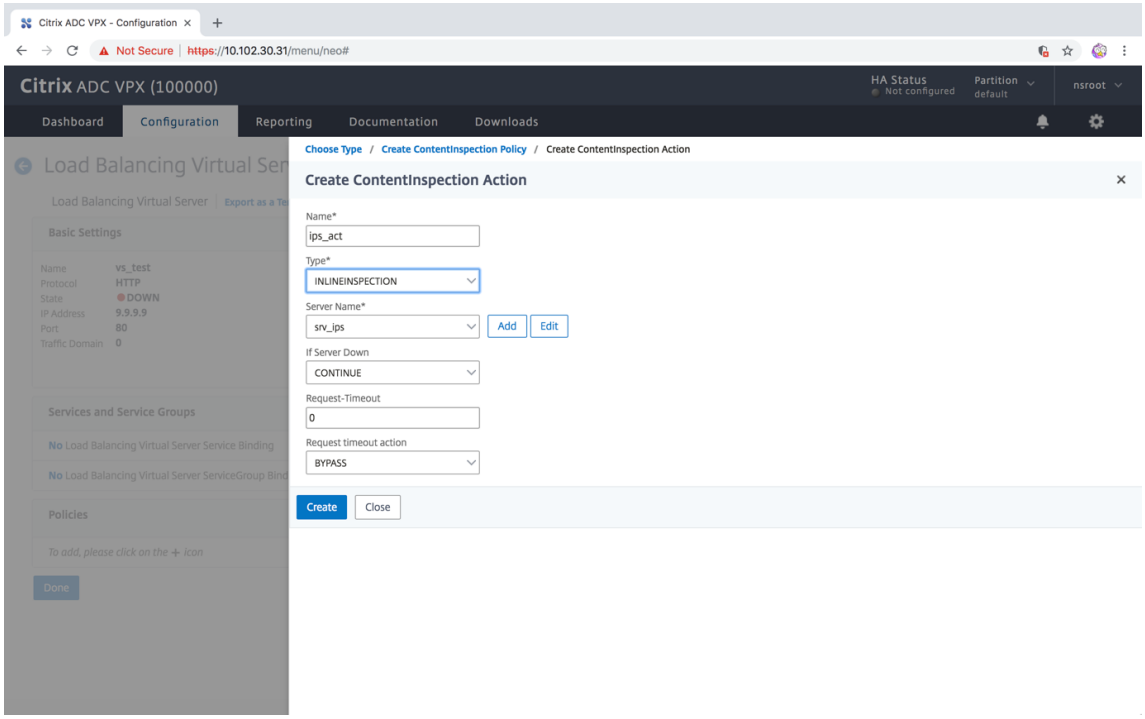


19. Click **Done**.
20. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**. Add a virtual server of type HTTP or SSL.
21. After entering the server details, click **OK** and again **OK**.
22. In the **Traffic Settings** section of Load Balancing Virtual Server, turn Layer 2 Parameters ON.
23. In the **Advanced Settings** section, click **Policies**.

24. Go the **Policies** section and click the “+” icon to configure the content inspection policy.
25. On the **Choose Policy** page, select Content Inspection. Click **Continue**.
26. In the **Policy Binding** section, click **Add** to add a Content Inspection policy.



27. In the **Create ContentInspection Policy** page, enter a name for the Inline content inspection policy.
28. In the **Action** field, click **Add** to create an Inline content inspection action.
29. In the **Create CI Action** page, set the following parameters:
    a) Name. Name of the content inspection Inline policy.
    b) Type. Select the type as inlineInspection.
    c) Server. Select the server/service as Inline devices.
    d) If Server Down. Select an operation if the server goes down.
    e) Request Time-out. Select a time-out value. You can use default values.
    f) Request Time-out Action. Select a time-out action. You can use default values.
30. Click **Create**.



31. Click **Create**.
32. In the **Create CI Policy** page, enter other details:
33. Click **OK** and **Close**.

Integration with IPS or NGFW as inline devices using SSL forward proxy

September 14, 2021

Security devices such as Intrusion Prevention System (IPS) and Next Generation Firewall (NGFW) protect servers from network attacks. These devices can inspect live traffic and are typically deployed in layer 2 inline mode. The SSL forward proxy appliance provides security of users and the enterprise network when accessing resources on the internet.

An SSL forward proxy appliance can be integrated with one or more inline devices to prevent threats and provide advanced security protection. The inline devices can be any security device, such as IPS and NGFW.

Some use cases where you can benefit by using the SSL forward proxy appliance and inline device integration are:

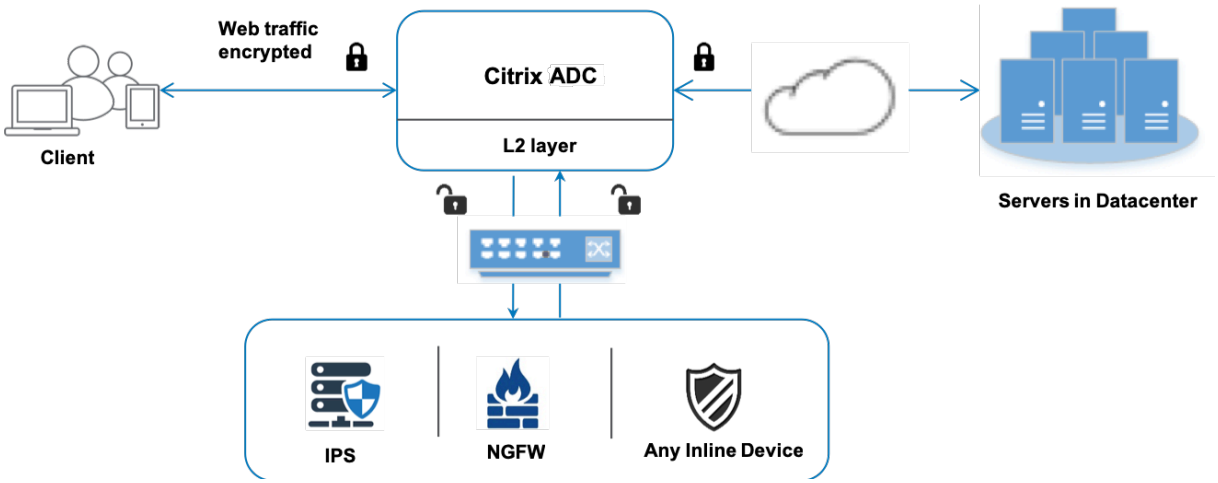
- **Inspecting encrypted traffic:** Most IPS and NGFW appliances bypass encrypted traffic, which can leave servers vulnerable to attacks. An SSL forward proxy appliance can decrypt traffic and

send it to the inline devices for inspection. This integration enhances the customer's network security.

- **Offloading inline devices from TLS/SSL processing:** TLS/SSL processing is expensive, which can result in high CPU utilization in IPS or NGFW appliances if they also decrypt the traffic. An SSL forward proxy appliance helps in offloading TLS/SSL processing from inline devices. As a result, inline devices can inspect a higher volume of traffic.
- **Loading balancing inline devices:** If you have configured multiple inline devices to manage heavy traffic, an SSL forward proxy appliance can load balance and distribute traffic evenly to these devices.
- **Smart selection of traffic:** Instead of sending all the traffic to the inline device for inspection, the appliance does a smart selection of traffic. For example, it skips sending text files for inspection to the inline devices.

SSL forward proxy integration with inline devices

The following diagram shows how an SSL forward proxy is integrated with inline security devices.



When you integrate inline devices with the SSL forward proxy appliance, the components interact as follows:

1. A client sends a request to an SSL forward proxy appliance.
2. The appliance sends the data to the inline device for content inspection based on the policy evaluation. For HTTPS traffic, the appliance decrypts the data and sends it in plain text to the inline device for content inspection.

> **Note**
>
> If there are two or more inline devices, the appliance load balances the devices and sends the traffic.

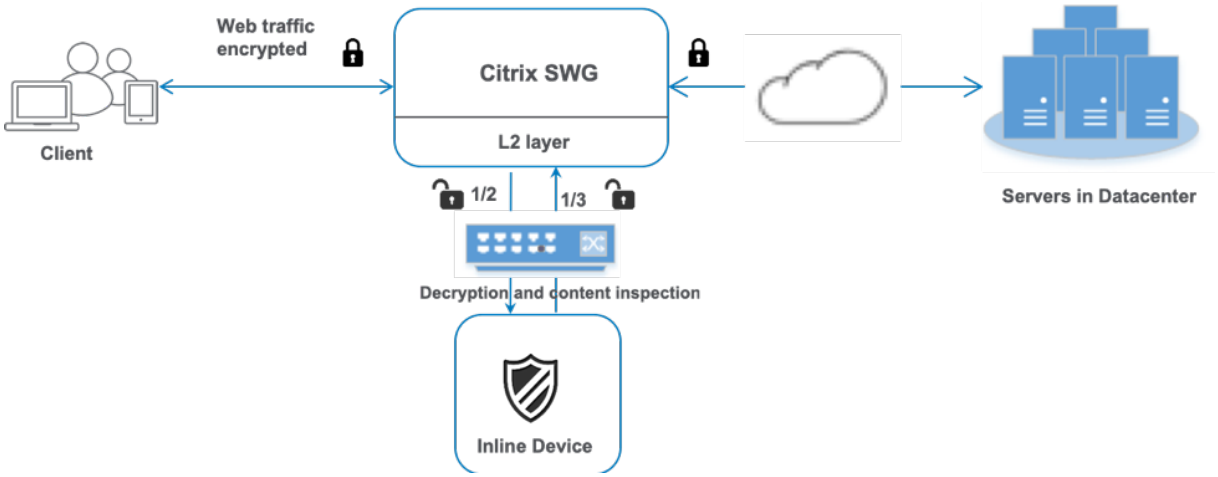
3. Add a content switching or an HTTP/HTTPS load balancing virtual server.
4. The inline device inspects the data for threats and decides whether to drop, reset, or send the data back to the appliance.
5. If there are security threats, the device modifies the data and sends it to the appliance.
6. For HTTPS traffic, the appliance re-encrypts the data and forwards the request to the back-end server.
7. The back-end server sends the response to the appliance.
8. The appliance again decrypts the data and sends it to the inline device for inspection.
9. The inline device inspects the data. If there are security threats, the device modifies the data and sends it to the appliance.
10. The appliance re-encrypts the data and sends the response to the client.

Configuring inline device integration

You can configure an SSL forward proxy appliance with an inline device in three different ways as follows:

Scenario 1: Using a single inline device

To integrate a security device (IPS or NGFW) in inline mode, you must enable content inspection and MAC-based forwarding (MBF) in global mode on the SSL forward proxy appliance. Then, add a content inspection profile, a TCP service, a content inspection action for inline devices to reset, block, or drop the traffic based on inspection. Also add a content inspection policy that the appliance uses to decide the subset of traffic to send to the inline devices. Finally, configure the proxy virtual server with layer 2 connection enabled on the server and bind the content inspection policy to this proxy virtual server.



Perform the following steps:

1. Enable MAC-based forwarding (MPF) mode.
2. Enable the content inspection feature.
3. Add a content inspection profile for the service. The content inspection profile contains the inline device settings that integrate the SSL forward proxy appliance with an inline device.
4. (Optional) Add a TCP monitor.

   > **Note:**
   >
   > Transparent devices do not have an IP address. Therefore, to perform health checks, you must explicitly bind a monitor.

5. Add a service. A service represents an inline device.
6. (Optional) Bind the service to the TCP monitor.
7. Add a content inspection action for the service.
8. Add a content inspection policy and specify the action.
9. Add an HTTP or HTTPS proxy (content switching) virtual server.
10. Bind the content inspection policy to the virtual server.

Configure using the CLI

Type the following commands at the command prompt. Examples are given after most commands.

1. Enable MBF.

```
enable ns mode mbf
```

1. Enable the feature.

```
enable ns feature contentInspection
```

1. Add a content inspection profile.

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile ipsprof -type InlineInspection -ingressinterface
 "1/2" -egressInterface "1/3"
```

1. Add a service. Specify a dummy IP address that is not owned by any of the devices, including the inline devices. Set `use source IP address` (USIP) to YES. Set `useproxyport` to NO. By default, health monitoring is ON, bind the service to a health monitor, and also set the TRANSPARENT option in the monitor ON.


```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor YES -usip YES -useproxyport NO
```

Example:

```
add service ips_service 198.51.100.2 TCP * -healthMonitor YES -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof
```

1. Add a health monitor. By default the health monitor is turned on and you also have the option to disable it, if necessary. At the command prompt, type:

   ```
   add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent
    <YES, NO>
   ```

Example:

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

1. Bind the service to the health monitor

After configuring the health monitor, you must bind the service to the health monitor. At the command prompt, type:

```
bind service <name> -monitorName <name>
```

Example:

```
bind service ips_svc -monitorName ips_tcp
```

1. Add a content inspection action.

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

Example:

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
ips_service
```

1. Add a content inspection policy.

```
add contentInspection policy <name> -rule <expression> -action <string>
```

Example:

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"CONNECT\")
"-action ips_action
```

1. Add a proxy virtual server.

```
add cs vserver <name> PROXY <IPAddress> <port> -cltTimeout <secs> -Listenpolicy
 <expression> -authn401 ( ON | OFF )-authnVsName <string> -l2Conn ON
```


> **Note:**
>
> Load balancing virtual servers of type HTTP/SSL are also supported.

Example:

```
add cs vserver transparentcs PROXY * * -cltTimeout 180 -Listenpolicy exp1 -
authn401 on -authnVsName swg-auth-vs-trans-http -l2Conn ON
```

1. Bind the policy to the virtual server.

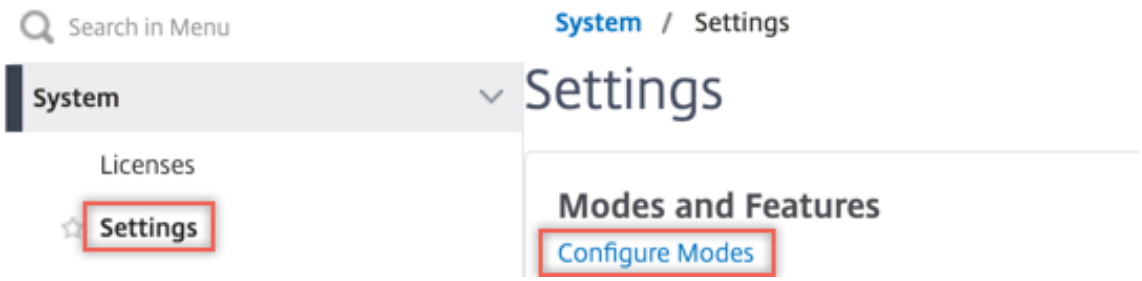
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

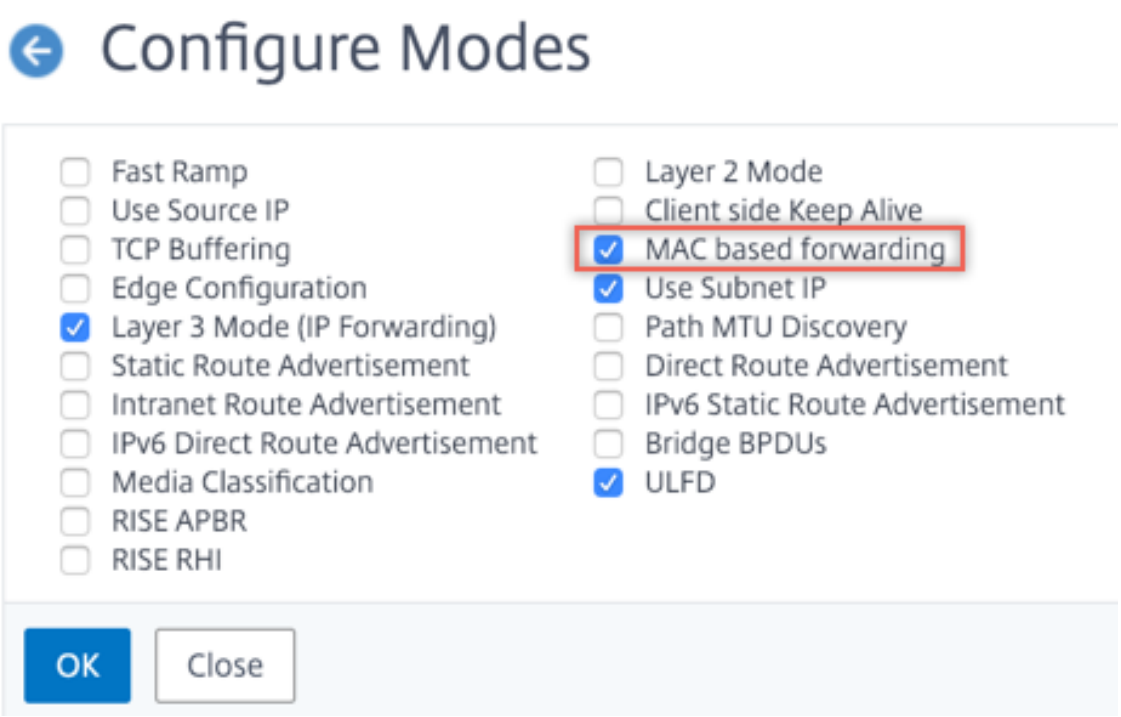
Example:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

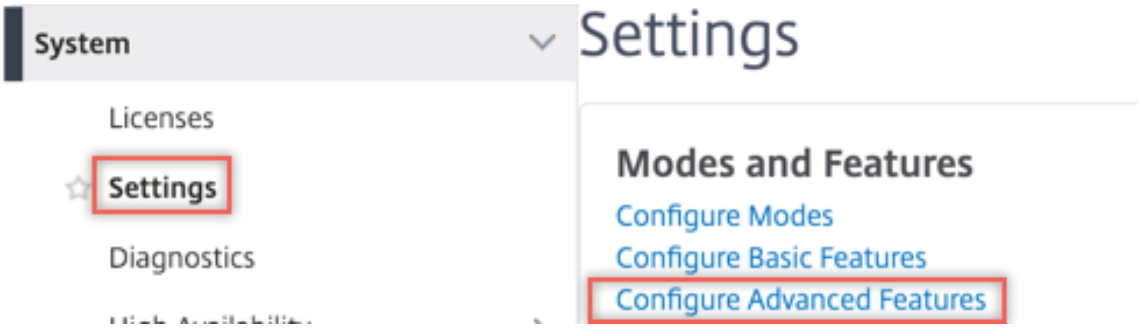
Configure using the GUI

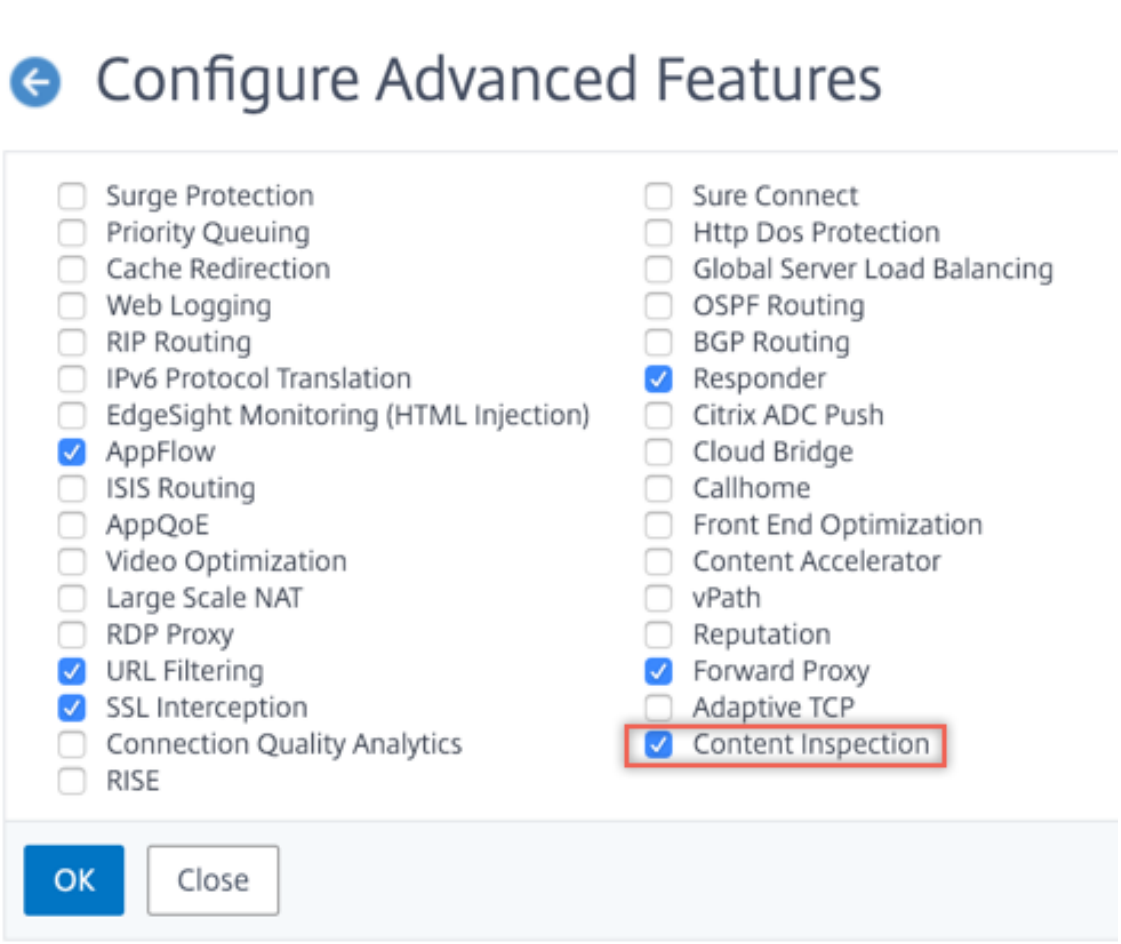
1. Navigate to **System > Settings**. In **Modes and Features**, click **Configure Modes**.



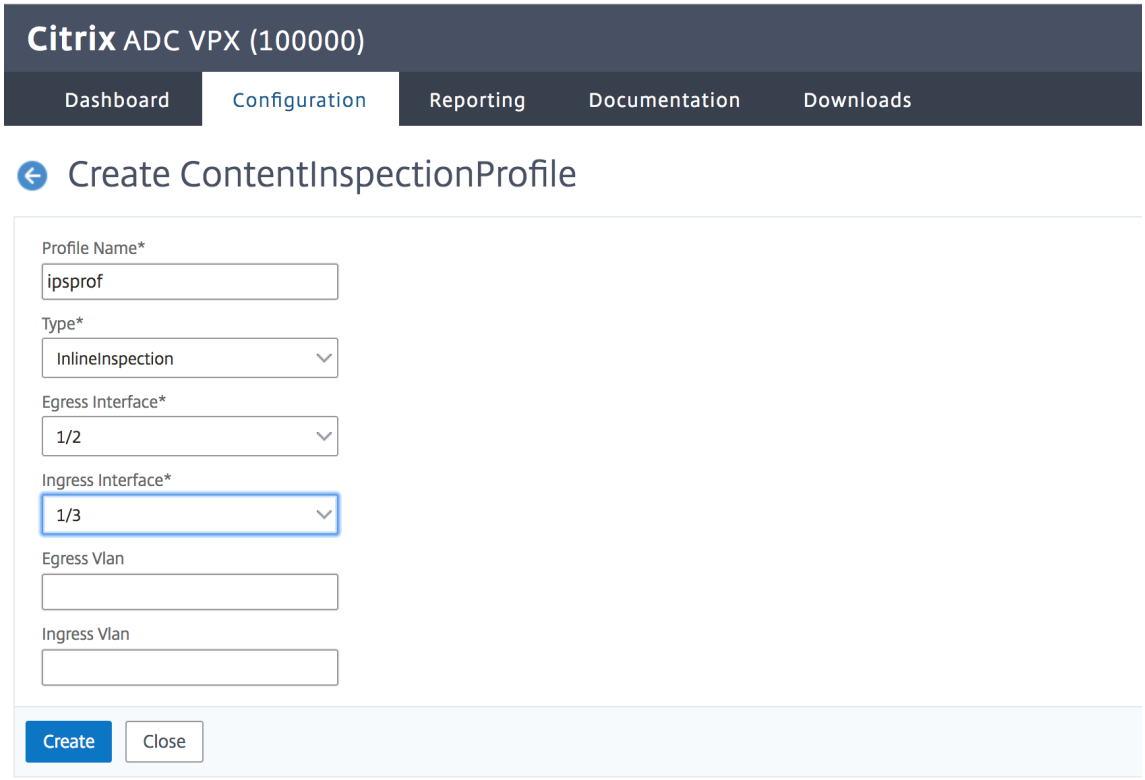


2. Navigate to **System > Settings**. In **Modes and Features**, click **Configure Advanced Features**.

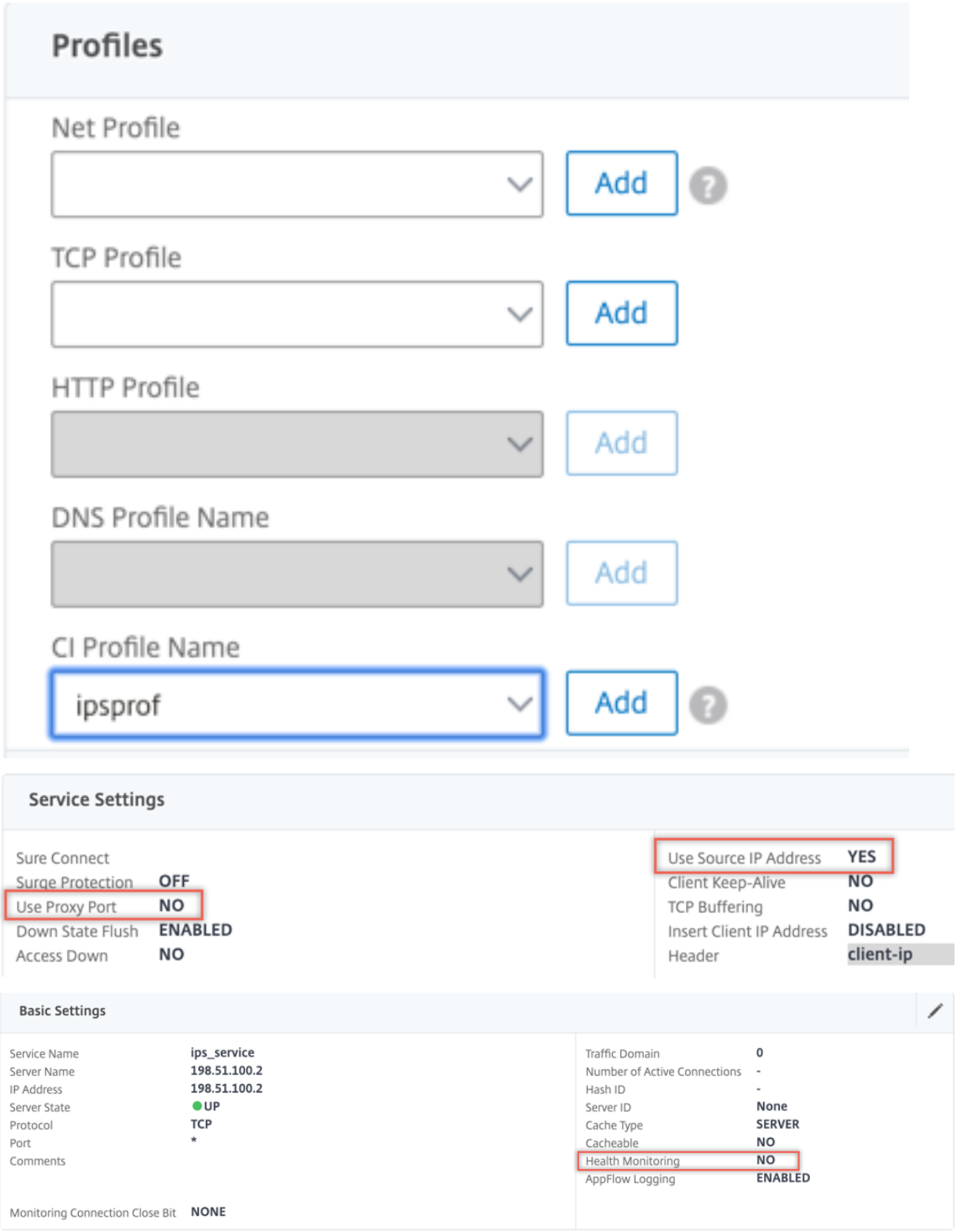




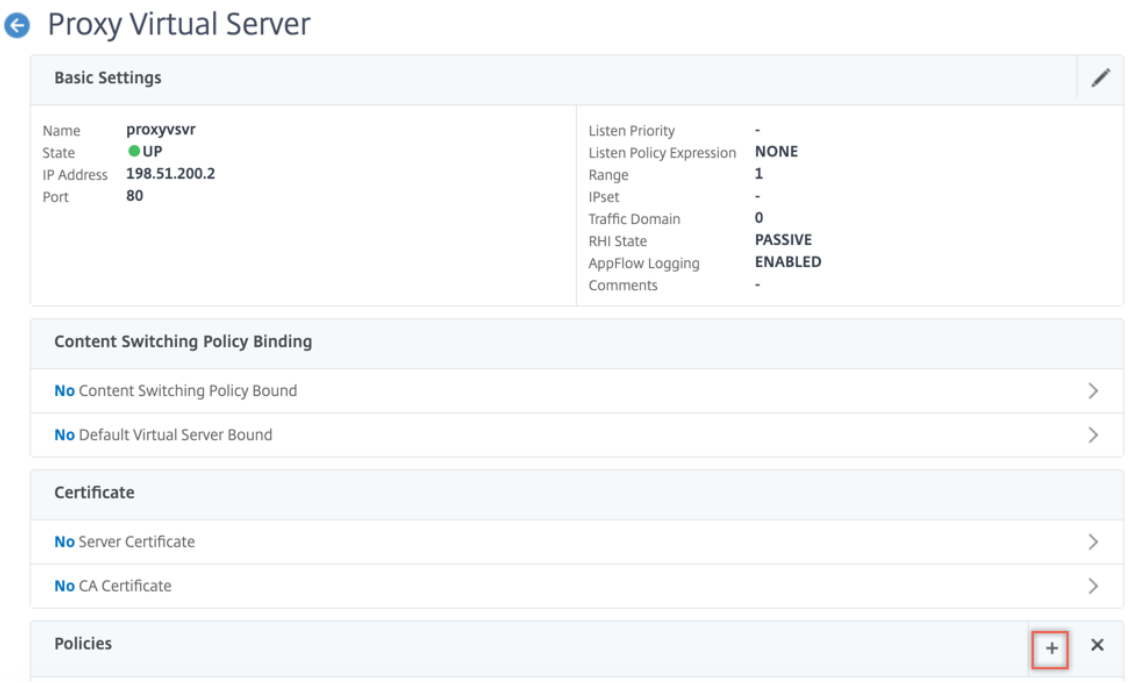
3. Navigate to **Secure Web Gateway > Content Inspection > Content Inspection Profiles**. Click **Add**.



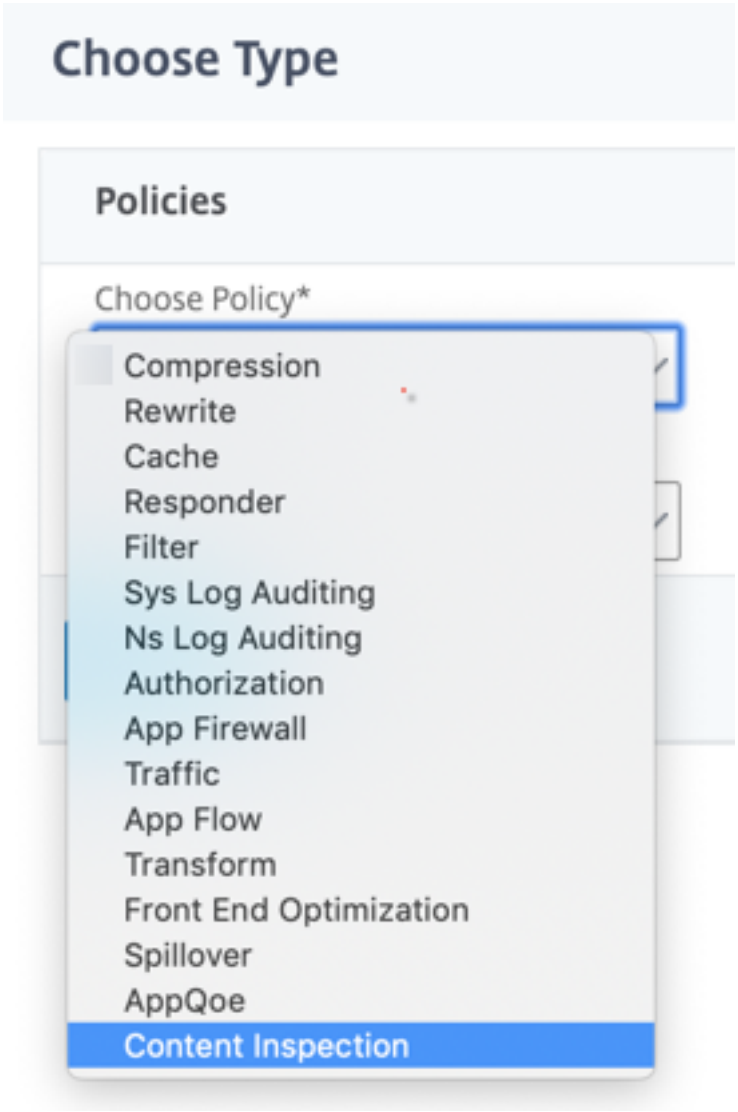
4. Navigate to **Load Balancing > Services > Add** and add a service. In **Advanced Settings**, click **Profiles**. In the **CI Profile Name** list, select the content inspection profile created earlier. In **Service Settings**, set **Use Source IP Address** to YES and **Use Proxy Port** to No. In **Basic Settings**, set **Health Monitoring** to NO. Turn on health monitoring only if you bind this service to a TCP monitor. If you bind a monitor to a service, then set the TRANSPARENT option in the monitor to ON.



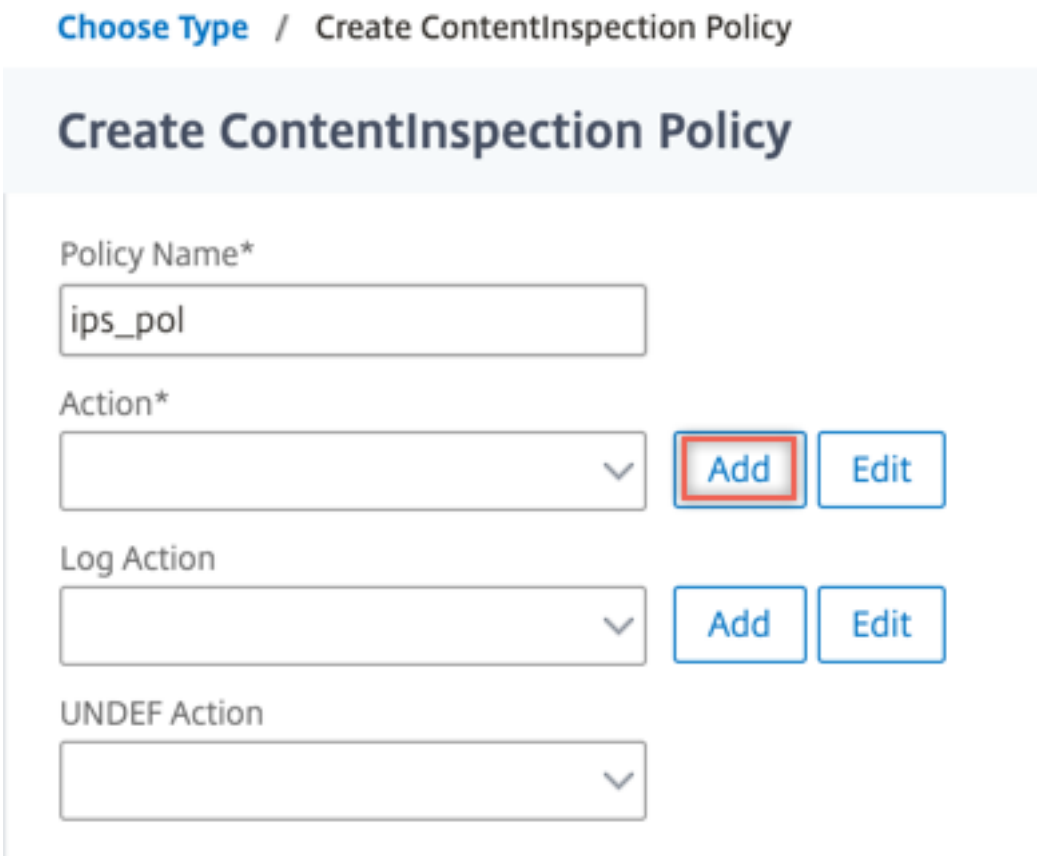
5. Navigate to **Secure Web Gateway > Proxy Virtual Servers> Add**. Specify a name, IP address, and port. In **Advanced Settings**, select **Policies**. Click the “+” sign.



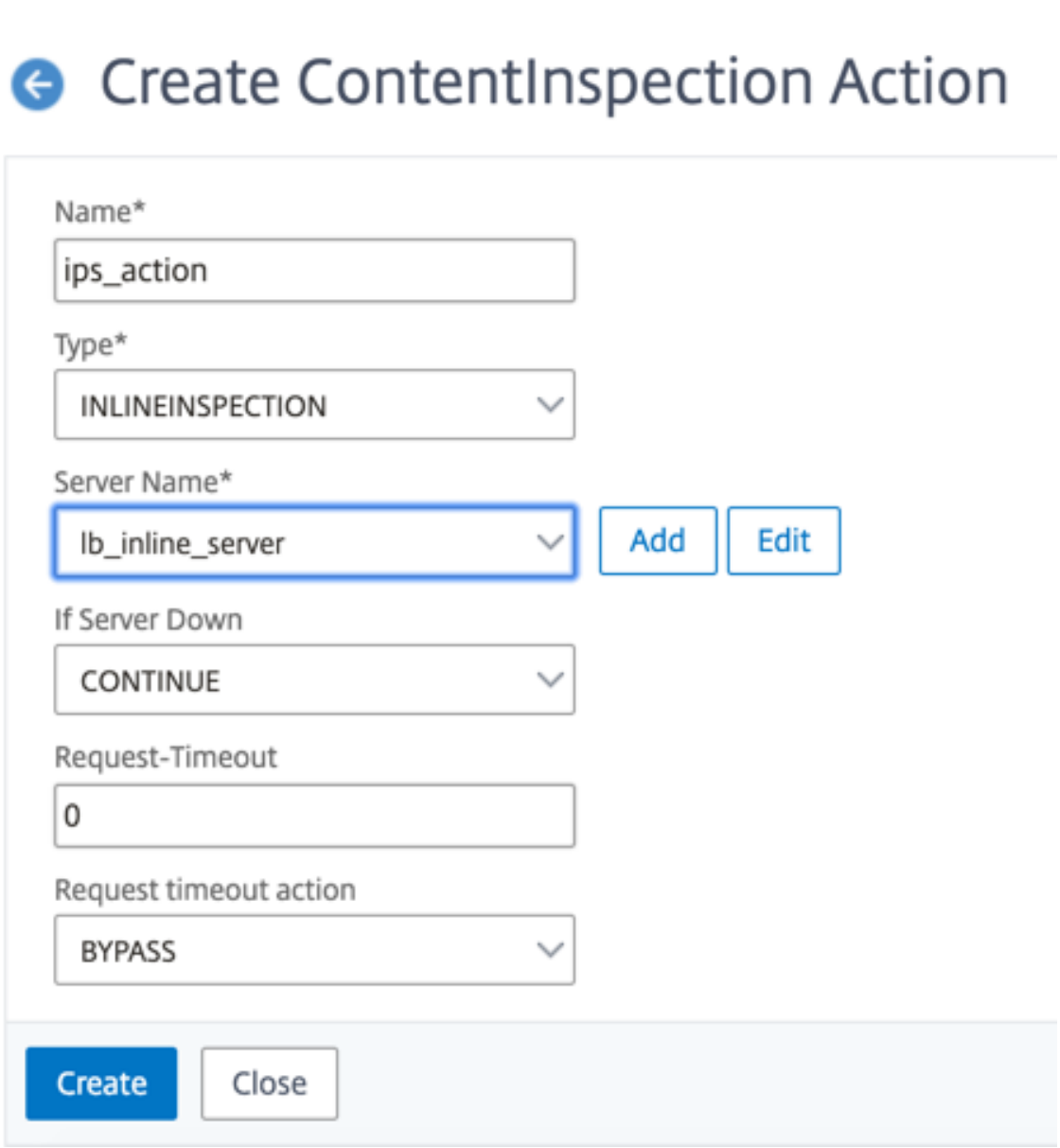
6. In **Choose Policy** select **Content Inspection**. Click **Continue**.



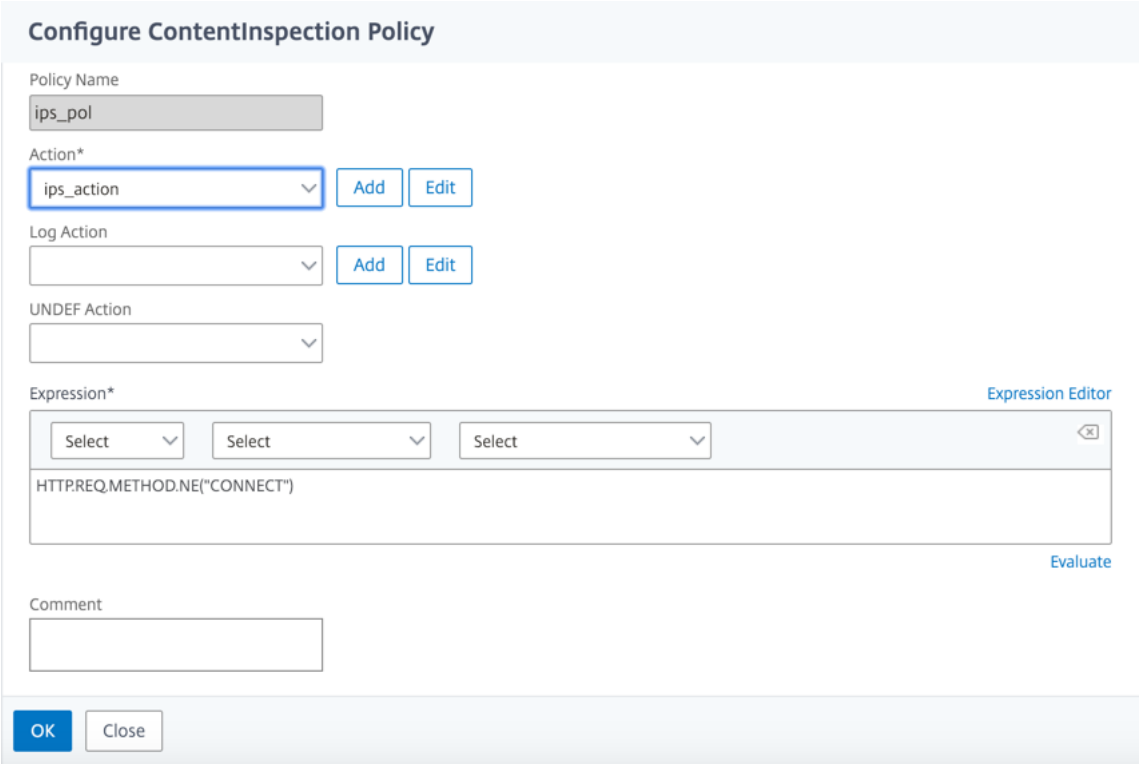
7. Click **Add**. Specify a name. In **Action**, click **Add**.



8. Specify a name. In **Type**, select **INLINEINSPECTION**. In **Server Name**, select the TCP service created earlier.



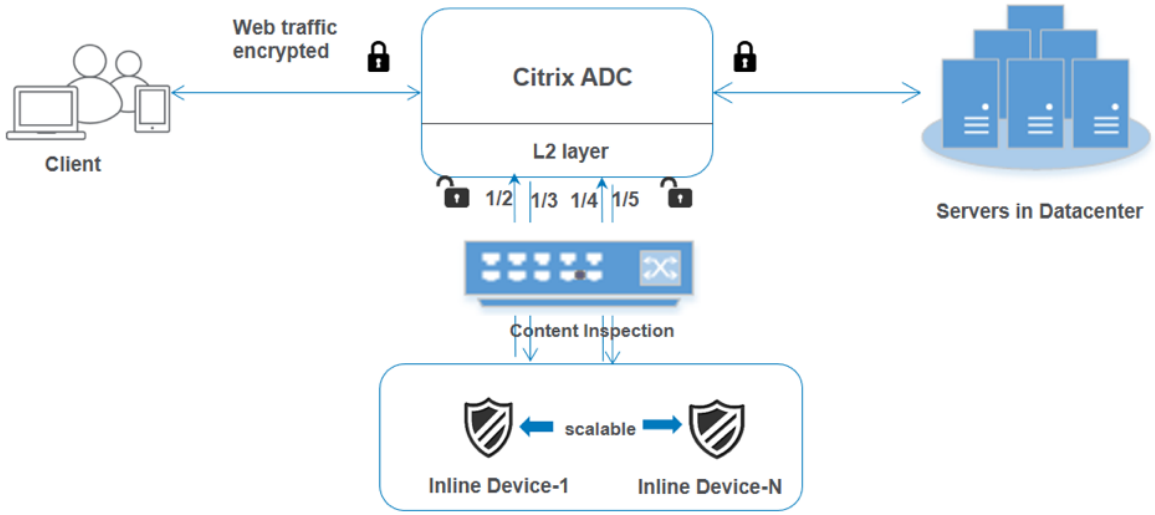
9. Click **Create**. Specify the rule and click **Create**.



10. Click **Bind**.
11. Click **Done**.

Scenario 2: Load balance multiple inline devices with dedicated interfaces

If you are using two or more inline devices, you can load balance the devices using different content inspection services with dedicated interfaces. In this case, the SSL forward proxy appliance load balances the subset of traffic sent to each device through a dedicated interface. The subset is decided based on the policies configured. For example, TXT or image files might not be sent for inspection to the inline devices.



The basic configuration remains the same as in scenario 1. However, you must create a content inspection profile for each inline device and specify the ingress and egress interface in each profile. Add a service for each inline device. Add a load balancing virtual server and specify it in the content inspection action. Perform the following extra steps:

1. Add content inspection profiles for each service.
2. Add a service for each device.
3. Add a load balancing virtual server.
4. Specify the load balancing virtual server in the content inspection action.

Configure using the CLI

Type the following commands at the command prompt. Examples are given after each command.

1. Enable MBF.

```
enable ns mode mbf
```

1. Enable the feature.

```
enable ns feature contentInspection
```

1. Add profile 1 for service 1.

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile ipsprof1 -type InlineInspection -ingressInterface
 "1/2"-egressInterface "1/3"
```


1. Add profile 2 for service 2.

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile ipsprof2 -type InlineInspection -ingressInterface
 "1/4"-egressInterface "1/5"
```

1. Add service 1. Specify a dummy IP address that is not owned by any of the devices, including the inline devices. Set `use source IP address` (USIP) to YES. Set `useproxyport` to NO. Turn on health monitoring with TCP monitor with TRANSPARENT option set ON.

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

Example:

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof1
```

1. Add service 2. Specify a dummy IP address that is not owned by any of the devices, including the inline devices. Set `use source IP address` (USIP) to YES. Set `useproxyport` to NO. Turn on health monitoring with TRANSPARENT option set ON.

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

Example:

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof2
```

1. Add a load balancing virtual server.

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

Example:

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. Bind the services to the load balancing virtual server.

```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

Example:

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```


1. Specify the load balancing virtual server in the content inspection action.

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

Example:

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
lb_inline_vserver
```

1. Add a content inspection policy. Specify the content inspection action in the policy.

```
add contentInspection policy <name> -rule <expression> -action <string>
```

Example:

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"CONNECT\")
"-action ips_action
```

1. Add a proxy virtual server.

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

Example:

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. Bind the content inspection policy to the virtual server.

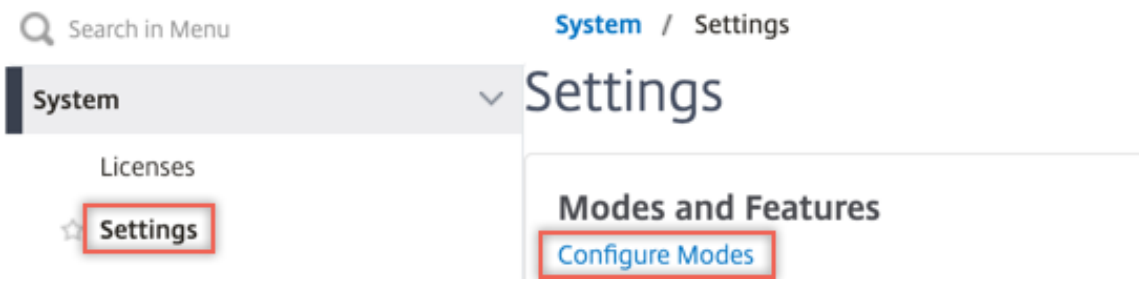
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

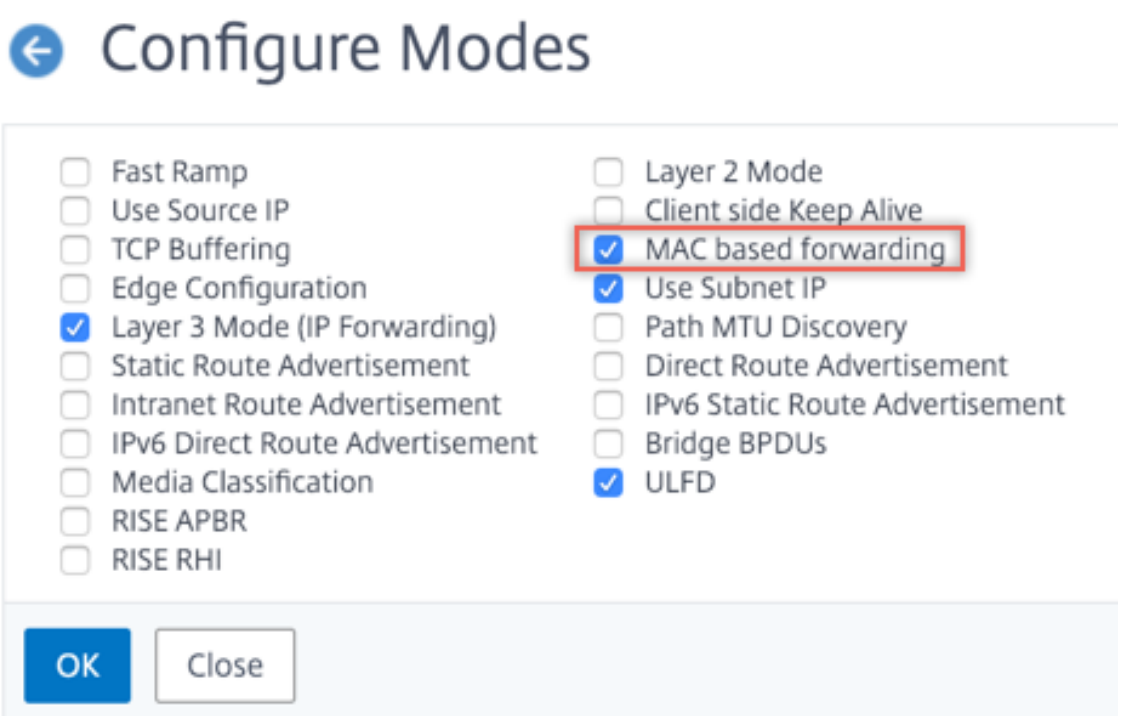
Example:

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

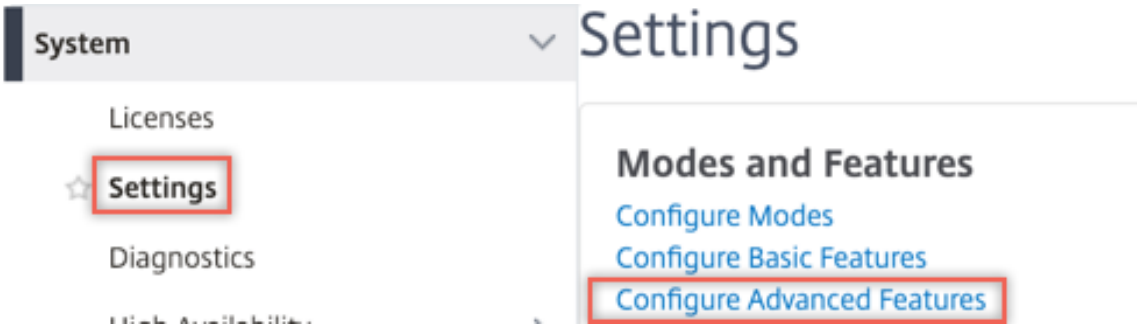
Configuration using the GUI

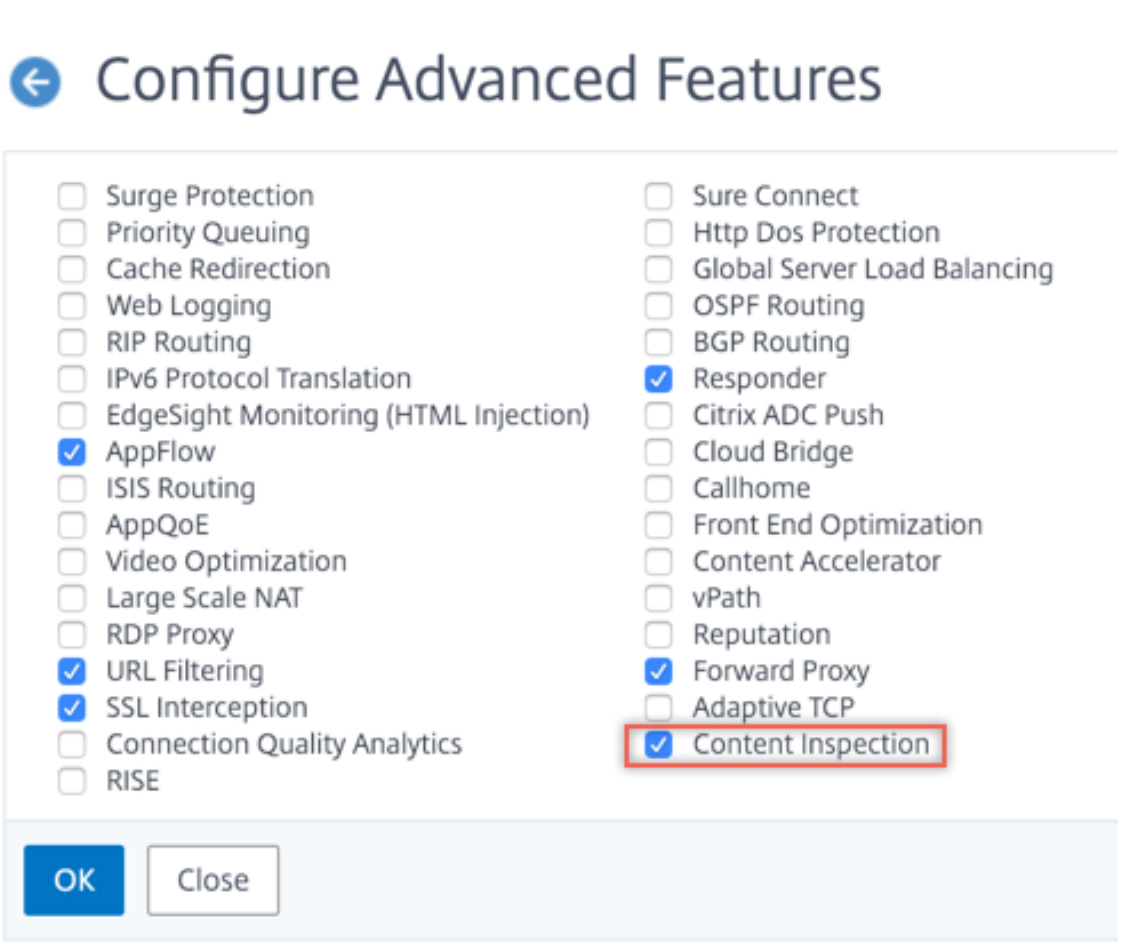
1. Navigate to **System > Settings**. In **Modes and Features**, click **Configure Modes**.



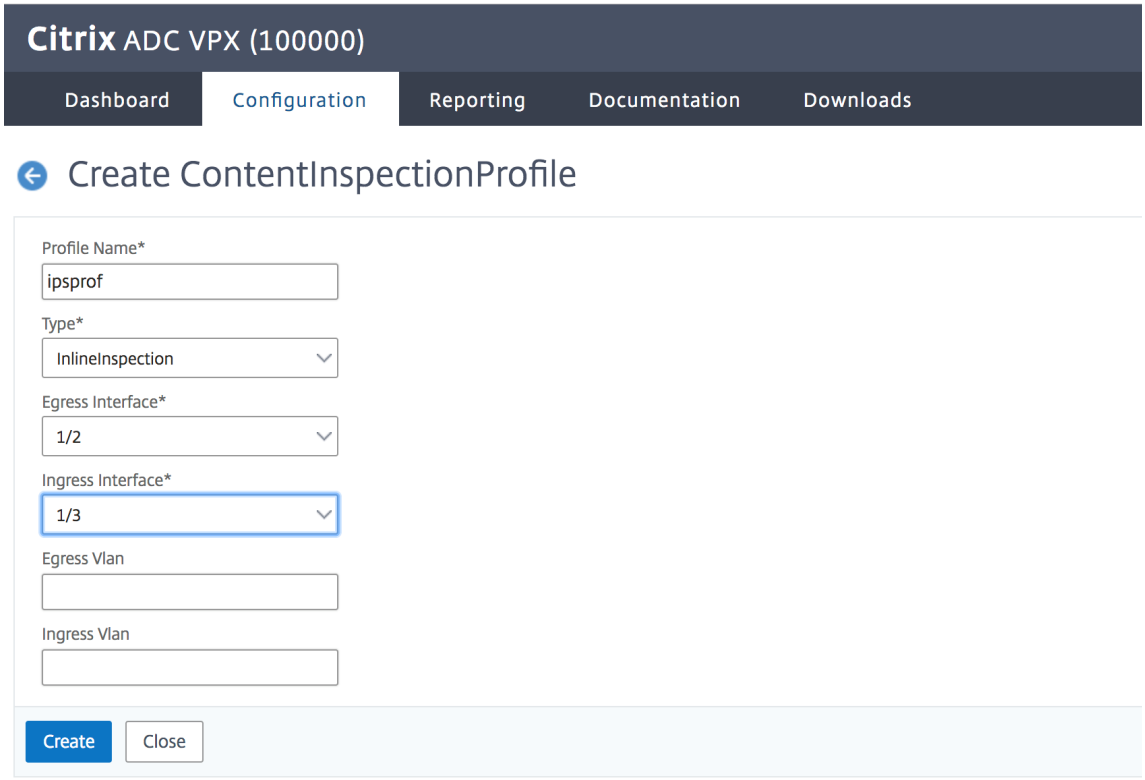


2. Navigate to **System > Settings**. In **Modes and Features**, click **Configure Advanced Features**.

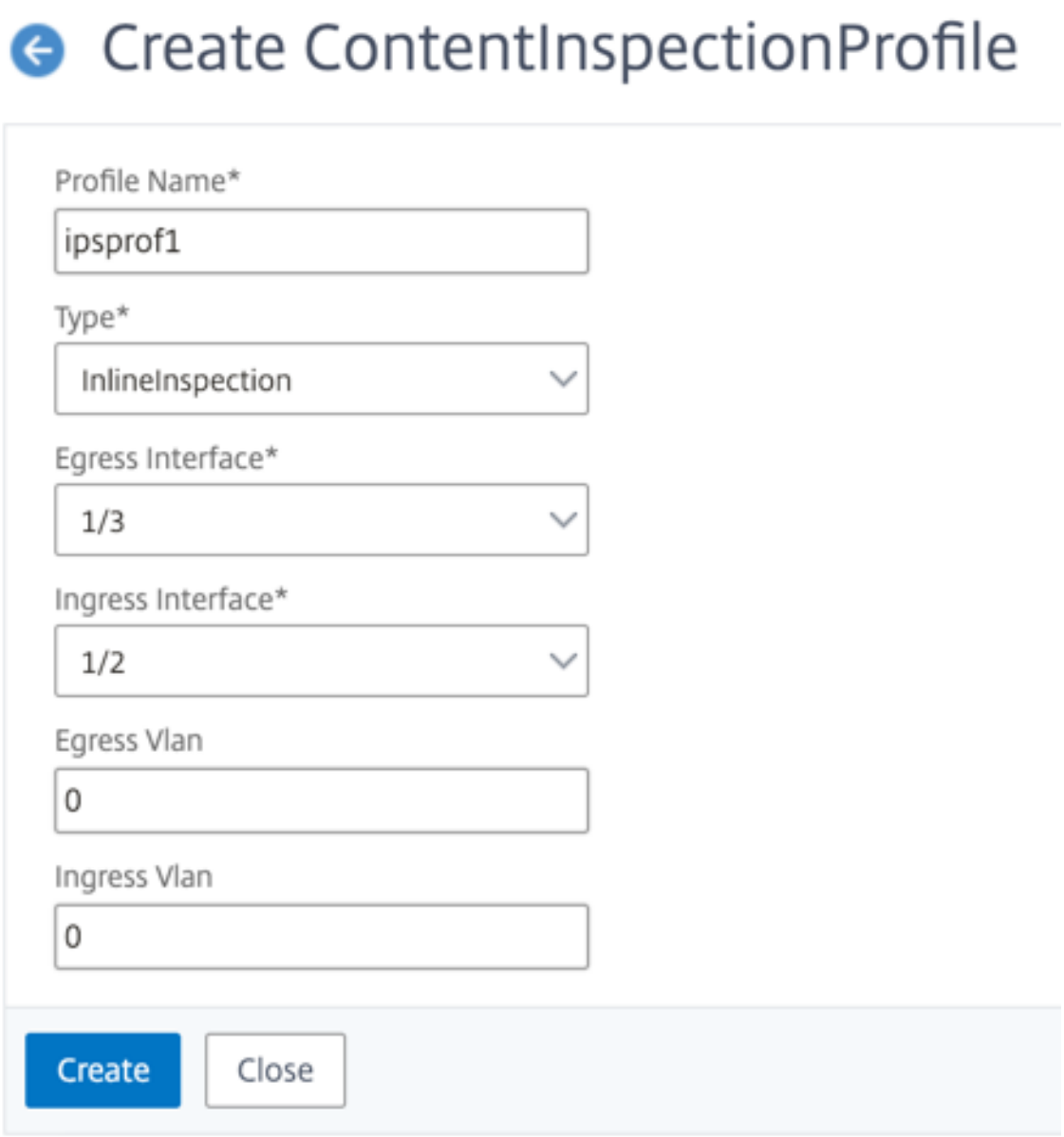




3. Navigate to **Secure Web Gateway > Content Inspection > Content Inspection Profiles**. Click **Add**.

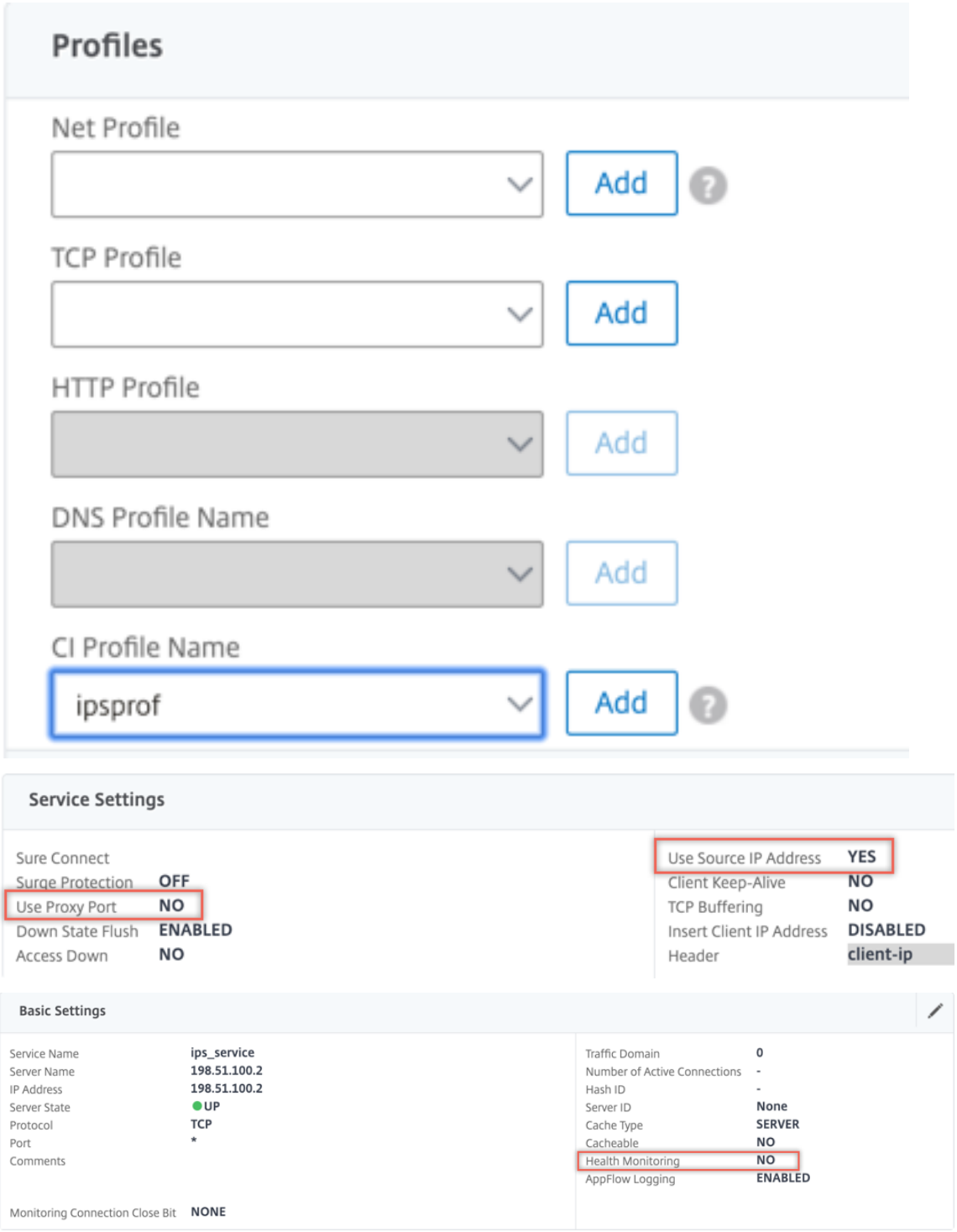


Specify the ingress and egress interfaces.



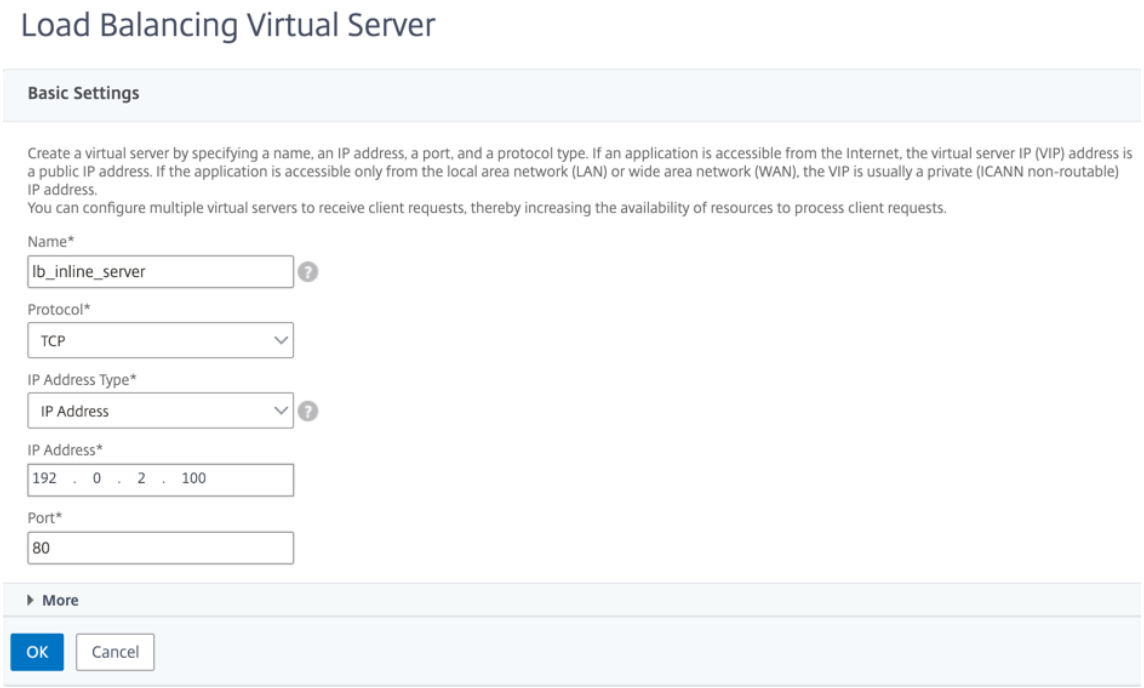
Create two profiles. Specify a different ingress and egress interface in the second profile.

4. Navigate to **Load Balancing > Services > Add** and add a service. In **Advanced Settings**, click **Profiles**. In the **CI Profile Name** list, select the content inspection profile created earlier. In **Service Settings**, set **Use Source IP Address** to YES and **Use Proxy Port** to No. In **Basic Settings**, set **Health Monitoring** to NO. Turn on health monitoring only if you bind this service to a TCP monitor. If you bind a monitor to a service, then set the TRANSPARENT option in the monitor to ON.



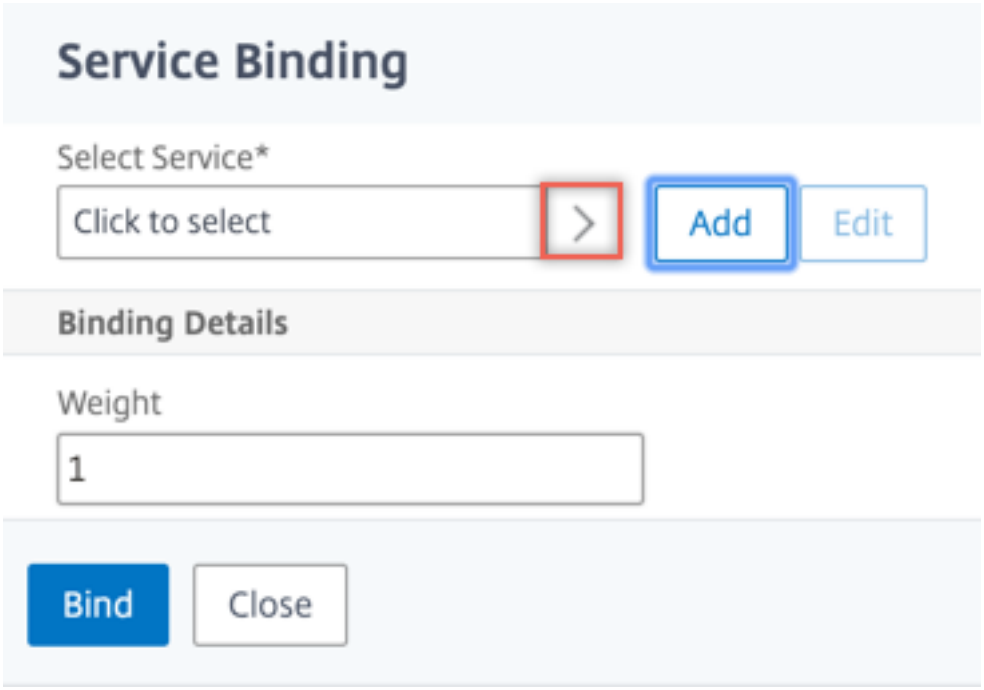
Create two services. Specify dummy IP addresses that are not owned by any of the devices, including the inline devices.

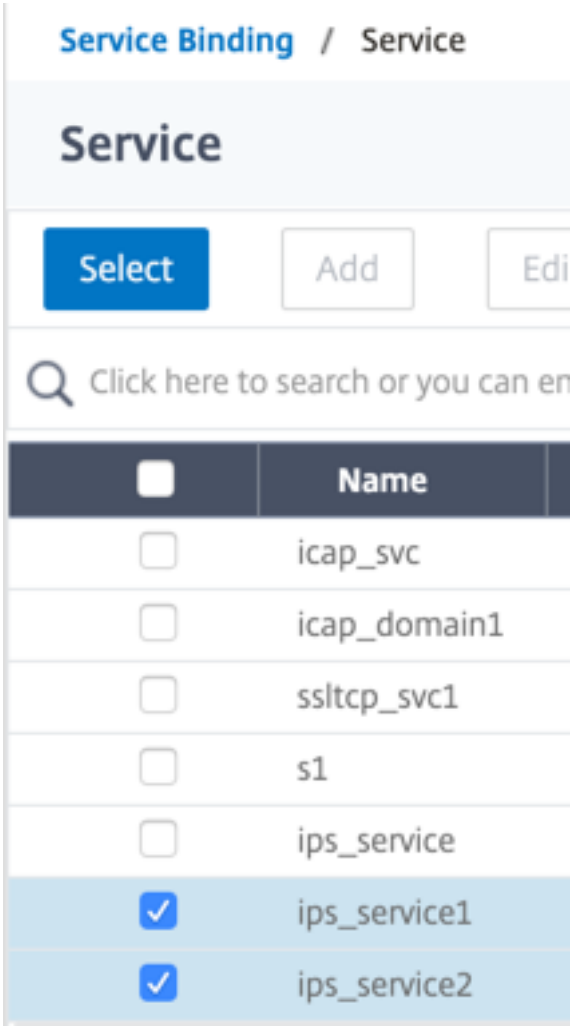
5. Navigate to **Load Balancing > Virtual Servers > Add**. Create a TCP load balancing virtual server.

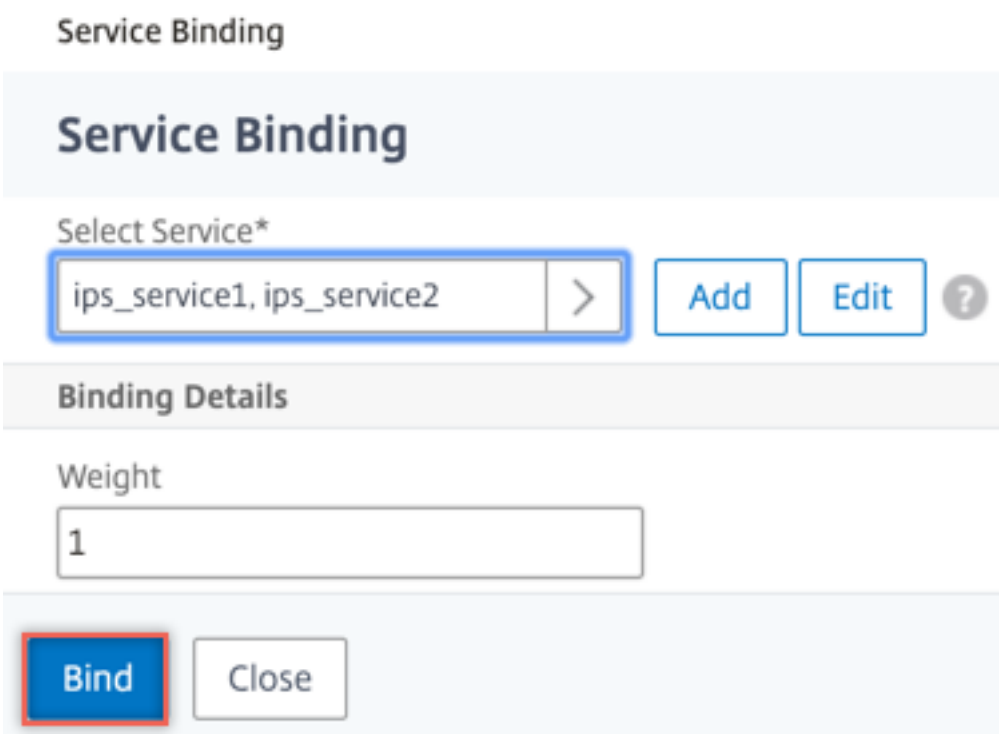


Click **OK**.

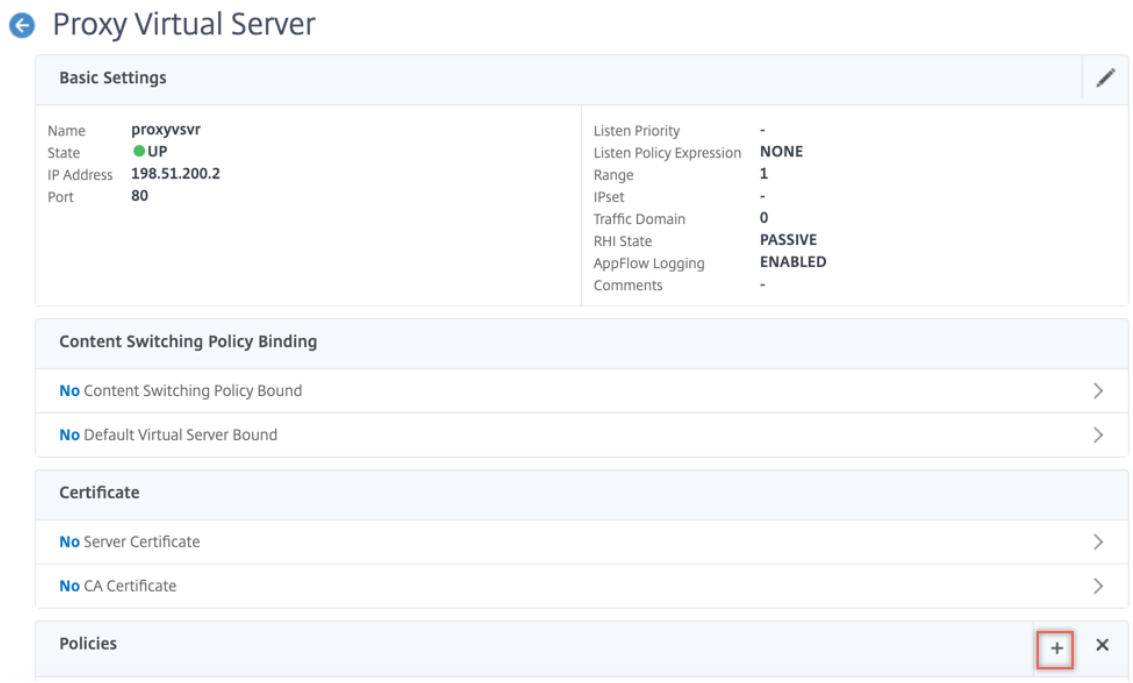
6. Click inside the **Load Balancing Virtual Server Service Binding** section. In **Service Binding**, click the arrow in **Select Service**. Select the two services created earlier, and click **Select**. Click **Bind**.



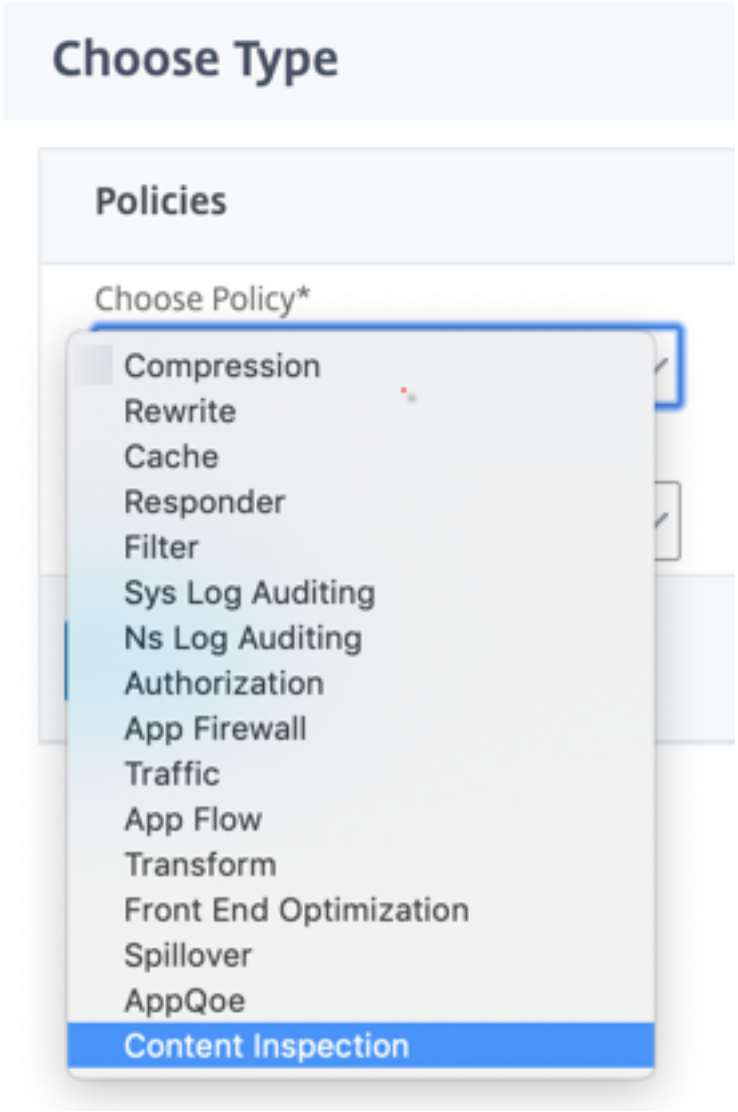




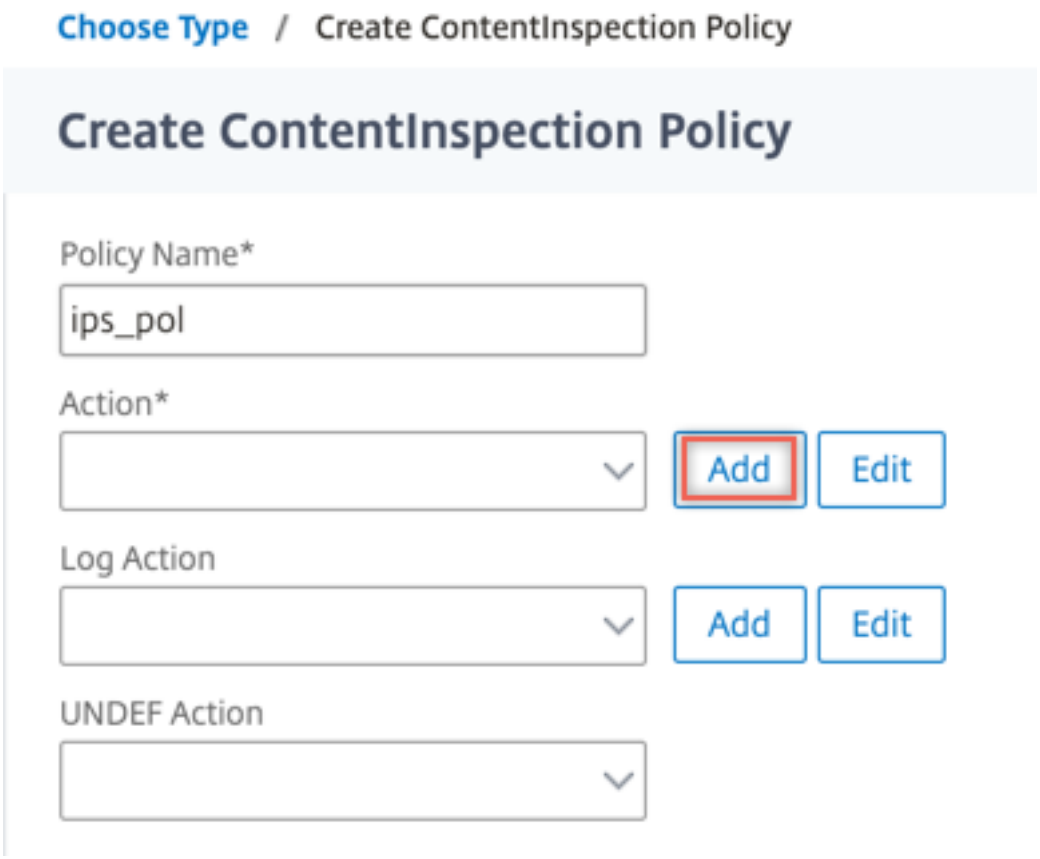
7. Navigate to **Secure Web Gateway > Proxy Virtual Servers> Add**. Specify a name, IP address, and port. In **Advanced Settings**, select **Policies**. Click the “+” sign.



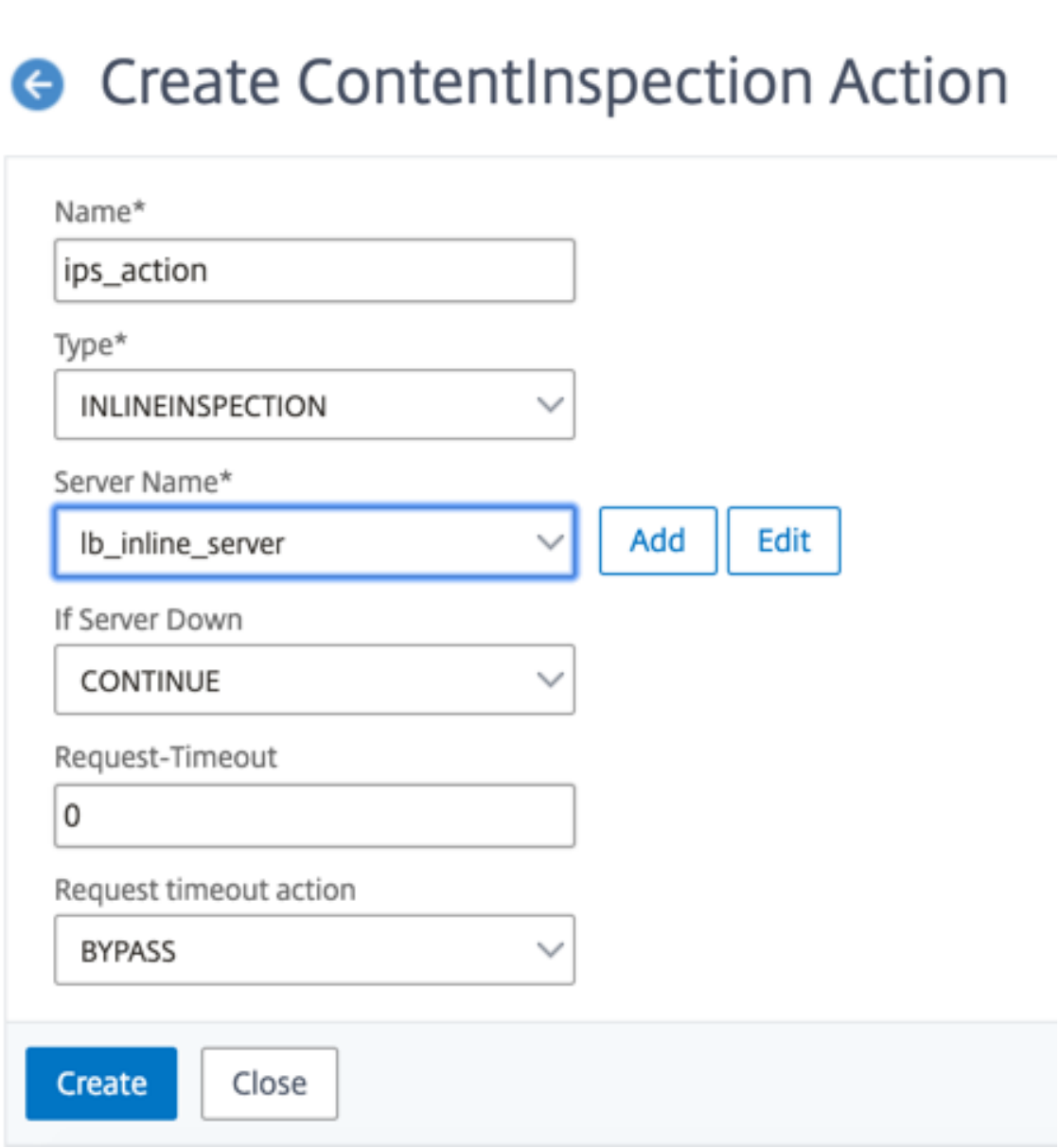
8. In **Choose Policy** select **Content Inspection**. Click **Continue**.



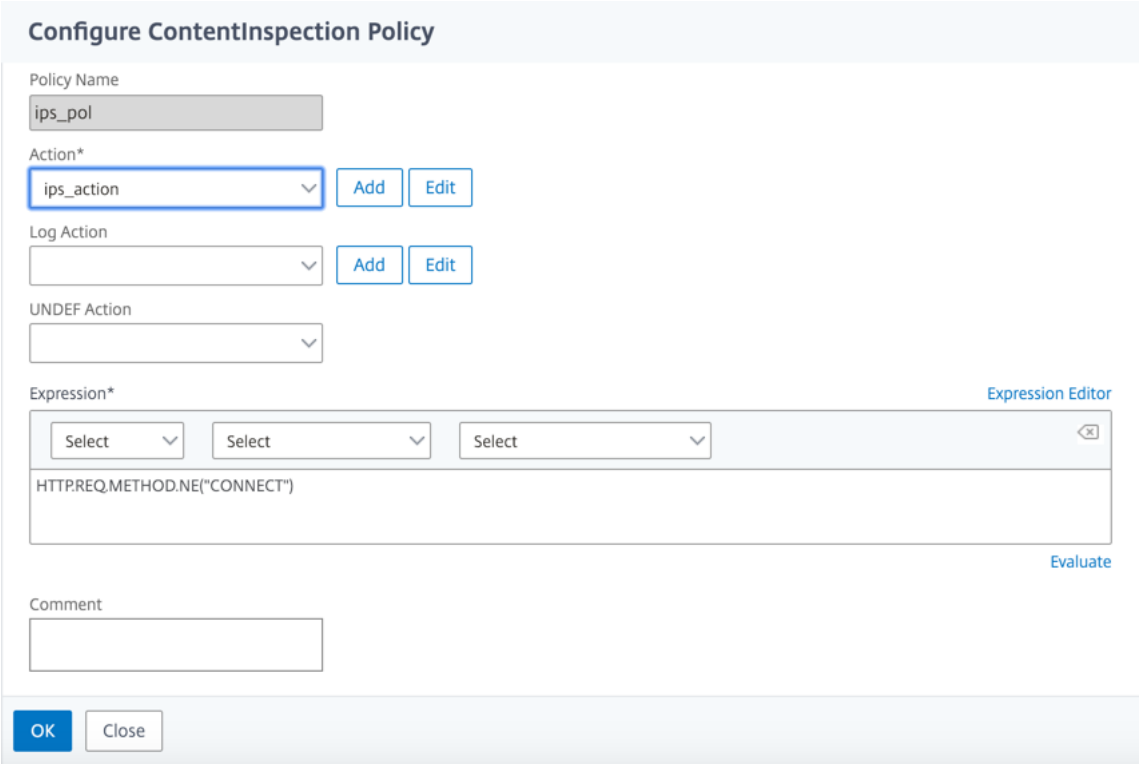
9. Click **Add**. Specify a name. In **Action**, click **Add**.



10. Specify a name. In **Type**, select **INLINEINSPECTION**. In **Server Name**, select the load balancing virtual server created earlier.



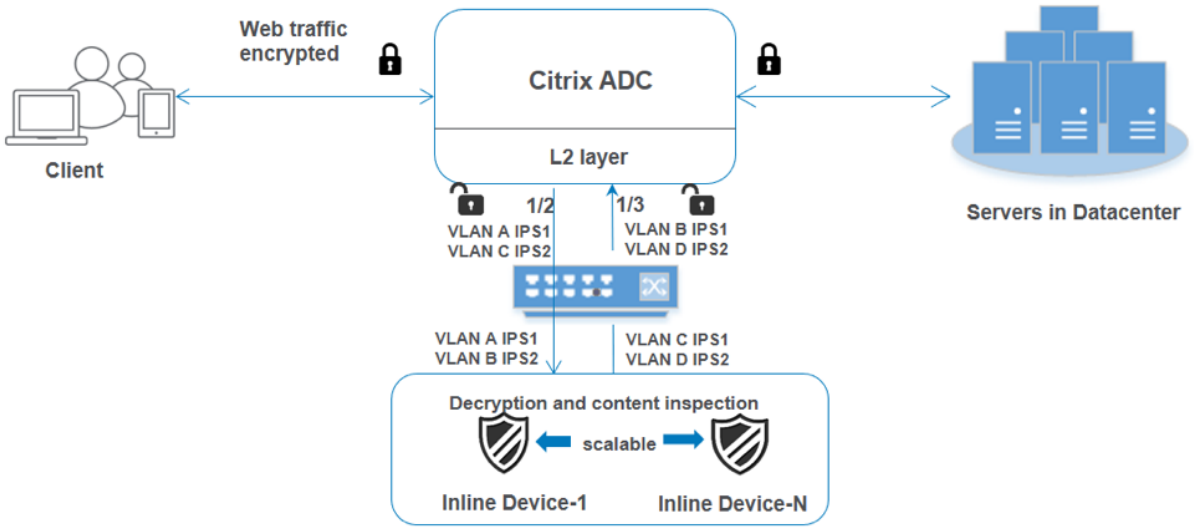
11. Click **Create**. Specify the rule and click **Create**.



12. Click **Bind**.
13. Click **Done**.

Scenario 3: Load balance multiple inline devices with shared interfaces

If you are using two or more inline devices, you can load balance the devices using different content inspection services with shared interfaces. In this case, the SSL forward proxy appliance load balances the subset of traffic sent to each device through a shared interface. The subset is decided based on the policies configured. For example, TXT or image files might not be sent for inspection to the inline devices.



The basic configuration remains the same as in scenario 2. For this scenario, bind the interfaces to different VLANs to segregate the traffic for each inline device. Specify the VLANs in the content inspection profiles. Perform the following extra steps:

1. Bind the shared interfaces to different VLANs.
2. Specify the ingress and egress VLANs in the content inspection profiles.

Configuration using the CLI

Type the following commands at the command prompt. Examples are given after each command.

1. Enable MBF.

```
enable ns mode mbf
```

1. Enable the feature.

```
enable ns feature contentInspection
```

1. Bind the shared interfaces to different VLANs.

```
bind vlan <id> -ifnum <interface> -tagged
```

Example:

```
bind vlan 100  - ifnum 1/2 tagged
bind vlan 200  - ifnum 1/3 tagged
bind vlan 300  - ifnum 1/2 tagged
bind vlan 400  - ifnum 1/3 tagged
<!--NeedCopy-->
```


1. Add profile 1 for service 1. Specify the ingress and egress VLANs in the profile.

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile ipsprof1 -type InlineInspection -egressInterface
 "1/3" -ingressinterface "1/2" -egressVlan 100 -ingressVlan 300
```

1. Add profile 2 for service 2. Specify the ingress and egress VLANs in the profile.

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

Example:

```
add contentInspection profile ipsprof2 -type InlineInspection -egressInterface
 "1/3" -ingressinterface "1/2" -egressVlan 200 -ingressVlan 400
```

1. Add service 1.

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

Example:

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof1
```

1. Add service 2.

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

Example:

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof2
```

1. Add a load balancing virtual server.

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

Example:

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. Bind the services to the load balancing virtual server.


```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

Example:

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```

1. Specify the load balancing virtual server in the content inspection action.

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

Example:

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
lb_inline_vserver
```

1. Add a content inspection policy. Specify the content inspection action in the policy.

```
add contentInspection policy <name> -rule <expression> -action <string>
```

Example:

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"CONNECT\")
"-action ips_action
```

1. Add a proxy virtual server.

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

Example:

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. Bind the content inspection policy to the virtual server.

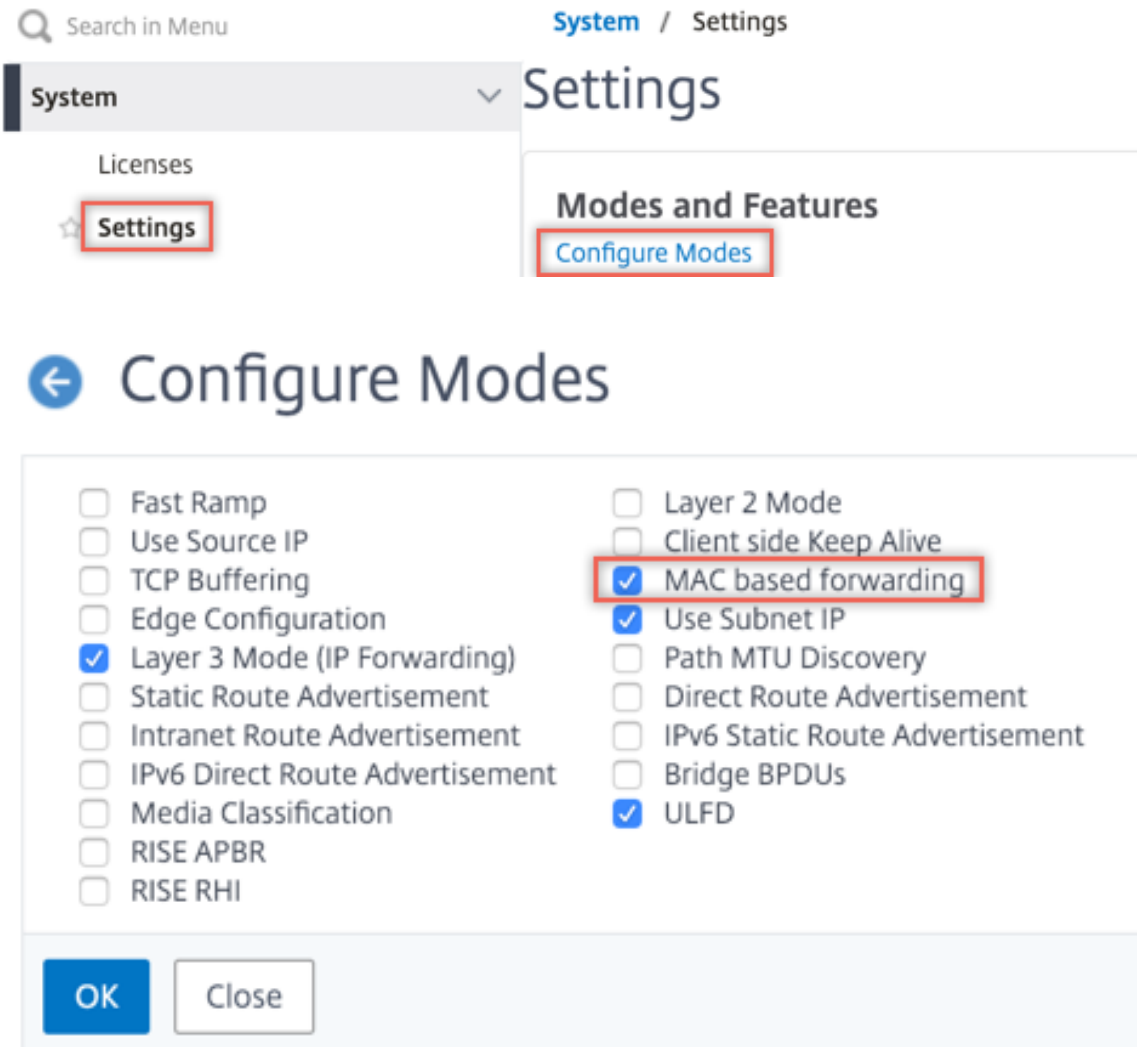
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

Example:

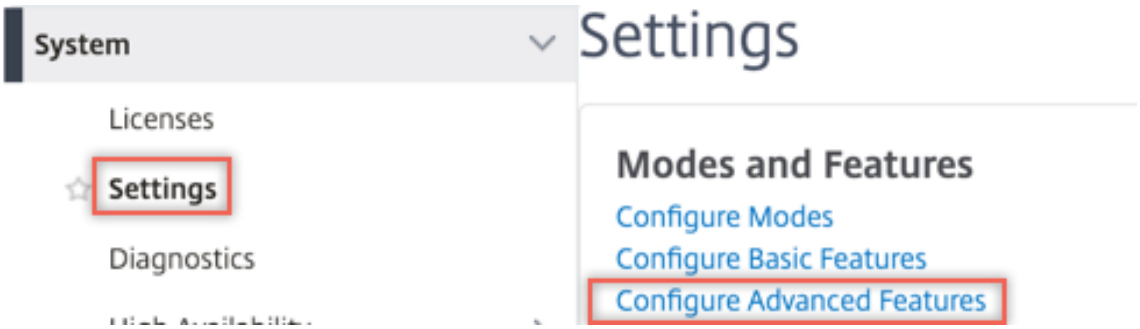
```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

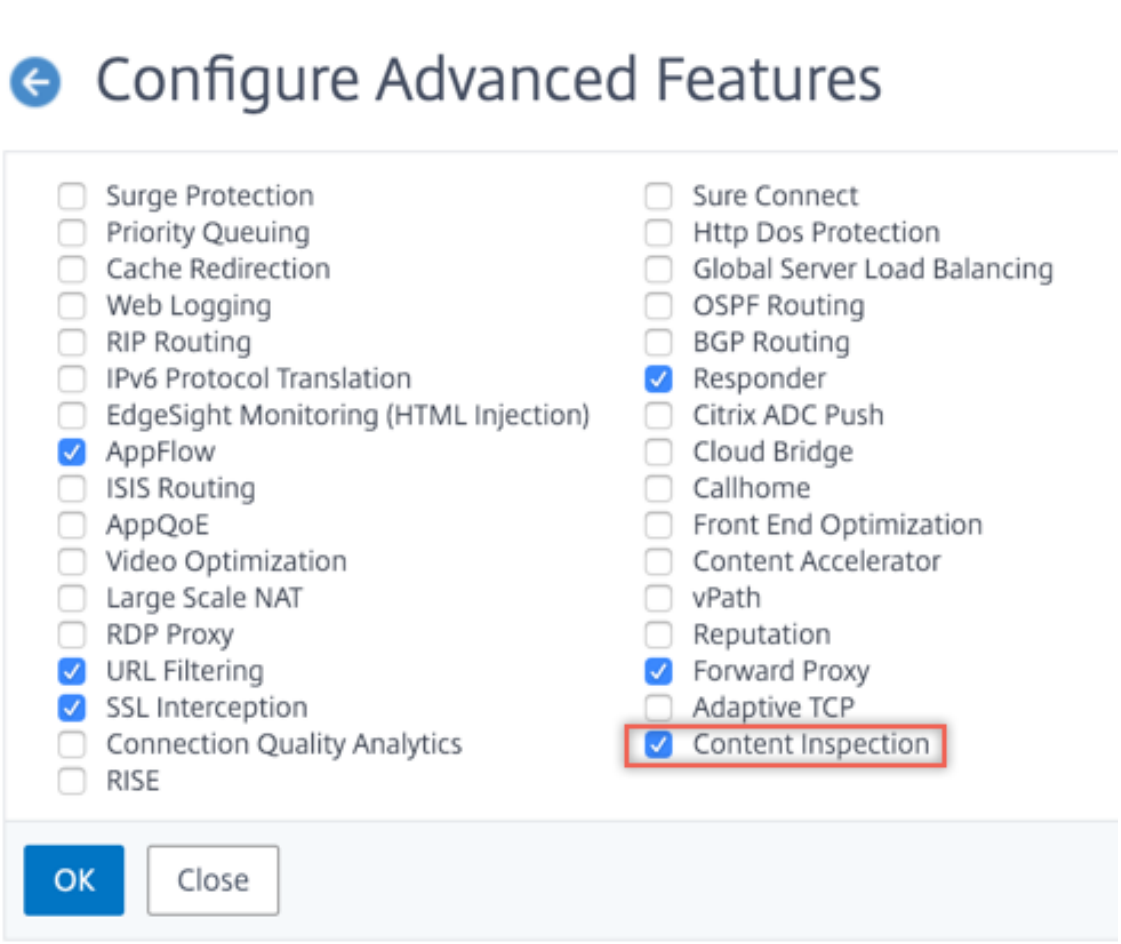
Configuration using the GUI

1. Navigate to **System > Settings**. In **Modes and Features**, click **Configure Modes**.

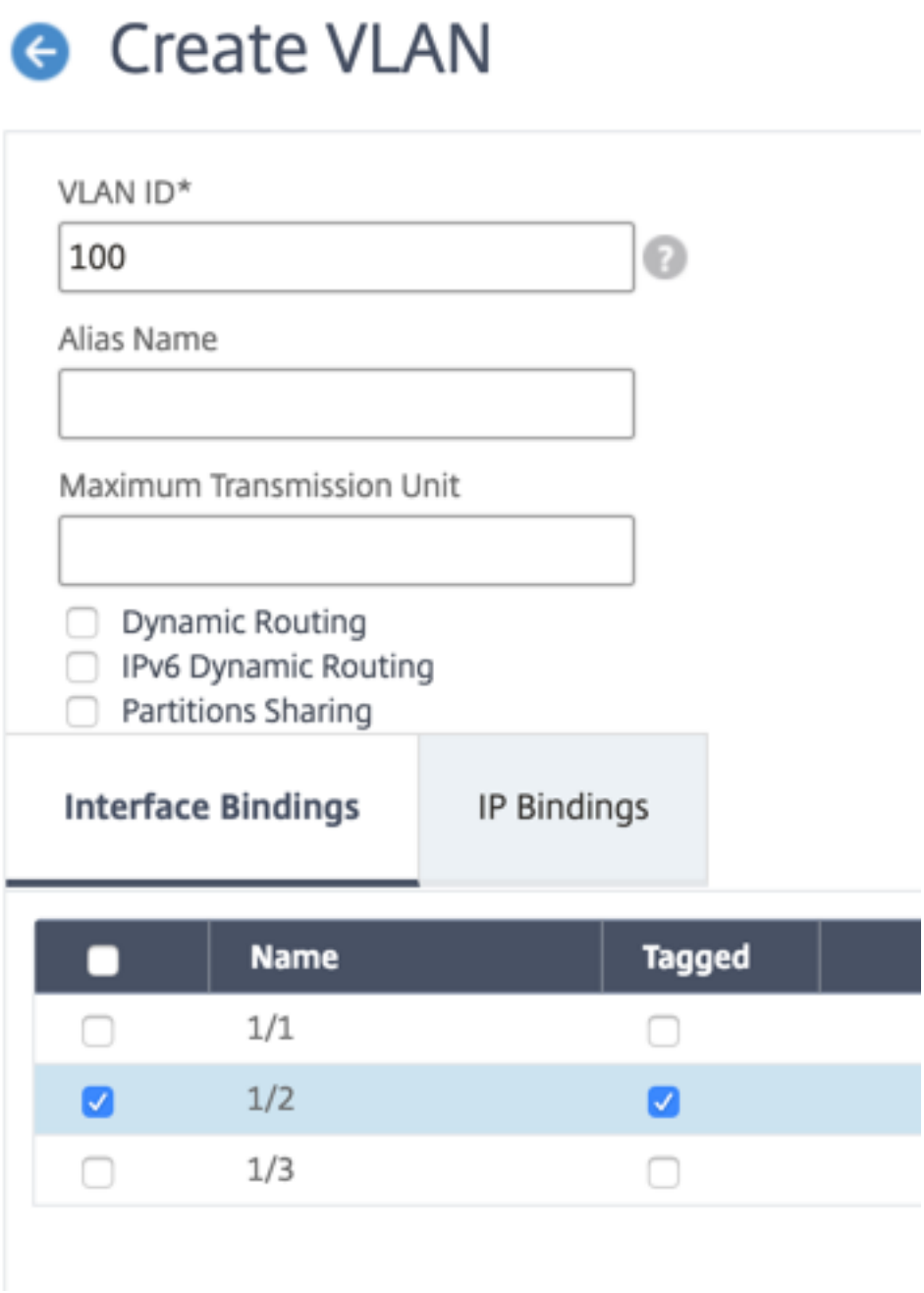


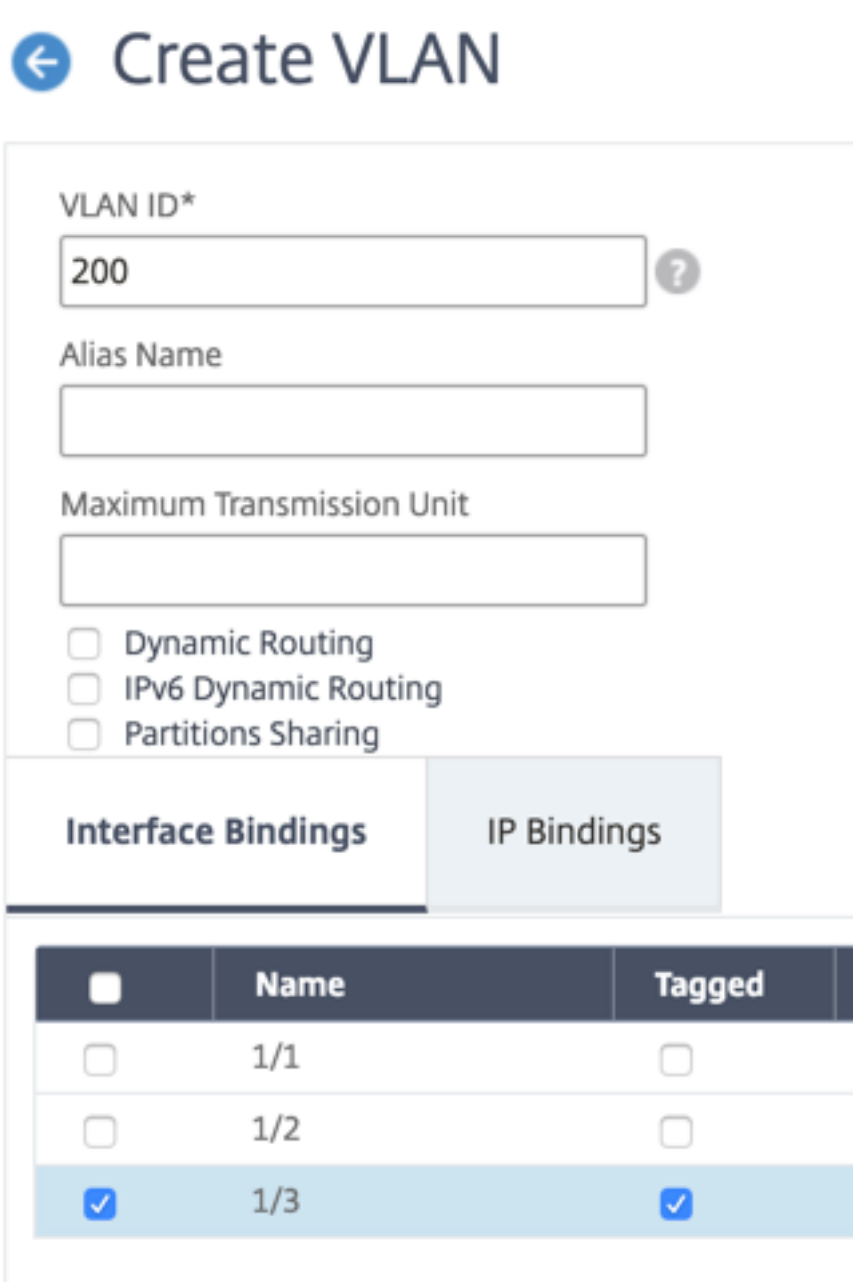
2. Navigate to **System > Settings**. In **Modes and Features**, click **Configure Advanced Features**.

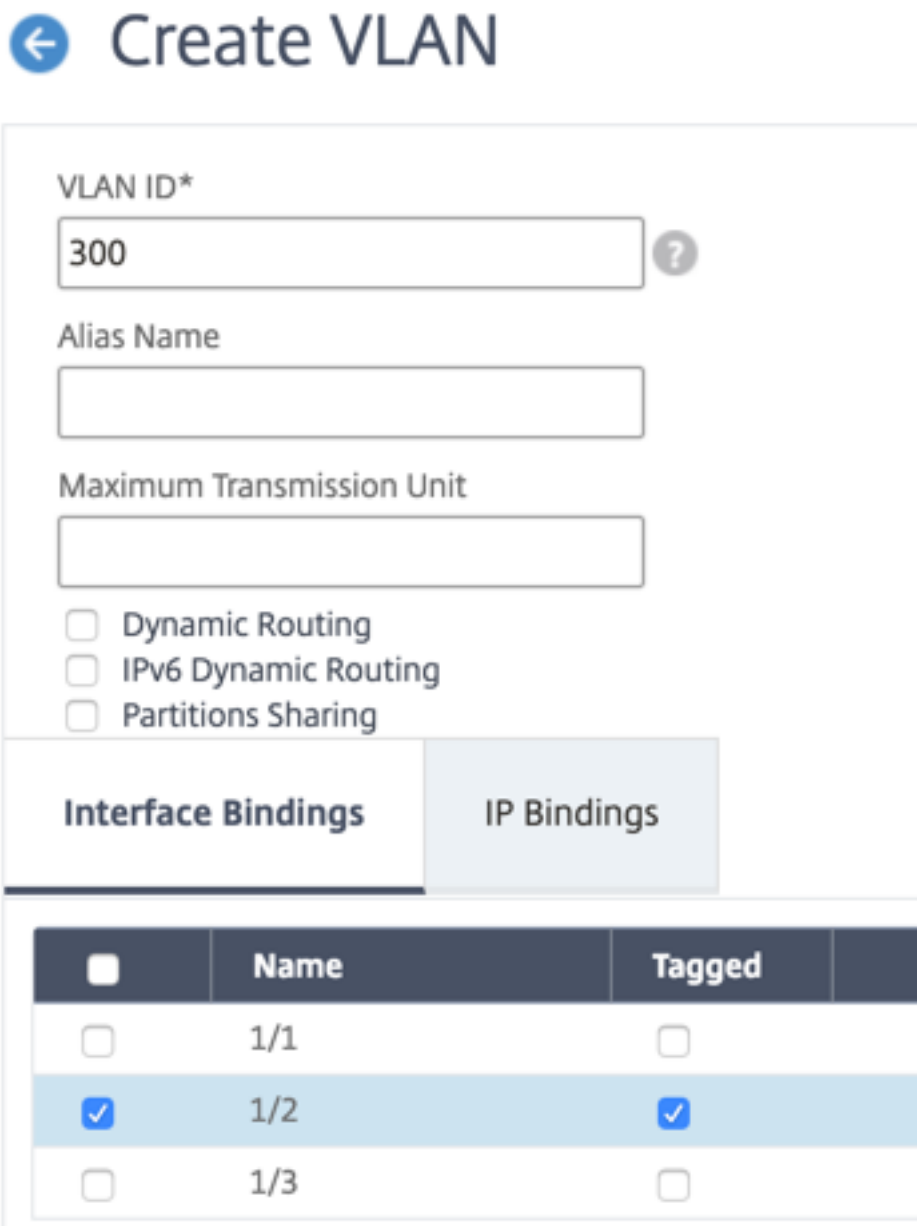


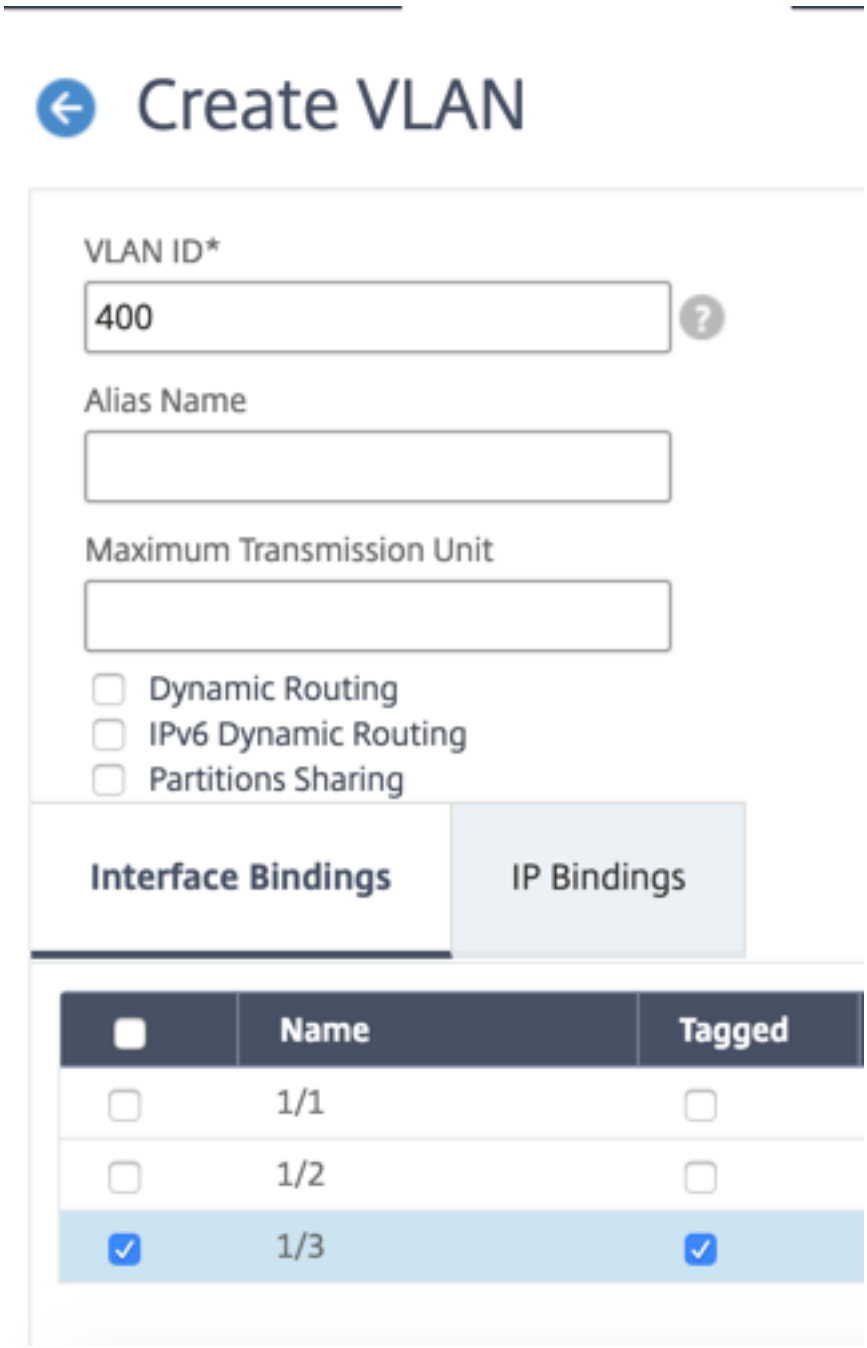


3. Navigate to **System > Network > VLANs > Add**. Add four VLANs and tag them to the interfaces.

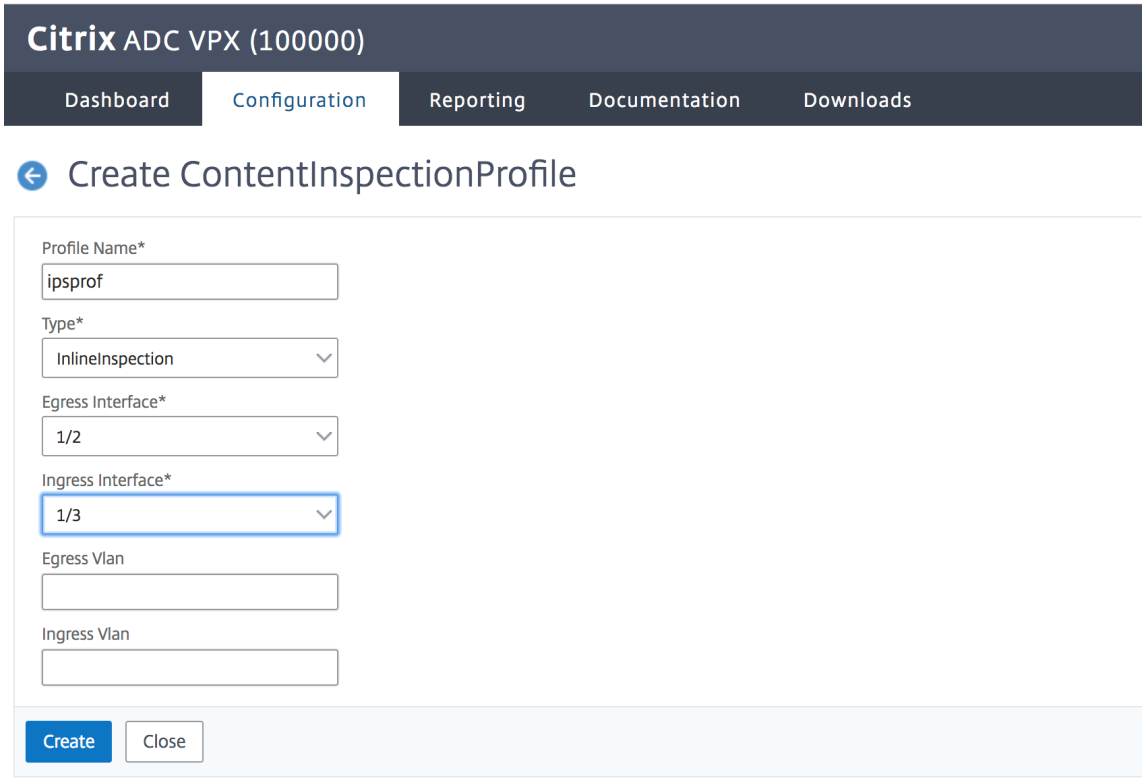




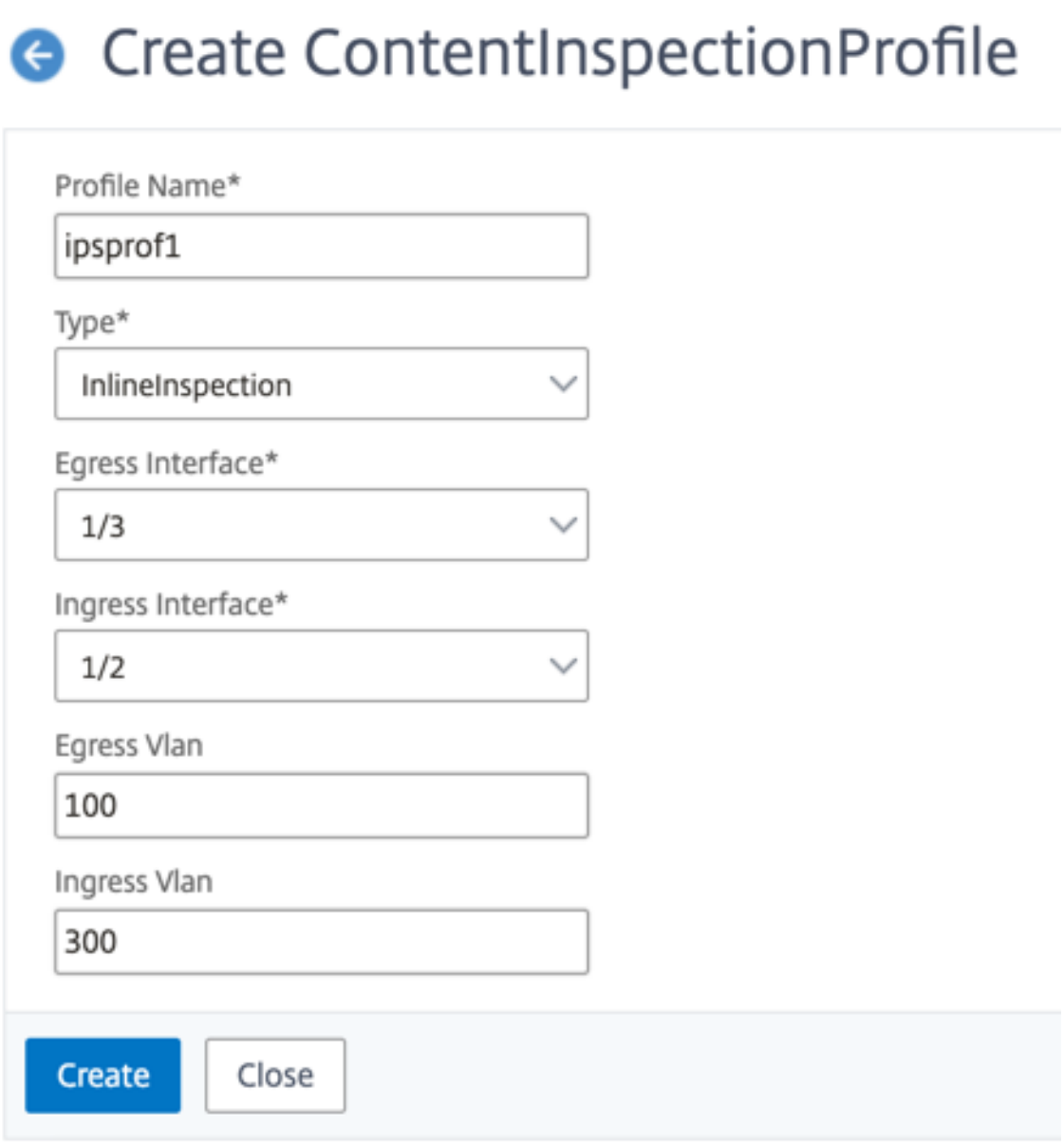




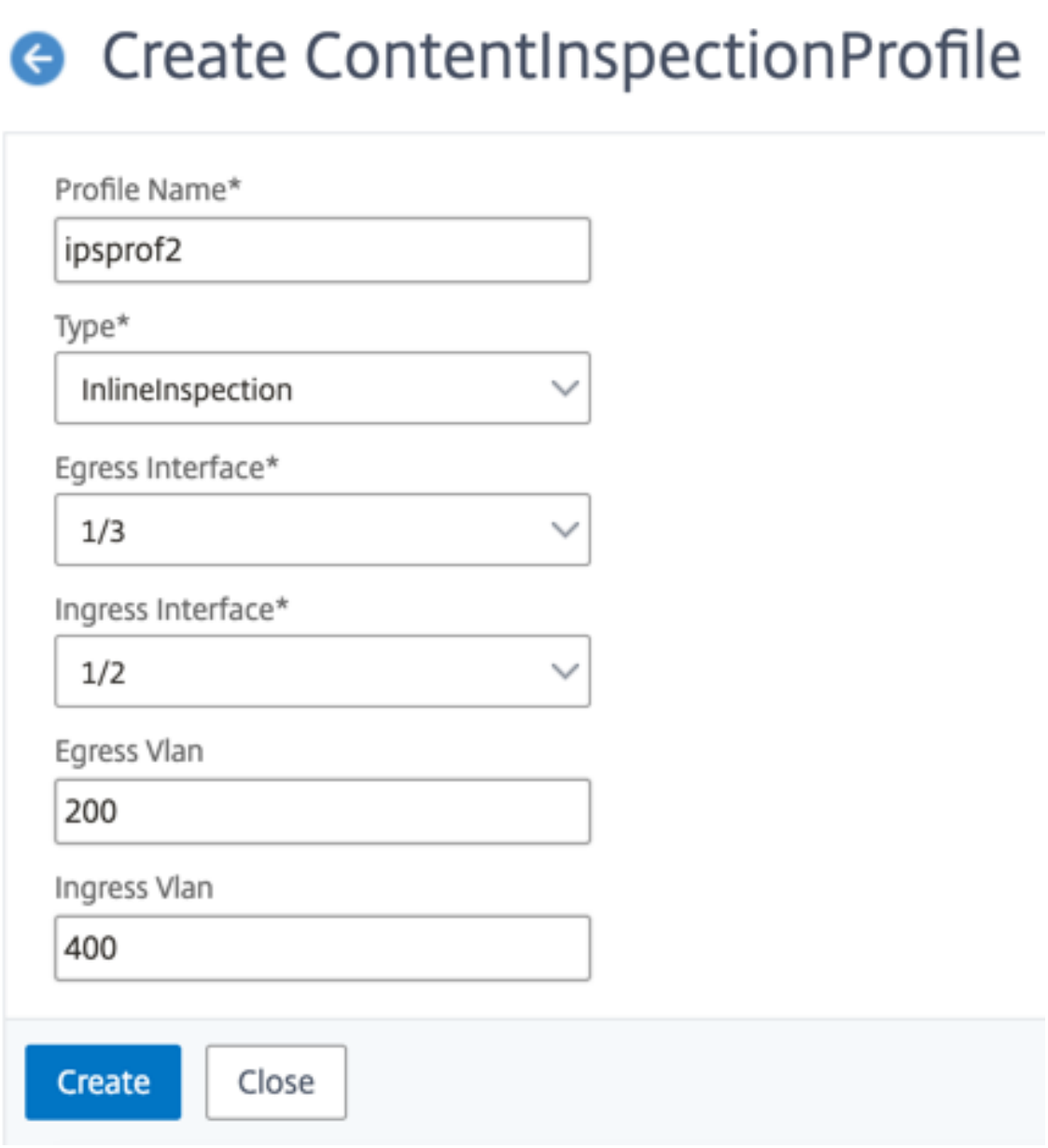
4. Navigate to **Secure Web Gateway > Content Inspection > Content Inspection Profiles**. Click **Add**.



Specify the ingress and egress VLANs.

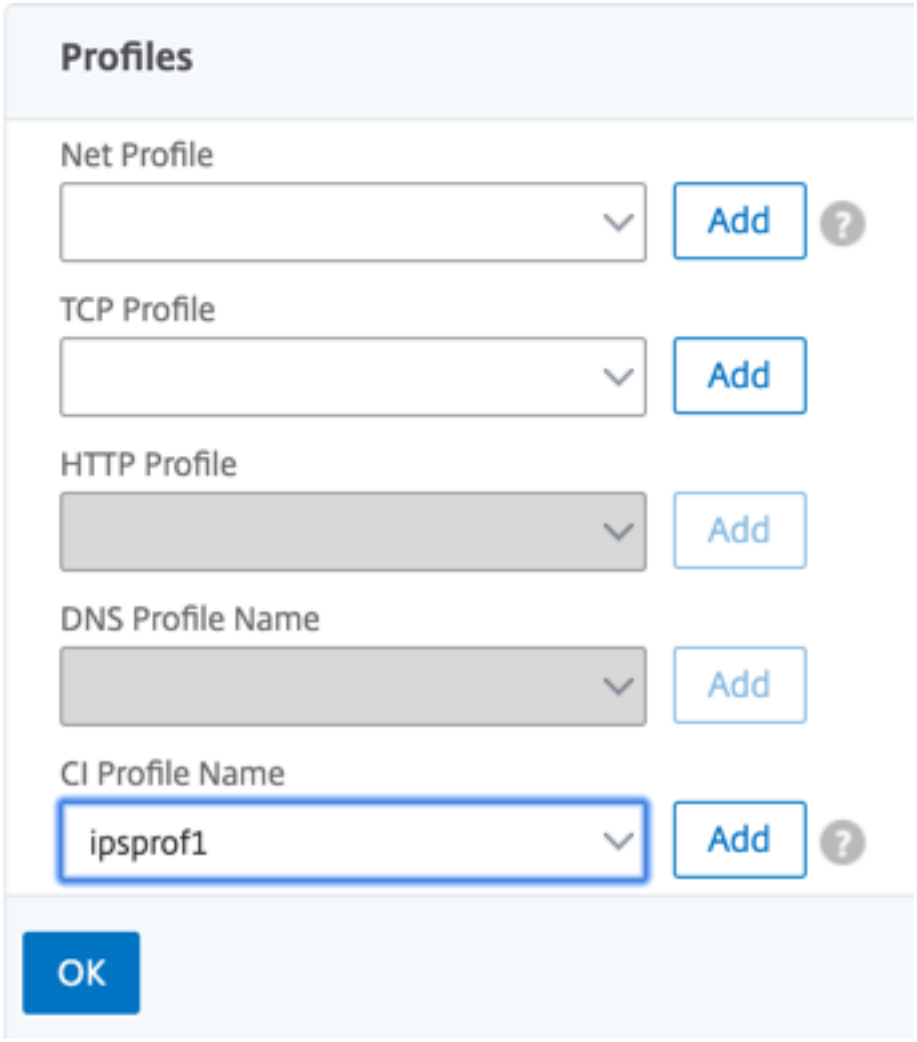


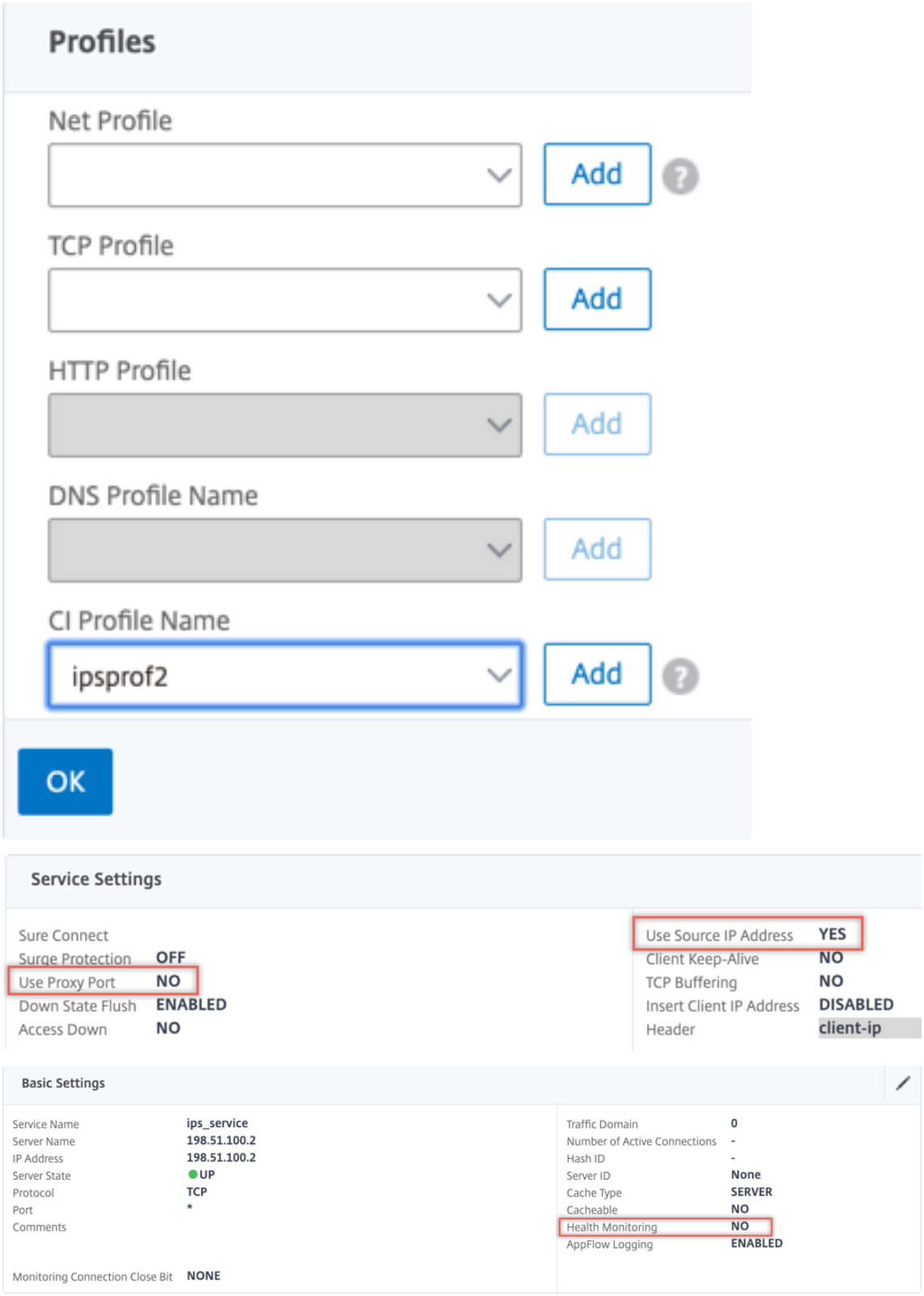
Create another profiles. Specify a different ingress and egress VLAN in the second profile.



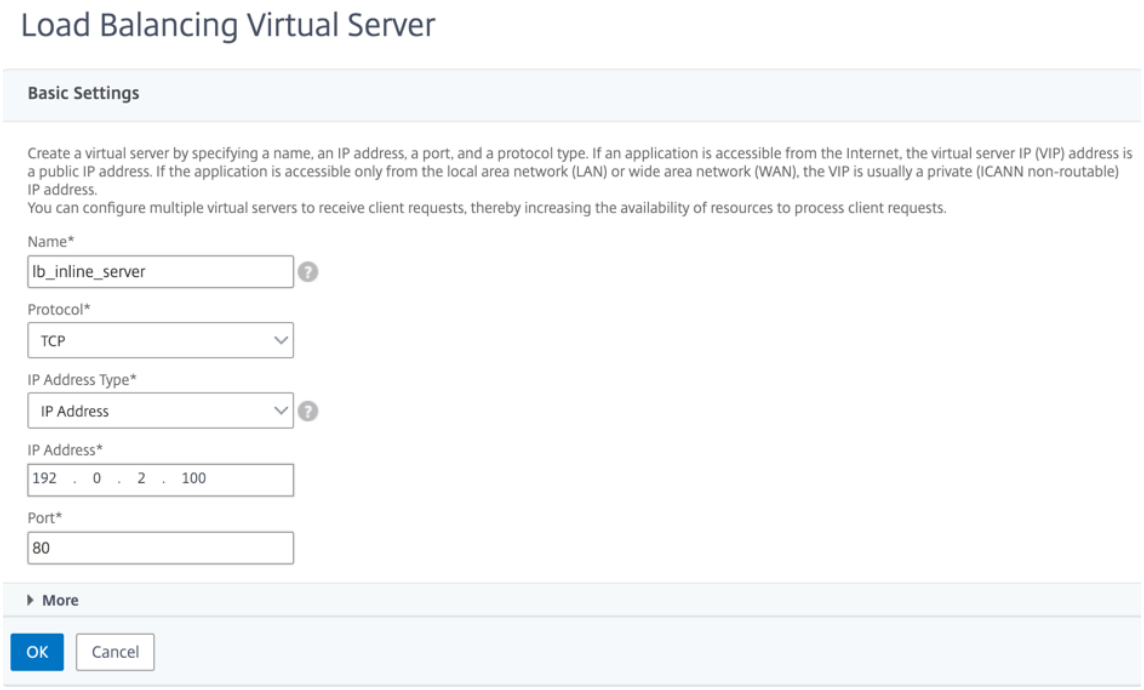
5. Navigate to **Load Balancing > Services > Add** and add a service. In **Advanced Settings**, click **Profiles**. In the **CI Profile Name** list, select the content inspection profile created earlier. In **Service Settings**, set **Use Source IP Address** to YES and **Use Proxy Port** to No. In **Basic Settings**, set **Health Monitoring** to NO.

   Create two services. Specify dummy IP addresses that are not owned by any of the devices, including the inline devices. Specify profile 1 in service 1, and profile 2 in service 2.

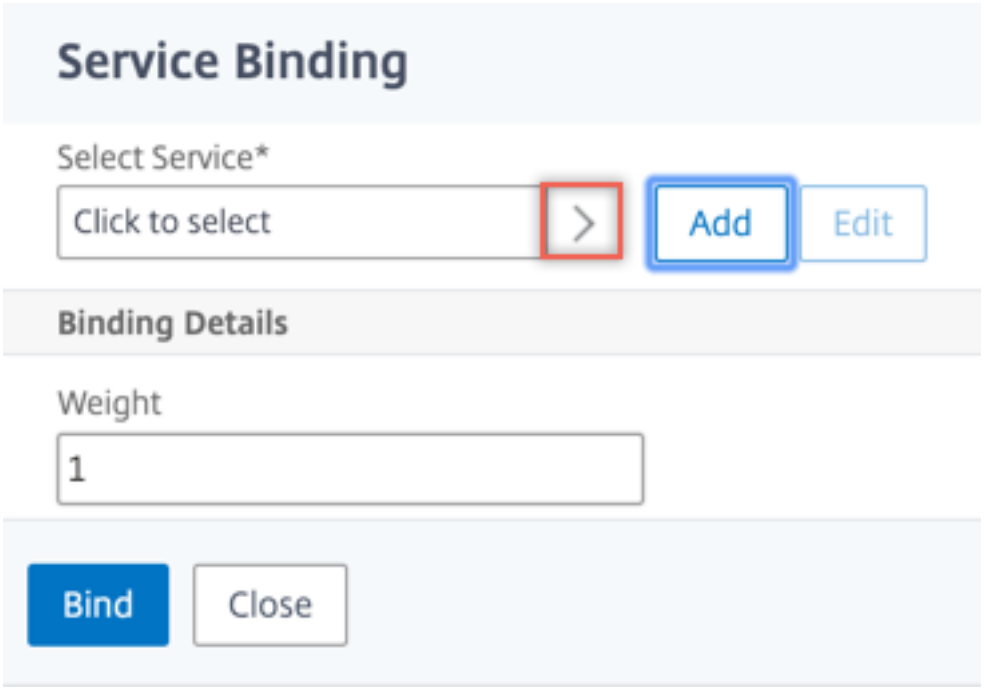


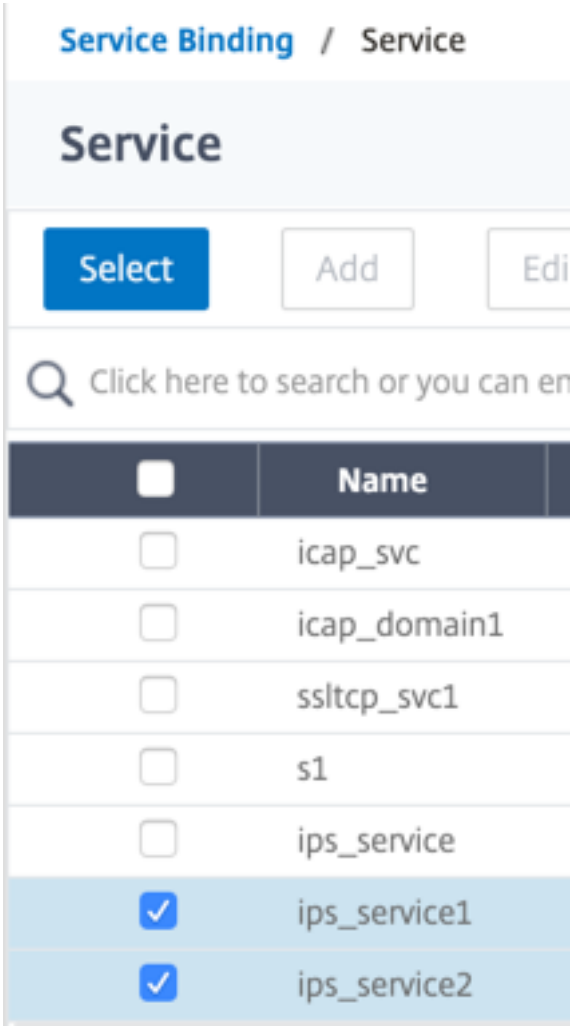


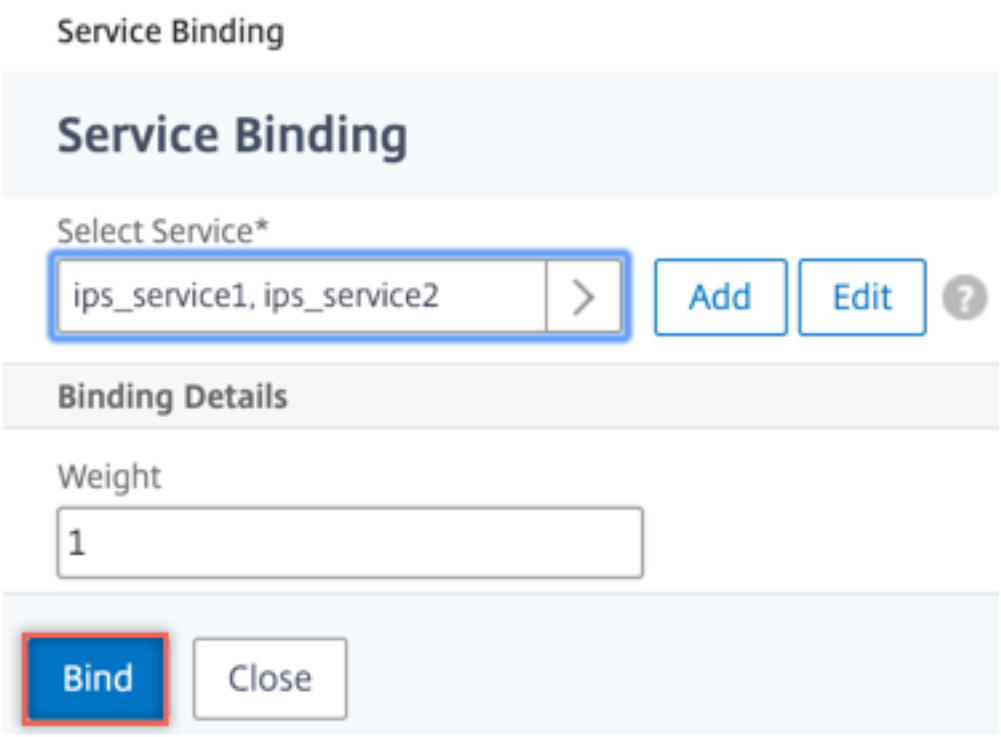
6. Navigate to **Load Balancing > Virtual Servers > Add**. Create a TCP load balancing virtual server.



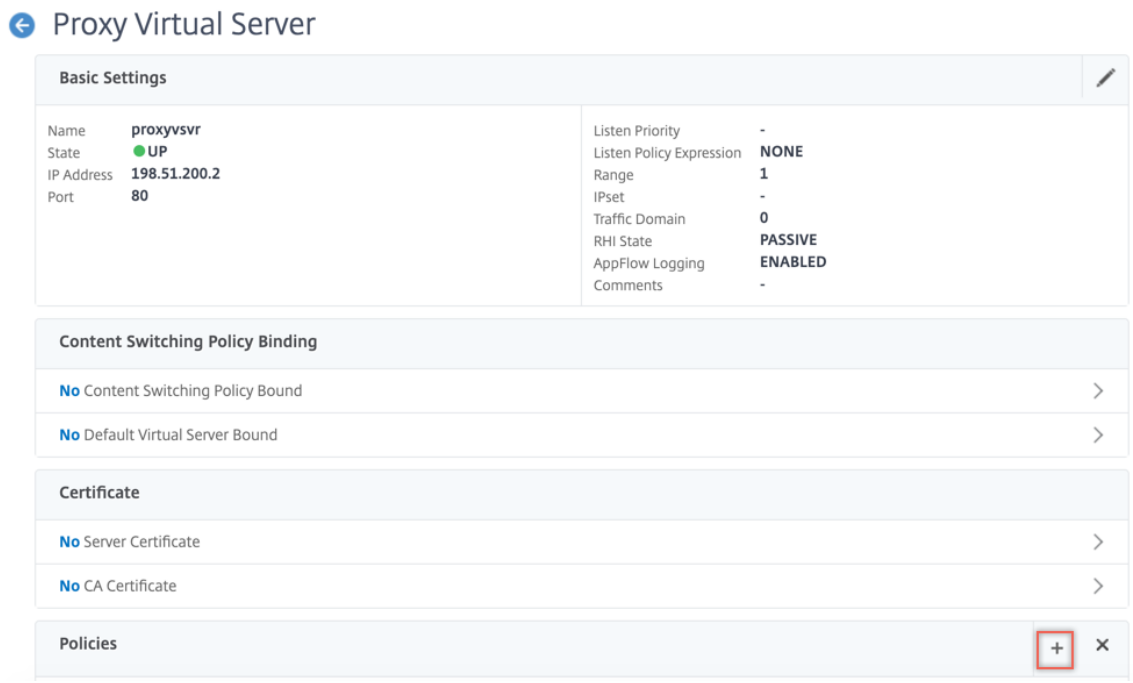
7. Click **OK**.
8. Click inside the **Load Balancing Virtual Server Service Binding** section. In **Service Binding**, click the arrow in **Select Service**. Select the two services created earlier, and click **Select**. Click **Bind**.



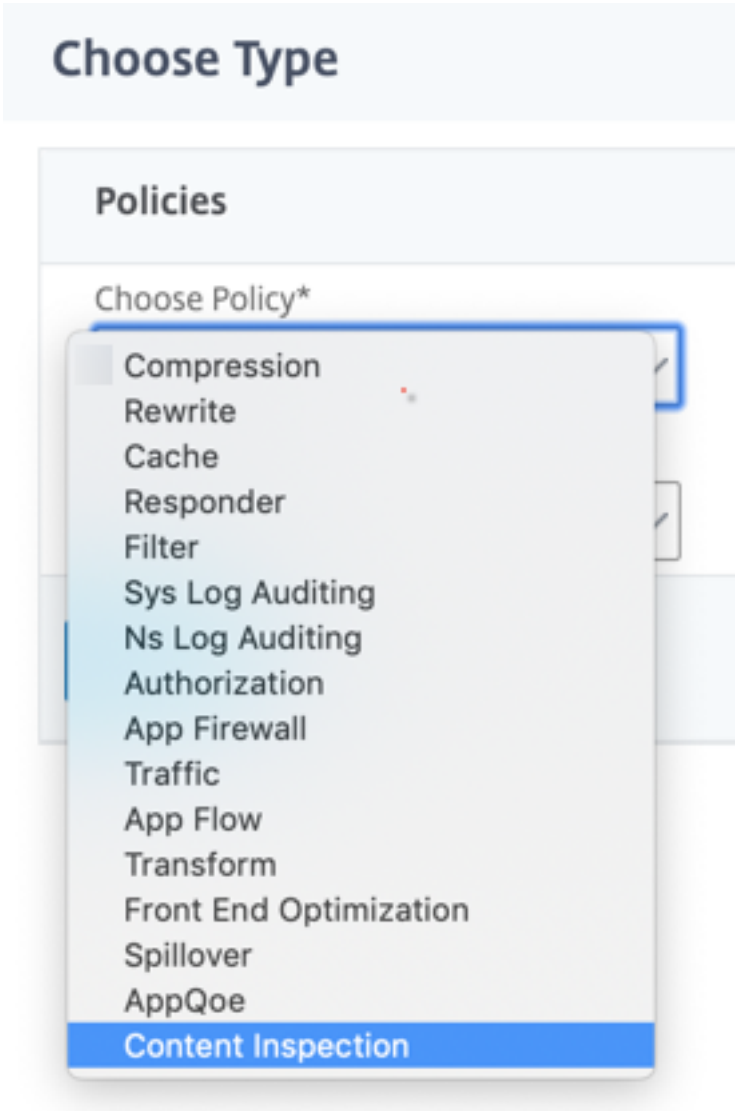




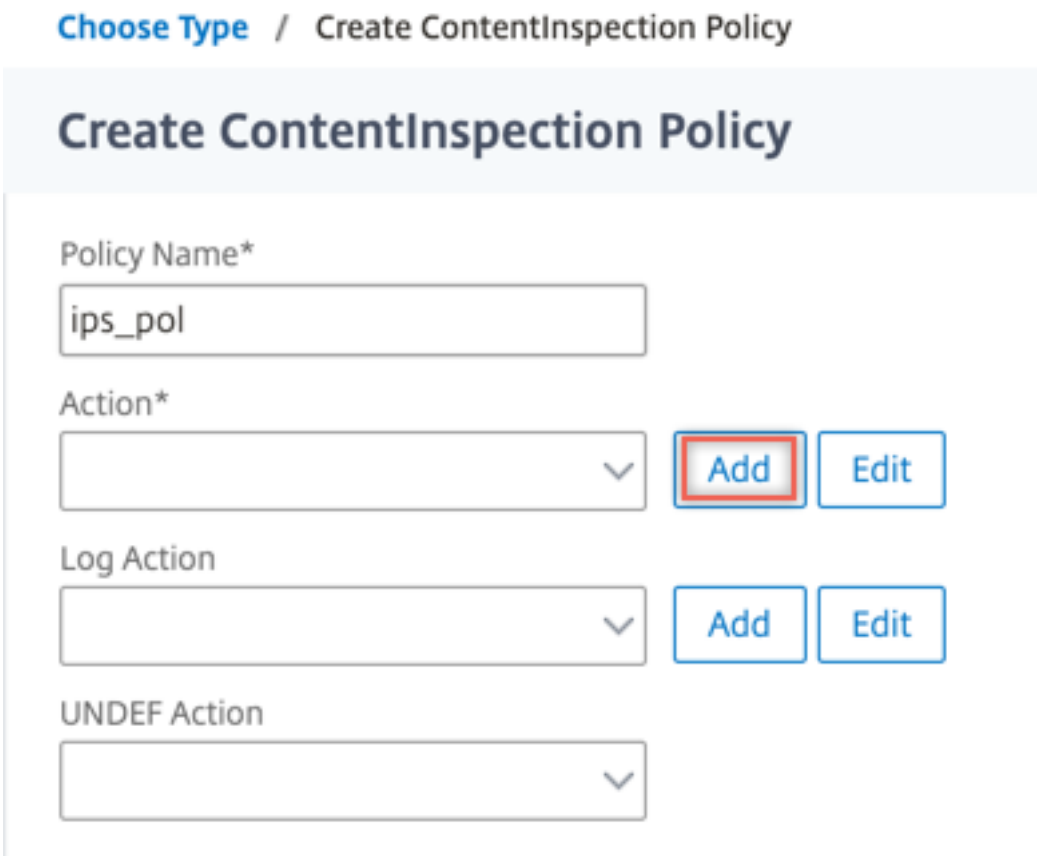
9. Navigate to **Secure Web Gateway > Proxy Virtual Servers> Add**. Specify a name, IP address, and port. In **Advanced Settings**, select **Policies**. Click the “+” sign.



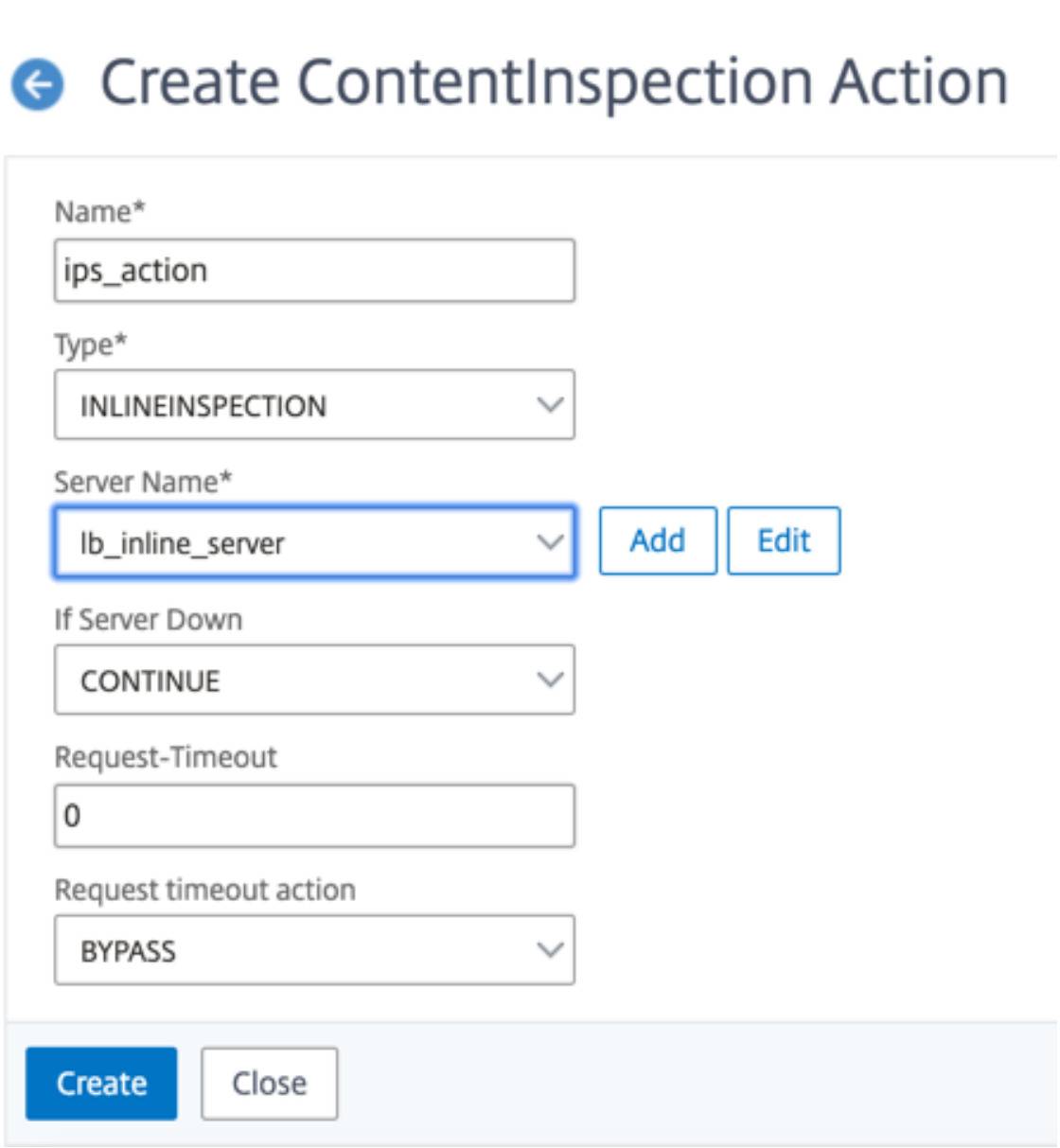
10. In **Choose Policy** select **Content Inspection**. Click **Continue**.



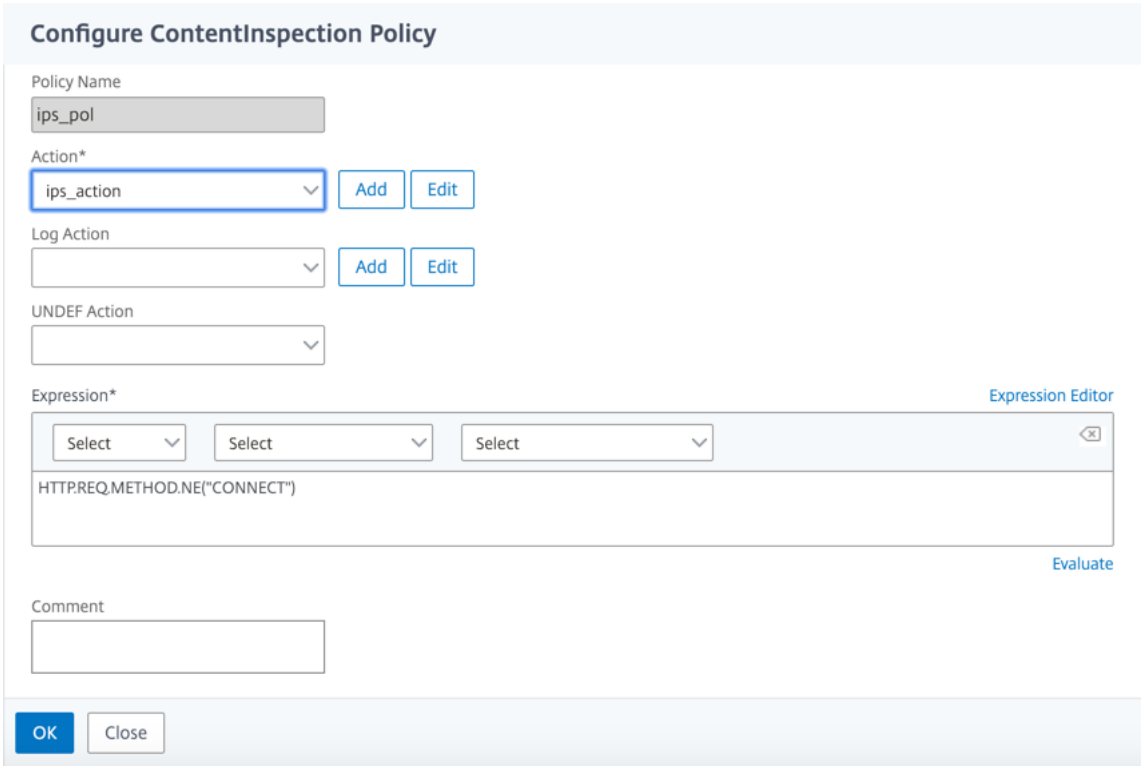
11. Click **Add**. Specify a name. In **Action**, click **Add**.



12. Specify a name. In **Type**, select **INLINEINSPECTION**. In **Server Name**, select the load balancing virtual server created earlier.



13. Click **Create**. Specify the rule and click **Create**.



14. Click **Bind**.
15. Click **Done**.

Integrating Citrix ADC with passive security devices (Intrusion Detection System)

September 14, 2021

A Citrix ADC appliance is now integrated with passive security devices such as the Intrusion Detection System (IDS). These passive devices store logs and trigger alerts when it detects a bad or non-compliant traffic. It also generates reports for the compliance purpose. If the Citrix ADC appliance is integrated with two or more IDS devices and when there is a high volume of traffic, the appliance can load balance the devices by cloning traffic at the virtual server level.

For advanced security protection, a Citrix ADC appliance is integrated with passive security devices such as IDS deployed in detection-only mode. These devices store log and trigger alerts when it sees a bad or non-compliant traffic. It also generates reports for the compliance purpose. Following are some of the benefits of integrating the Citrix ADC with an IDS device.

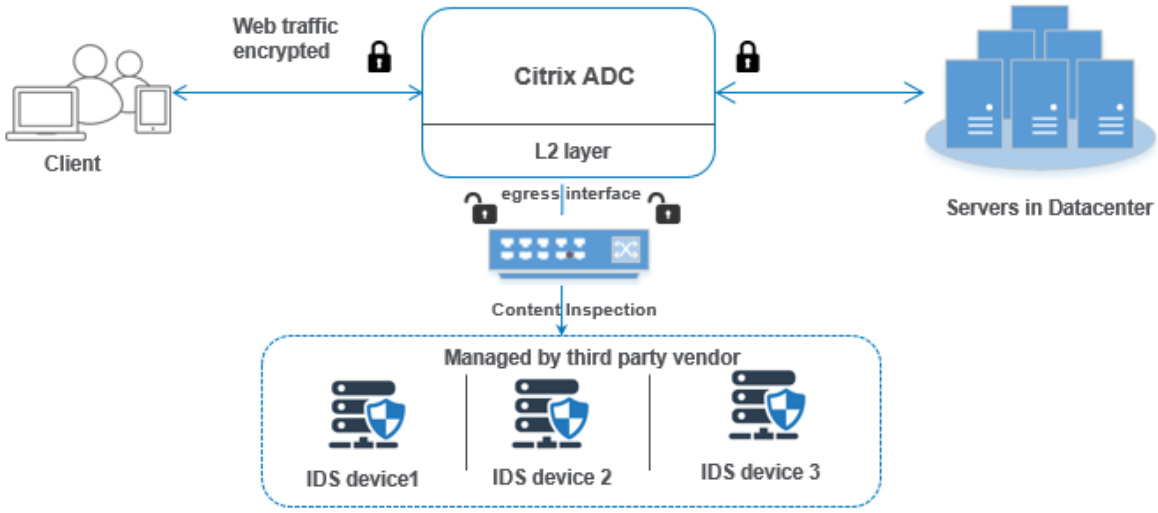
- **Inspecting encrypted traffic**. Most security devices bypass encrypted traffic, thereby leaving servers vulnerable to attacks. A Citrix ADC appliance can decrypt traffic and send it to IDS de-

vices for enhancing the customer's network security.

- **Offloading inline devices from TLS/SSL processing**. TLS/SSL processing is expensive and it results in high system CPU in intrusion detection devices if they decrypt the traffic. As encrypted traffic is growing at a fast pace, these systems fail to decrypt and inspect encrypted traffic. Citrix ADC helps in offloading traffic to IDS devices from TLS/SSL processing. This way of offloading data results in an IDS device supporting a high volume of traffic inspection.
- **Loading balancing IDS devices**. The Citrix ADC appliance load balances multiple IDS devices when there is a high volume of traffic by cloning traffic at the virtual server level.
- **Replicating traffic to passive devices**. The traffic flowing into the appliance can be replicated to other passive devices for generating compliance reports. For example, few government agencies mandate every transaction to be logged in some passive devices.
- **Fanning traffic to multiple passive devices**. Some customers prefer to fan out or replicate incoming traffic into multiple passive devices.
- **Smart selection of traffic**. Every packet flowing into the appliance might not be must be content inspected, for example download of text files. User can configure the Citrix ADC appliance to select specific traffic (for example .exe files) for inspection and send the traffic to IDS devices for processing data.

How Citrix ADC is integrated with IDS device with L2 connectivity

The following diagram shows how IDS is integrated with a Citrix ADC appliance.



The component interaction is given as follows:

1. A client sends an HTTP/HTTPS request to the Citrix ADC appliance.
2. The appliance intercepts the traffic and replicates it to an IDS device based on content inspection policy evaluation.

3. If the traffic is an encrypted one, the appliance decrypts the data and sends it as a plain text.
4. Based on policy evaluation, the appliance applies a “MIRROR” type content inspection action.
5. The action has the IDS service or load balancing service (for multiple IDS device integrations) configured in it.
6. The IDS device is configured as content inspection service type “Any” on the appliance. The content inspection service is then associated to the content inspection profile of type “MIRROR” which specifies the egress interface through which the data has to be forwarded to the IDS device. Note: Optionally, you can also configure a VLAN tag in the content inspection profile.
7. The appliance then, replicates the data through the egress interface to one or more IDS devices.
8. Similarly, when the back-end server sends a response to the Citrix ADC, the appliance replicates the data and forwards it to the IDS device.
9. If your appliance is integrated to one or more IDS devices and if you prefer to load balance the devices, then you can use the load balancing virtual server.

Software licensing

To deploy the inline device integration, your Citrix ADC appliance must be provisioned with one of the following licenses:

1. ADC Premium
2. ADC Advanced
3. Telco Advanced
4. Telco Premium

Configuring intrusion detection system integration

You can integrate the IDS device with the Citrix ADC in two different ways.

Scenario 1: Integration with a single IDS device

Following are the steps you must configure using the command line interface.

1. Enable content inspection
2. Add content inspection profile of type MIRROR for service representing IDS device.
3. Add IDS service of type “ANY”
4. Add content inspection action of type “MIRROR”
5. Add content inspection policy for IDS inspection
6. Bind content inspection policy to content switching or load balancing virtual service of type HTTP/SSL

Enable Content Inspection

If you want the Citrix ADC appliance to send the content for inspection to the IDS devices, you must enable the Content Inspection and load balancing features irrespective of performing decryption.

At the command prompt, type:

```
enable ns feature contentInspection LoadBalancing
```

Add Content Inspection profile of type “MIRROR

The Content Inspection profile of type “MIRROR” explains how you can connect to the IDS device.
At the command prompt, type.

```
add contentInspection profile <name> -type MIRROR -egressInterface <interface_name
> [-egressVlan <positive_integer>]
```

Example:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 10
```

Add IDS service

You must configure a service of type “ANY” for each IDS device that is integrated with the appliance. The service has the IDS device configuration details. The service represents the IDS device.

At the command prompt, type:

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName <
Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

Example:

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
IDS_profile1 -healthMonitor OFF
```

Add content inspection action of type MIRROR for IDS service

After you enable the Content Inspection feature and then add the IDS profile and service, you must add the Content Inspection action for handling the request. Based on the content inspection action, the appliance can drop, reset, block, or send data to the IDS device.

At the command prompt, type:

```
add ContentInspection action < action_name > -type MIRROR -serverName
Service_name/Vserver_name>
```

Example:


```
add ContentInspection action IDS_action -type MIRROR -serverName IDS_service
```

Add content inspection policy for IDS inspection

After you create a Content Inspection action, you must add Content Inspection policies to evaluate requests for inspection. The policy is based on a rule which consists of one or more expressions. The policy evaluates and selects the traffic for inspection based on the rule.

At the command prompt, type the following:

```
add contentInspection policy < policy_name > -rule <Rule> -action <action_name
>
```

Example:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

Bind content inspection policy to content switching or load balancing virtual service of type HTTP/SSL

To receive the web traffic, you must add a load balancing virtual server.
At the command prompt, type:

```
add lb vserver <name> <vserver name>
```

Example:

```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

Bind Content Inspection policy to content switching virtual server or load balancing virtual server of type HTTP/SSL

You must bind the load balancing virtual server or content switching virtual server of type HTTP/SSL to the Content Inspection policy.

At the command prompt, type the following:

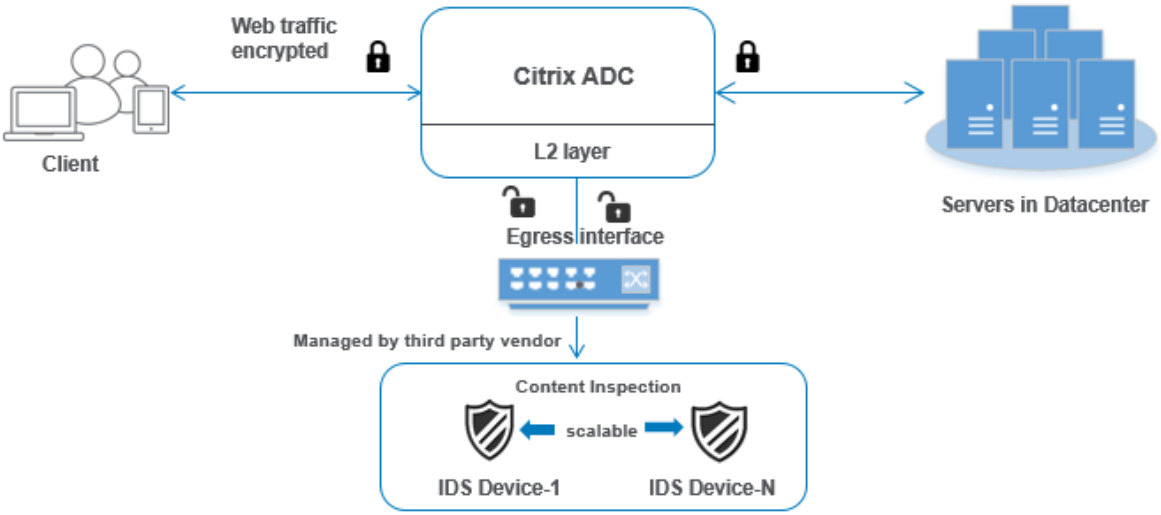
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

Example:

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```


Scenario 2: Load balancing multiple IDS devices

If you are using two or more IDS devices, you must load balance the devices using different content inspection services. In this case, the Citrix ADC appliance load balances the devices on top of sending a subset of traffic to each device.
For basic configuration steps, refer to scenario 1.



Following are the steps you must configure using the command line interface.

1. Add content inspection profile 1 of type MIRROR for IDS service 1
2. Add content inspection profile 2 of type MIRROR for IDS service 2
3. Add IDS service 1 of type ANY for IDS device 1
4. Add IDS service 2 of type ANY for IDS device 2
5. Add load balancing virtual server of type ANY
6. Bind IDS service 1 to load balancing virtual server
7. Bind IDS service 2 to load balancing virtual server
8. Add content inspection action for the load balancing of IDS devices.
9. Add content inspection policy for inspection
10. Add content switching or load balancing virtual server of type HTTP/SSL
11. Bind content inspection policy to load balancing virtual server of type HTTP/SSL

Add content inspection profile1 of type MIRROR for IDS service 1

IDS configuration can be specified in an entity called the Content Inspection profile. The profile has a collection of device settings. The Content Inspection profile1 is created for IDS service 1.

At the command prompt, type:

```
add contentInspection profile <name> -type ANY -egressInterface <interface_name
> [-egressVlan <positive_integer>]
```


Example:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 1
```

Add content inspection profile 2 for type MIRROR for IDS service 2

The Content Inspection profile 2 is added for service 2 and the inline device communicates with the appliance through the egress 1/1 interface.

At the command prompt, type:

```
add contentInspection profile <name> -type MIRROR -egressInterface -egressVlan
 <positive_integer>]
```

Example:

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 1
```

Add IDS service 1 of type ANY for IDS device 1

After you enable the Content Inspection feature and add the inline profile, you must add an inline service 1 for inline device 1 to be part of the load balancing setup. The service that you add, provides all the inline configuration details.

At the command prompt, type:

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileName
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

Example:

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
IDS_profile1 -usip ON -useproxyport OFF
```

> **Note**
>
> The IP address mentioned in the example is a dummy one.

Add IDS service 2 of type ANY for IDS device 2

After you enable the Content Inspection feature and add the inline profile, you must add an inline service 2 for inline device 2. The service that you add, provides all the inline configuration details.

At the command prompt, type:

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName <
Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```


Example:

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```

> **Note**
>
> The IP address mentioned in the example is a dummy one.

Add load balancing virtual server

After you have added the inline profile and the services, you must add a load balancing virtual server for load balancing the services.

At the command prompt, type:

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

Example:

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

Bind IDS service 1 to load balancing virtual server

After you add the load balancing virtual server, now bind the load balancing virtual server to the first service.

At the command prompt, type:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

Example:

```
bind lb vserver lb-IDS_vserver IDS_service1
```

Bind IDS service 2 to load balancing virtual server

After you add the load balancing virtual server, now bind the server to the second service.

At the command prompt, type:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

Example:

```
bind lb vserver lb-IDS_vserver IDS_service2
```


Add content inspection action for the IDS service

After you enable the Content Inspection feature, you must add the Content Inspection action for handling the inline request information. Based on the action selected, the appliance drops, resets, blocks, or sends traffic to the IDS device.

At the command prompt, type:

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>]
```

Example:

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-IDS_vserver
```

Add content inspection policy for inspection

After you create a Content Inspection action, you must add a Content Inspection policy to evaluate requests for service.

At the command prompt, type the following:

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

Example:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

Add content switching or load balancing virtual server of type HTTP/SSL

Add a content switching or load balancing virtual server to accept web traffic. Also you must enable the layer2 connection on the virtual server.

For more information about load balancing, refer to How load balancing works topic.

At the command prompt, type:

```
add lb vserver <name> <vserver name>
```

Example:

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

Bind Content Inspection policy to load balancing virtual server of type HTTP/SSL

You must bind the content switching or load balancing virtual server of type HTTP/SSL to the Content Inspection policy.

At the command prompt, type the following:

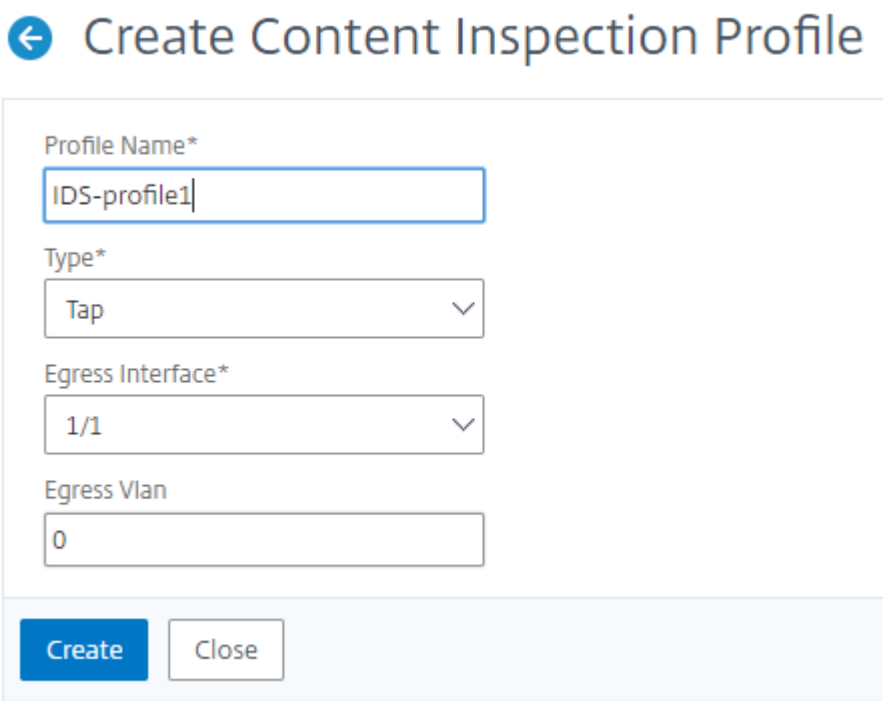
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <REQUEST>
```

Example:

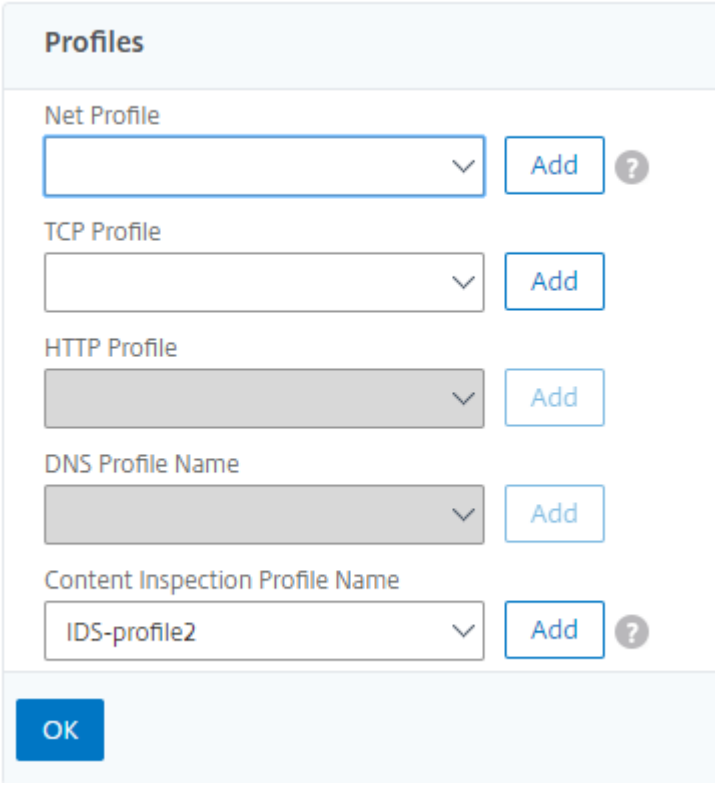
```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

Configure inline service integration using the Citrix ADC GUI

1. Navigate to **Security** > **Content Inspection** > **Content Inspection Profiles**.
2. In the **Content Inspection Profile** page, click **Add**.
3. In the **Create Content Inspection Profile** page, set the following parameters.
   a) Profile Name. Name of the content inspection profile for IDS.
   b) Type. Select the profile types as MIRROR.
   c) Egress Interface. The interface through which the traffic is sent from the Citrix ADC to the IDS device.
   d) Egress VLAN (optional). The interface VLAN ID through which the traffic is sent to the IDS device.
4. Click **Create**.



5. Navigate to **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
6. In the **Load Balancing Service** page, enter the content inspection service details.
7. In the **Advanced Settings** section, click **Profiles**.
8. Go to the **Profiles** section and click the **Pencil** icon to add the content inspection profile.
9. Click **OK**.

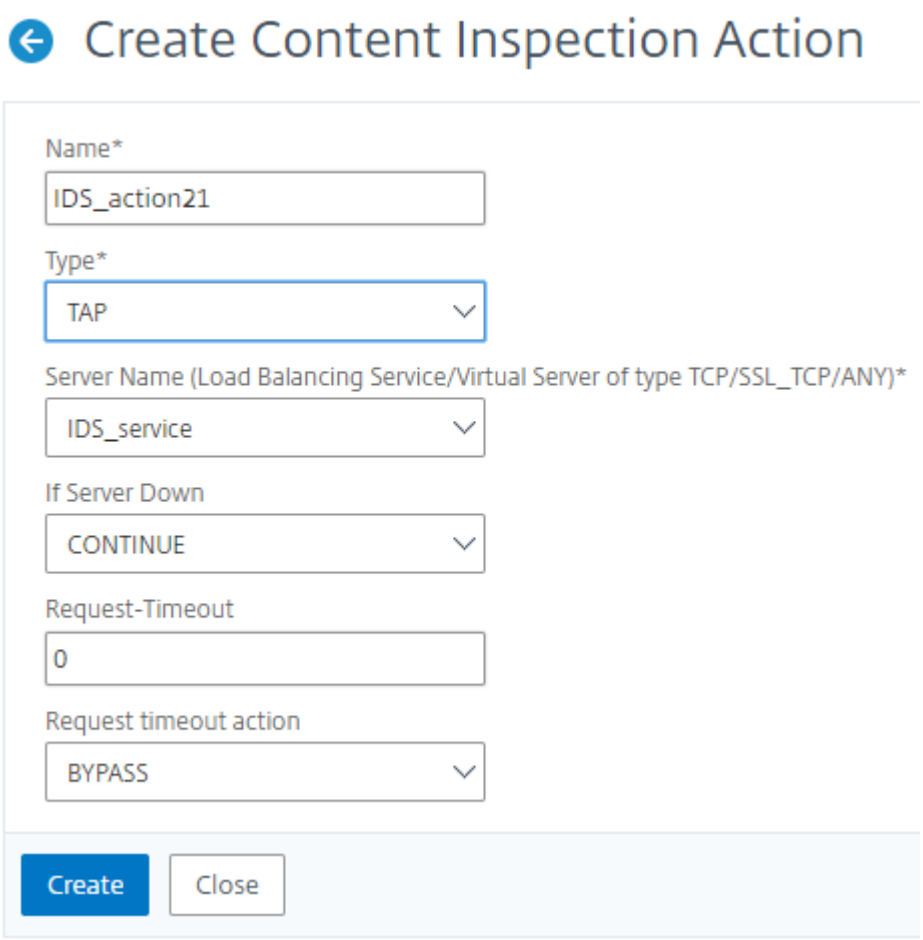


10. Navigate to **Load Balancing** > **Servers**. Add a virtual server of type HTTP or SSL.
11. After entering the server details, click **OK** and again **OK**.
12. In the **Advanced Settings** section, click **Policies**.
13. Go the **Policies** section and click the **Pencil** icon to configure the content inspection policy.
14. On the **Choose Policy** page, select **Content Inspection**. Click **Continue**.
15. In the **Policy Binding** section, click "+" to add a Content Inspection policy.
16. In the **Create CI Policy** page, enter a name for the Inline content inspection policy.
17. In the **Action** field, click the "+" sign to create an IDS content inspection action of type MIRROR.
18. In the **Create CI Action** page, set the following parameters.

    a. Name. Name of the content inspection Inline policy.

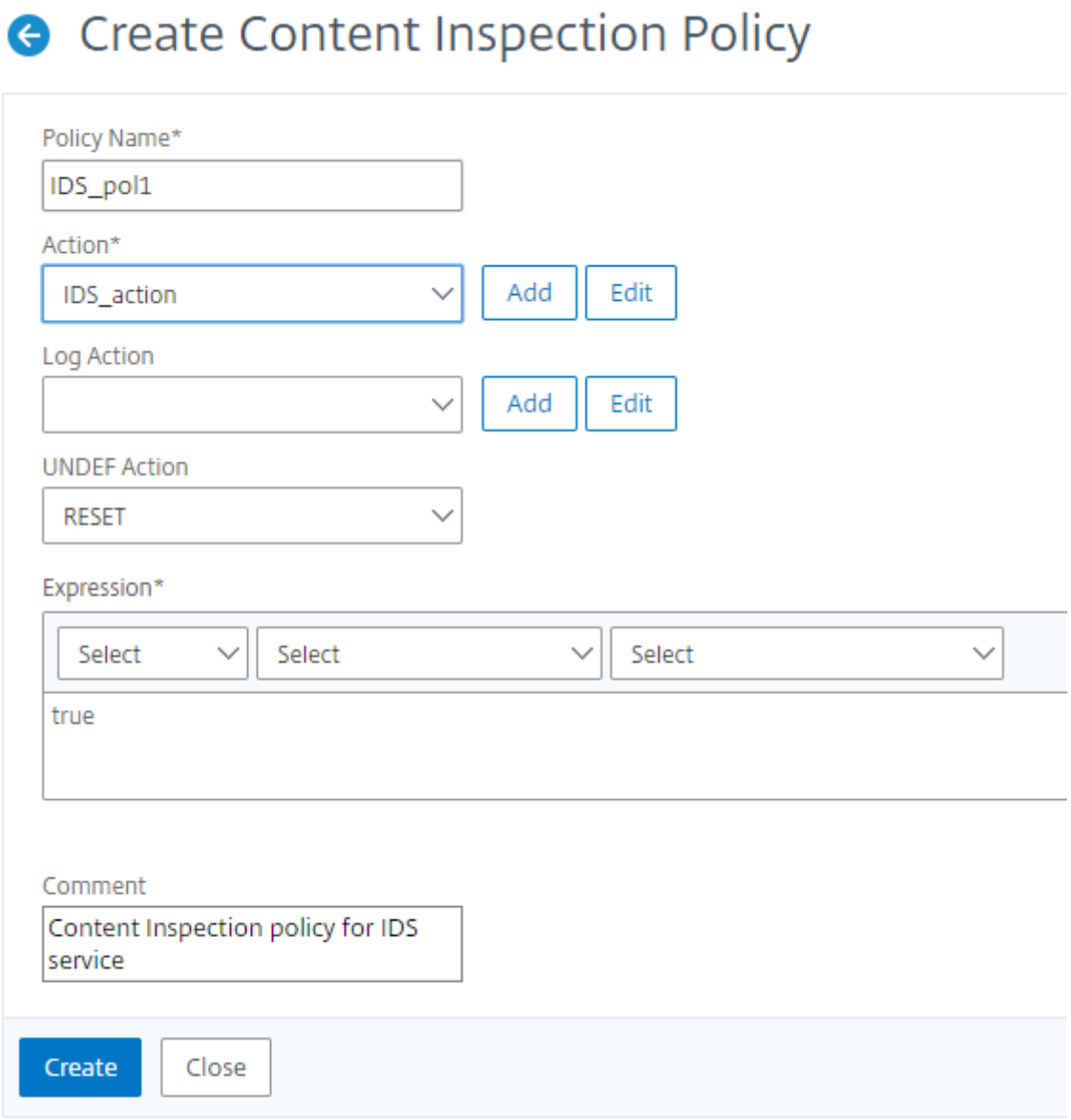
b. Type. Select the type as MIRROR.
c. Server Name. Select the server/service name as Inline devices.
d. If Server Down. Select an operation if the server goes down.
e. Request Time-out. Select a time-out value. Default values can be used.
f. Request Time-out Action. Select a time-out action. Default values can be used.

19. Click **Create**.



20. In the **Create CI Policy** page, enter other details.
21. Click **OK** and **Close**.

For information about the Citrix ADC GUI configuration for load balancing and replicating the traffic to IDS devices, see Load Balancing.



For information about the Citrix ADC GUI configuration for load balancing and forwarding the traffic to the back-end origin server after content transformation, see Load Balancing topic.

Integrating Citrix ADC layer 3 with passive security devices (Intrusion Detection System)

September 14, 2021

A Citrix ADC appliance is now integrated with passive security devices such as the Intrusion Detection System (IDS). In this setup, the appliance sends a copy of the original traffic securely to remote IDS

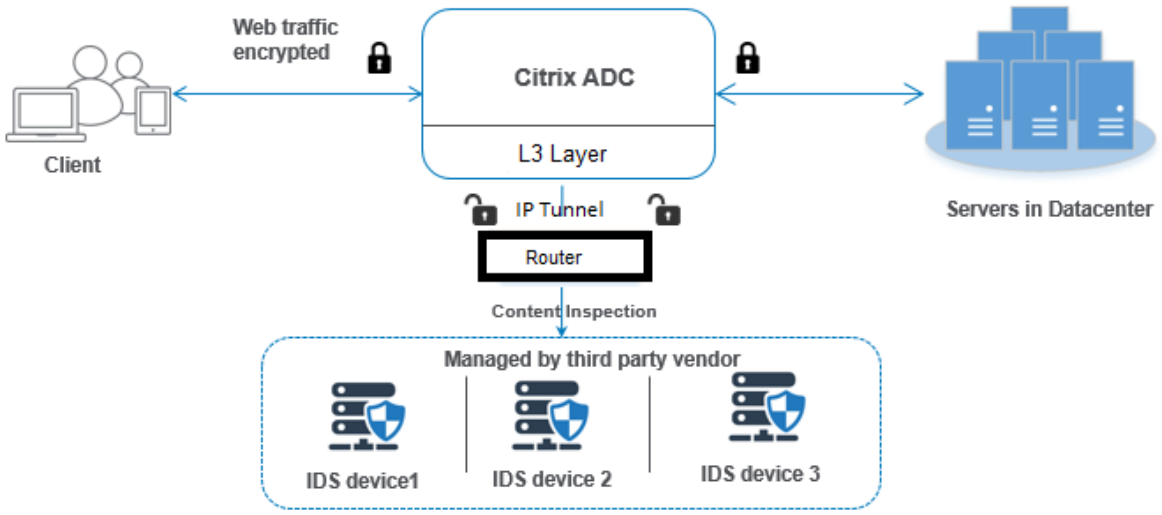
devices. These passive devices store logs and trigger alerts when it detects a bad or non-compliant traffic. It also generates reports for the compliance purpose. If a Citrix ADC appliance is integrated with two or more IDS devices and when there is a high volume of traffic, the appliance can load balance the devices by cloning traffic at the virtual server level.

For advanced security protection, a Citrix ADC appliance is integrated with passive security devices such as IDS deployed in detection-only mode. These devices store log and trigger alerts when it sees a bad or non-compliant traffic. It also generates reports for the compliance purpose. Following are some of the benefits of integrating the Citrix ADC with an IDS device.

- **Inspecting encrypted traffic**. Most security devices bypass encrypted traffic, thereby leaving servers vulnerable to attacks. A Citrix ADC appliance can decrypt traffic and send it to IDS devices for enhancing the customer's network security.
- **Offloading inline devices from TLS/SSL processing**. TLS/SSL processing is expensive and it results in high system CPU in intrusion detection devices if they decrypt the traffic. As encrypted traffic is growing at a fast pace, these systems fail to decrypt and inspect encrypted traffic. Citrix ADC helps in offloading traffic to IDS devices from TLS/SSL processing. This way of offloading data results in an IDS device supporting a high volume of traffic inspection.
- **Loading balancing IDS devices**. The Citrix ADC appliance load balances multiple IDS devices when there is a high volume of traffic by cloning traffic at the virtual server level.
- **Replicating traffic to passive devices**. The traffic flowing into the appliance can be replicated to other passive devices for generating compliance reports. For example, few government agencies mandate every transaction to be logged in some passive devices.
- **Fanning traffic to multiple passive devices**. Some customers prefer to fan out or replicate incoming traffic into multiple passive devices.
- **Smart selection of traffic**. Every packet flowing into the appliance might not be must be content inspected, for example download of text files. User can configure the Citrix ADC appliance to select specific traffic (for example .exe files) for inspection and send the traffic to IDS devices for processing data.

How Citrix ADC is integrated with IDS device with L3 connectivity

The following diagram shows how the IDS is integrated with a Citrix ADC appliance.



The component interaction is given as follows:

1. A client sends an HTTP/HTTPS request to the Citrix ADC appliance.
2. The appliance intercepts the traffic and sends the data to remote IDS devices across different data centers or even in a cloud. This integration is done through IP tunneled layer 3. For more information about IP tunneling in a Citrix ADC appliance, see IP tunnels topic.
3. If the traffic is an encrypted one, the appliance decrypts the data and sends it as a plain text.
4. Based on policy evaluation, the appliance applies a “MIRROR” type content inspection action.
5. The action has an IDS service or load balancing service (for multiple IDS device integrations) configured in it.
6. The IDS device is configured as content inspection service type “Any” on the appliance. The content inspection service is then associated to the content inspection profile of type “MIRROR” and the tunnel parameter which specifies the IP tunneled layer 3 interface through which the data is forwarded to the IDS device.

> **Note** Optionally, you can also configure a VLAN tag in the content inspection profile.

1. Similarly, when the back-end server sends a response to the Citrix ADC, the appliance replicates the data and forwards it to the IDS device.
2. If your appliance is integrated to one or more IDS devices and if you prefer to load balance the devices, then you can use the load balancing virtual server.

Software licensing

To deploy the IDS integration, your Citrix ADC appliance must be provisioned with one of the following licenses:

1. ADC Premium
2. ADC Advanced

Configuring intrusion detection system integration

You can integrate IDS device with a Citrix ADC in two different ways.

Scenario 1: Integration with a single IDS device

Following are the steps you must configure using the command line interface.

1. Enable content inspection
2. Add content inspection profile of type MIRROR for service representing IDS device.
3. Add IDS service of type “ANY”
4. Add content inspection action of type “MIRROR”
5. Add content inspection policy for IDS inspection
6. Bind content inspection policy to content switching or load balancing virtual service of type HTTP/SSL

Enable Content Inspection

If you want the Citrix ADC appliance to send the content for inspection to the IDS devices, you must enable the Content Inspection and load balancing features irrespective of performing decryption.

At the command prompt, type:

```
enable ns feature contentInspection LoadBalancing
```

Add Content Inspection profile of type “MIRROR”

The Content Inspection profile of type “MIRROR” explains how you can connect to the IDS device.
At the command prompt, type.

> **Note**
> The IP tunnel parameter must be used only for layer 3 IDS topology. Otherwise, you must use the egress interface with the egress VLAN option.

```
add contentInspection profile <name> -type MIRROR -ipTunnel <iptunnel_name>
```

Example:

```
add contentInspection profile IDS_profile1 -type MIRROR -ipTunnel ipsect-
tunnel1
```

Add IDS service

You must configure a service of type “ANY” for each IDS device that is integrated with the appliance. The service has the IDS device configuration details. The service represents the IDS device.

At the command prompt, type:

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName <
Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

Example:

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
IDS_profile1 -healthMonitor OFF
```

Add content inspection action of type MIRROR for IDS service

After you enable the Content Inspection feature and then add the IDS profile and service, you must add the Content Inspection action for handling the request. Based on the content inspection action, the appliance can drop, reset, block, or send data to the IDS device.

At the command prompt, type:

```
add ContentInspection action < action_name > -type MIRROR -serverName
Service_name/Vserver_name>
```

Example:

```
add ContentInspection action IDS_action -type MIRROR -serverName IDS_service
```

Add content inspection policy for IDS inspection

After you create a Content Inspection action, you must add Content Inspection policies to evaluate requests for inspection. The policy is based on a rule which consists of one or more expressions. The policy evaluates and selects the traffic for inspection based on the rule.

At the command prompt, type the following:

```
add contentInspection policy < policy_name > -rule <Rule> -action <action_name
>
```

Example:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

Bind content inspection policy to content switching or load balancing virtual service of type HTTP/SSL

To receive the web traffic, you must add a load balancing virtual server.
At the command prompt, type:

```
add lb vserver <name> <vserver name>
```

Example:


```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

Bind Content Inspection policy to content switching virtual server or load balancing virtual server of type HTTP/SSL

You must bind the load balancing virtual server or content switching virtual server of type HTTP/SSL to the Content Inspection policy.

At the command prompt, type the following:

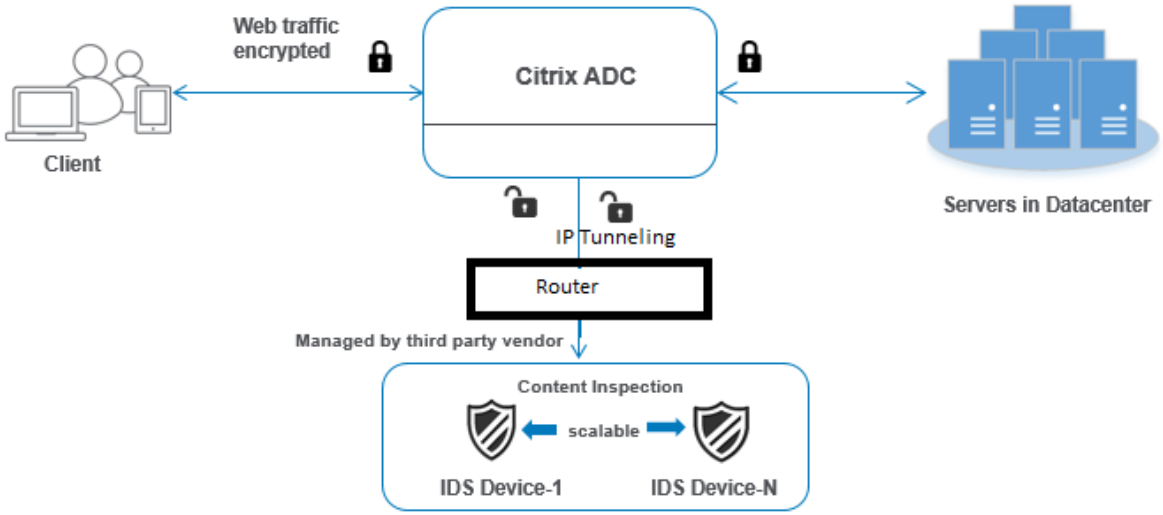
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

Example:

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

Scenario 2: Load balancing multiple IDS devices

If you are using two or more IDS devices, you must load balance the IDS devices using different content inspection services. In this case, the Citrix ADC appliance load balances the devices on top of sending a subset of traffic to each device.
For basic configuration steps, refer to scenario 1.



Following are the steps you must configure using the command line interface.

1. Add content inspection profile 1 of type MIRROR for IDS service 1
2. Add content inspection profile 2 of type MIRROR for IDS service 2
3. Add IDS service 1 of type ANY for IDS device 1

4. Add IDS service 2 of type ANY for IDS device 2
5. Add load balancing virtual server of type ANY
6. Bind IDS service 1 to load balancing virtual server
7. Bind IDS service 2 to load balancing virtual server
8. Add content inspection action for the load balancing of IDS devices.
9. Add content inspection policy for inspection
10. Add content switching or load balancing virtual server of type HTTP/SSL
11. Bind content inspection policy to load balancing virtual server of type HTTP/SSL

Add content inspection profile1 of type MIRROR for IDS service 1

IDS configuration can be specified in an entity called the Content Inspection profile. The profile has a collection of device settings. The Content Inspection profile1 is created for IDS service 1.

> **Note**:
> IP tunnel parameter must be used only for layer 3 IDS topology. Otherwise, you must use the egress interface with the egress VLAN option.

At the command prompt, type:

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name>
```

Example:

```
add contentInspection profile IDS_profile1 -type MIRROR - ipTunnel ipsect_tunnel1
```

Add content inspection profile 2 for type MIRROR for IDS service 2

The Content Inspection profile 2 is added for service 2 and the inline device communicates with the appliance through the egress 1/1 interface.

At the command prompt, type:

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name>
```

Example:

```
add contentInspection profile IDS_profile2 -type ANY - ipTunnel ipsect_tunnel2
```

Add IDS service 1 of type ANY for IDS device 1

After you enable the Content Inspection feature and add the inline profile, you must add an inline service 1 for inline device 1 to be part of the load balancing setup. The service that you add, provides all the inline configuration details.

At the command prompt, type:

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileName
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

Example:

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
IDS_profile1 -usip ON -useproxyport OFF
```

> **Note**:
>
> The IP address mentioned in the example is a dummy one.

Add IDS service 2 of type ANY for IDS device 2

After you enable the Content Inspection feature and add the inline profile, you must add an inline service 2 for inline device 2. The service that you add, provides all the inline configuration details.

At the command prompt, type:

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName <
Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

Example:

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```

> **Note**:
>
> The IP address mentioned in the example is a dummy one.

Add load balancing virtual server

After you have added the inline profile and the services, you must add a load balancing virtual server for load balancing the services.

At the command prompt, type:

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

Example:

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

Bind IDS service 1 to load balancing virtual server

After you add the load balancing virtual server, now bind the load balancing virtual server to the first service.

At the command prompt, type:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

Example:

```
bind lb vserver lb-IDS_vserver IDS_service1
```

Bind IDS service 2 to load balancing virtual server

After you add the load balancing virtual server, now bind the server to the second service.

At the command prompt, type:

```
bind lb vserver <Vserver_name> <Service_name_1>
```

Example:

```
bind lb vserver lb-IDS_vserver IDS_service2
```

Add content inspection action for the IDS service

After you enable the Content Inspection feature, you must add the Content Inspection action for handling the inline request information. Based on the action selected, the appliance drops, resets, blocks, or sends traffic to the IDS device.

At the command prompt, type:

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>]
```

Example:

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-IDS_vserver
```

Add content inspection policy for inspection

After you create a Content Inspection action, you must add the Content Inspection policy to evaluate requests for service.

At the command prompt, type the following:

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

Example:

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```


Add content switching or load balancing virtual server of type HTTP/SSL

Add a content switching or load balancing virtual server to accept web traffic. Also you must enable the layer2 connection on the virtual server.

For more information about load balancing, refer to How load balancing works topic.

At the command prompt, type:

```
add lb vserver <name> <vserver name>
```

Example:

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

Bind Content Inspection policy to load balancing virtual server of type HTTP/SSL

You must bind the content switching or load balancing virtual server of type HTTP/SSL to the Content Inspection policy.

At the command prompt, type the following:

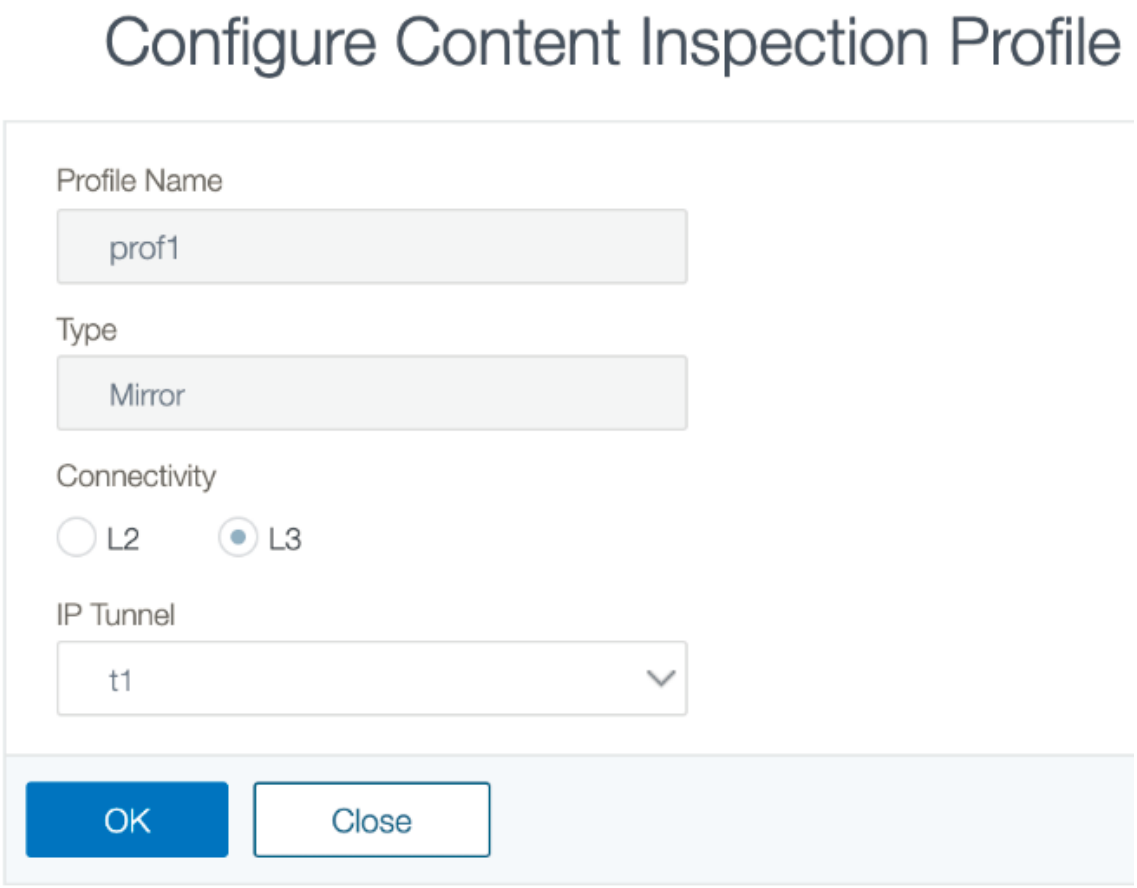
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <REQUEST>
```

Example:

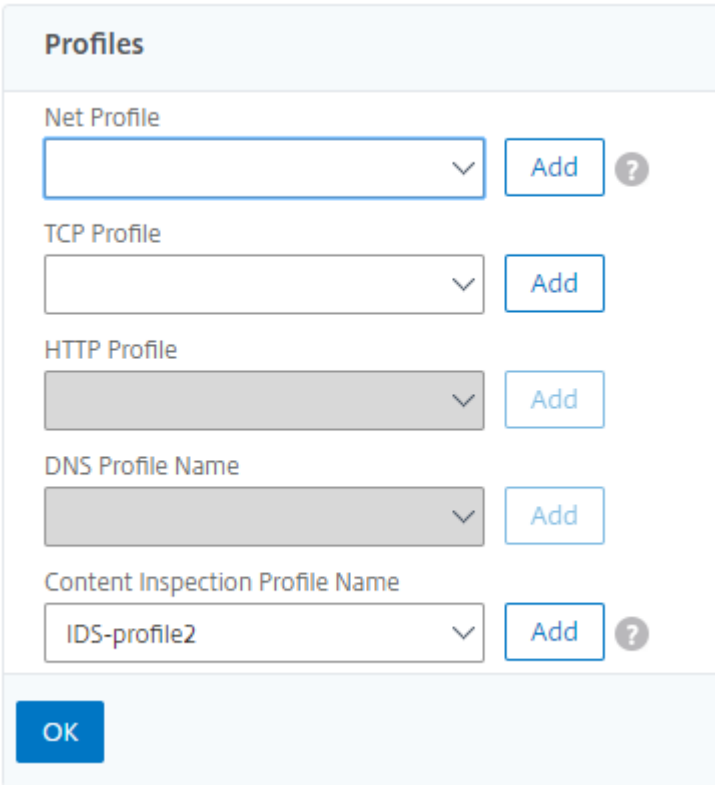
```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

Configure inline service integration using the Citrix ADC GUI

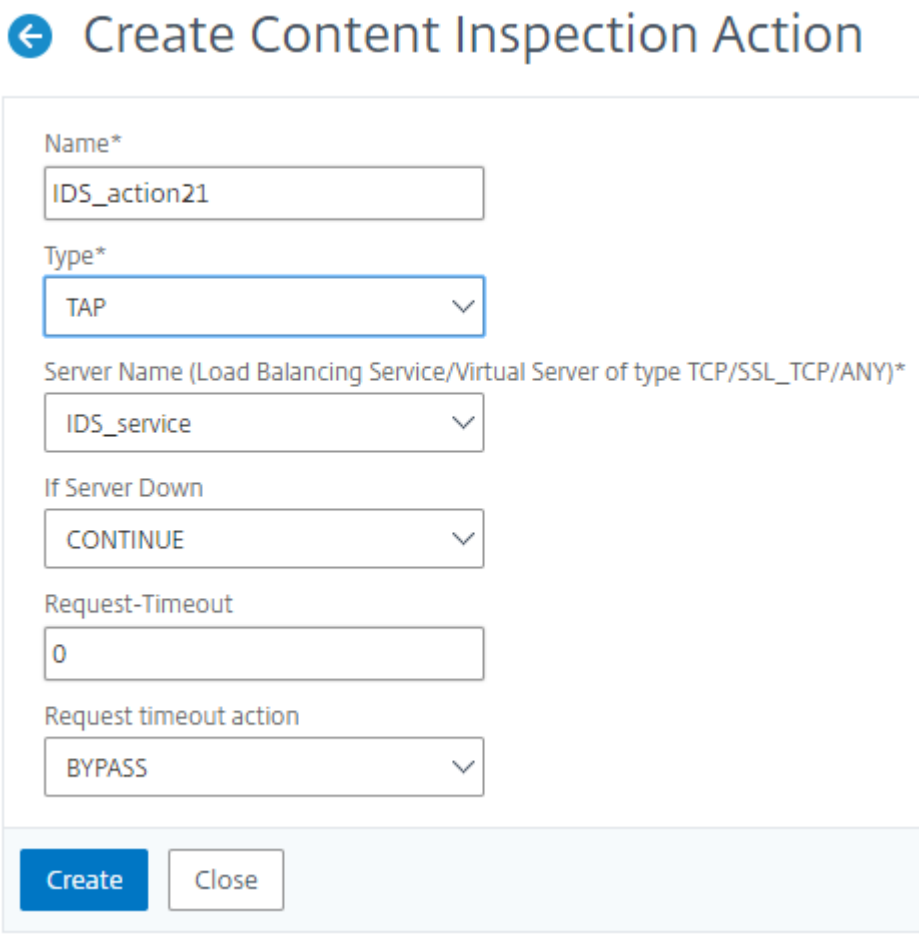
1. Navigate to **Security** > **Content Inspection** > **ContentInspection Profiles**.
2. In the **ContentInspection Profile** page, click **Add**.
3. In the **Create ContentInspectionProfile** page, set the following parameters.
    a) Profile Name. Name of the content inspection profile for IDS.
    b) Type. Select the profile types as MIRROR.
    c) Connectivity. Layer 2 or Layer 3 interface.
    d) IP Tunnel. Select the network communication channel between the two networks.
4. Click **Create**.



5. Navigate to **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
6. In the **Load Balancing Service** page, enter the content inspection service details.
7. In the **Advanced Settings** section, click **Profiles**.
8. Go to the **Profiles** section and click the **Pencil** icon to add the content inspection profile.
9. Click **OK**.

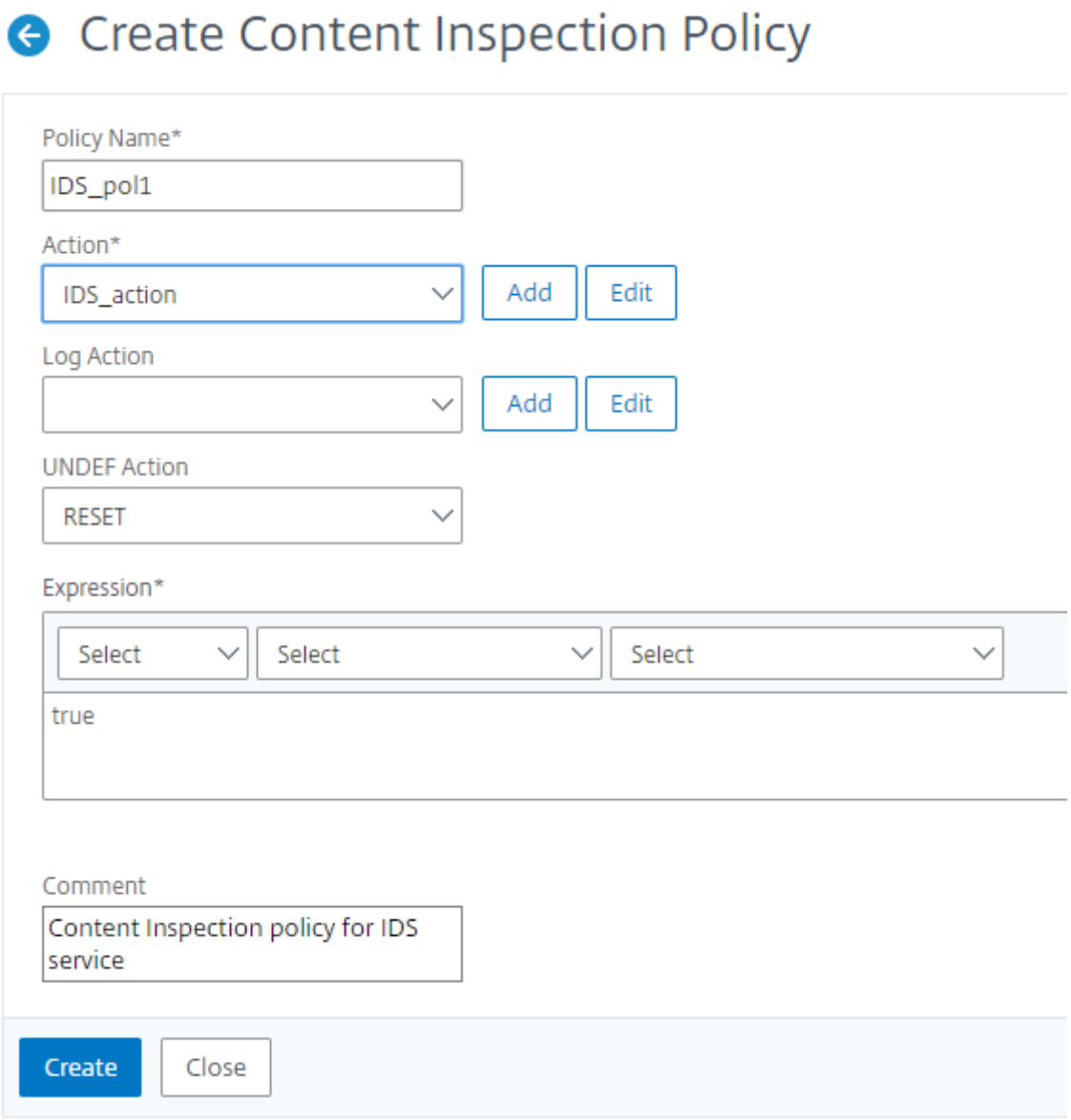


10. Navigate to **Load Balancing** > **Servers**. Add a virtual server of type HTTP or SSL.
11. After entering the server details, click **OK** and again **OK**.
12. In the **Advanced Settings** section, click **Policies**.
13. Go the **Policies** section and click the **Pencil** icon to configure the content inspection policy.
14. On the **Choose Policy** page, select **Content Inspection**. Click **Continue**.
15. In the **Policy Binding** section, click "+" to add a Content Inspection policy.
16. In the **Create CI Policy** page, enter a name for the Inline content inspection policy.
17. In the **Action** field, click the "+" sign to create an IDS content inspection action of type MIRROR.
18. In the **Create CI Action** page, set the following parameters.
    a) Name. Name of the content inspection Inline policy.
    b) Type. Select the type as MIRROR.
    c) Server Name. Select the server/service name as Inline devices.
    d) If Server Down. Select an operation if the server goes down.
    e) Request Time-out. Select a time-out value. Default values can be used.
    f) Request Time-out Action. Select a time-out action. Default values can be used.
19. Click **Create**.



20. In the **Create CI Policy** page, enter other details.
21. Click **OK** and **Close**.

For information about the Citrix ADC GUI configuration for load balancing and replicating the traffic to IDS devices, see Load Balancing.



For information about the Citrix ADC GUI configuration for load balancing and forwarding the traffic to the back-end origin server after content transformation, see Load Balancing.

Content inspection statistics for ICAP, IPS, and IDS

September 14, 2021

The content inspection statistics for ICAP, inline device integration (IDS), and Intrusion Prevention System (IPS) devices is a detailed output (summary) of request, response, and server action details.

The content inspection statistics is a collection of statistical data that includes the HTTP/HTTPS re-

quest sent for content inspection. HTTP/HTTPS response received from IPS, IDS, and ICAP devices and back-end server action.

To display Content inspection statistics by using the CLI:

At the command prompt, type:

```
> stat contentInspection
```

```
ContentInspection Stats

Inline Statistics
                                    Total
Requests                             10
Responses                             6
Request Bytes Sent                   3235
Request Bytes Received               2977
Response Bytes Sent                  17302
Response Bytes Received              19681
Serverdown Reset Action taken         1
Serverdown Drop Action taken          0
Serverdown BYPASS Action taken        0
Inline device Generated Response      3

Mirror Statistics
                                    Total
Requests                             4
Responses                            4
Requests Bytes Sent                 2763
Responses Bytes Sent                16732
Serverdown Reset Action taken        0
Serverdown Drop Action taken         0
Serverdown BYPASS Action taken       1

ICAP Statistics
                                    Total
REQMOD requests Sent                   6
RESPMOD requests Sent                  4
Preview requests                       1
204 Responses Received                 6
100 Continue Responses Received        1
204 NO content Received                5
Adaptive Requests                      0
Adaptive Responses                     4
Callout requests Initiated             1
Callout requests completed             1
```



```
38 ICAP Req/Resp Errors handled         1
39 Serverdown Reset Action taken        1
40 Serverdown Drop Action taken         0
41 Serverdown BYPASS Action taken       1
42
43 Done
44 <!--NeedCopy-->
```

SSL forward proxy

September 14, 2021

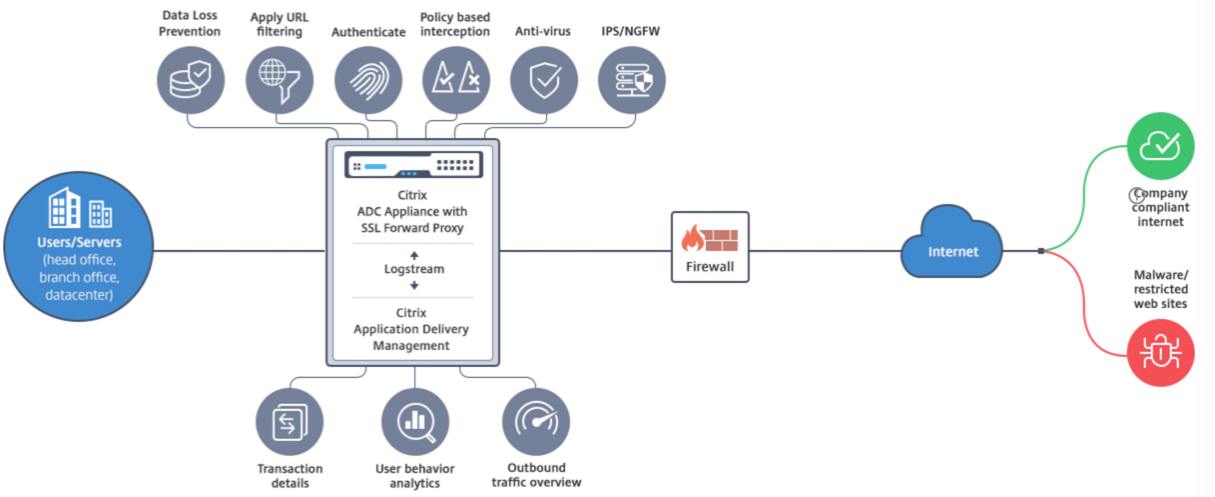
Note: SSL forward proxy feature is available with the ADC Premium license.

Web traffic has increased exponentially in recent years, and corporations are increasingly relying on the internet for their day to day operations. That, combined with the emergence of more diverse endpoints, mobility, and BYOD, along with a growing attacker base, is making users easy targets of modern malware. They are increasingly vulnerable to identity theft and having their data compromised. Traditionally, enterprises have inspected HTTP traffic for malware and viruses. They bypassed HTTPS/TLS traffic, because it was not as prominent. It was used sparingly for content that was sensitive and trusted. But that has changed rapidly as most public internet websites now prefer to use HTTPS to protect user privacy. As a result, the inability to inspect encrypted packets allows malware or intrusions into the enterprise network. The SSL forward proxy solution offers tools that enterprises can use to protect against internet threats.

A proxy is a server that controls all the traffic between users and the Internet or SaaS applications. Since all the traffic passes through this proxy, it performs security-related functions, such as user authentication and URL categorization.

The following figure is an overview of the SSL forward proxy implementation. Traffic flows through the enterprise network from the head office, branch offices, data center, and remote employees. A Citrix ADC appliance at the edge of the network acts a proxy. The appliance can operate in transparent proxy mode or explicit proxy mode and offers controls to intercept internet traffic, including HTTPS. Policies configured on the appliance determine whether it intercepts, bypasses, or blocks a particular request. Access to restricted sites can be blocked by using URL filtering. A user is authenticated before logging on to the enterprise network. All requests and responses are tagged to identify the user, and internet-site access is categorized. User activity is logged and used to generate reports. If a breach occurs, administrators can isolate the infected system, determine whether the devices of any other users who visited that website are compromised, and take appropriate action. When you integrate Citrix Application Delivery Management (ADM) with SSL forward proxy, the logged user activity and the subsequent records in the appliance are exported to Citrix ADM by using `logstream`. Citrix ADM

collates and presents information about the activities of users, from websites visited to the time spent online. It also provides information about bandwidth use and detected threats, such as malware and phishing sites. You can use these key metrics to monitor your network, and use the SSL forward proxy feature to take corrective actions.



SSL forward proxy enables IT directors to do the following:

- Gain visibility into the otherwise bypassed secure traffic.
- Block access to malicious or unknown sites and avoid infecting users within the enterprise.
- Control access to some websites, such as personal mail, social networking, and job search websites, from the enterprise network.
- Apply intelligent content control policies to ensure maximum user productivity.

Getting started with the SSL forward proxy feature

September 14, 2021

Important:

- OCSP check requires an internet connection to check the validity of certificates. If your appliance is not accessible from the internet by using the NSIP address, add access control lists (ACLs) to perform NAT from the NSIP address to the subnet IP (SNIP) address. The SNIP must be able to access the internet. For example,

```
1  add ns acl a1 ALLOW -srcIP = <NSIP> -destIP  "!="
       10.0.0.0-10.255.255.255
2
3  add rnat RNAT-1 a1
4
5  bind rnat RNAT-1 -<SNIP>
```



```
6
7   apply acls
8   <!--NeedCopy-->
```

- Specify a DNS name server to resolve domain names.
- Make sure that the date on the appliance is synchronized with the NTP servers. If the date is not synchronized, the appliance cannot effectively verify whether an origin server certificate is an expired one.

To use the SSL forward proxy feature, you must perform the following tasks:

- Add a proxy server in explicit or transparent mode.
- Enable SSL interception.
  - Configure an SSL profile.
  - Add and bind SSL policies to the proxy server.
  - Add and bind a CA certificate-key pair for SSL interception.

Note:

An ADC appliance configured in transparent proxy mode can intercept only HTTP and HTTPS protocols. To bypass any other protocol, such as telnet, you must add the following listen policy on the proxy virtual server.

The virtual server now accepts only HTTP and HTTPS incoming traffic.

```
1 set cs vserver transparent-pxy1 PROXY * * -cltTimeout 180 -Listenpolicy
      "CLIENT.TCP.DSTPORT.EQ(80) || CLIENT.TCP.DSTPORT.EQ(443)"`
2 <!--NeedCopy-->
```

You might need to configure the following features, depending on your deployment:

- Authentication Service (recommended) – to authenticate users. Without the Authentication Service, user activity is based on client IP address.
- URL Filtering – to filter URLs by categories, reputation score, and URL lists.
- Analytics – to view user activity, user risk indicators, bandwidth consumption, and transactions break down in Citrix Application Delivery Management (ADM).

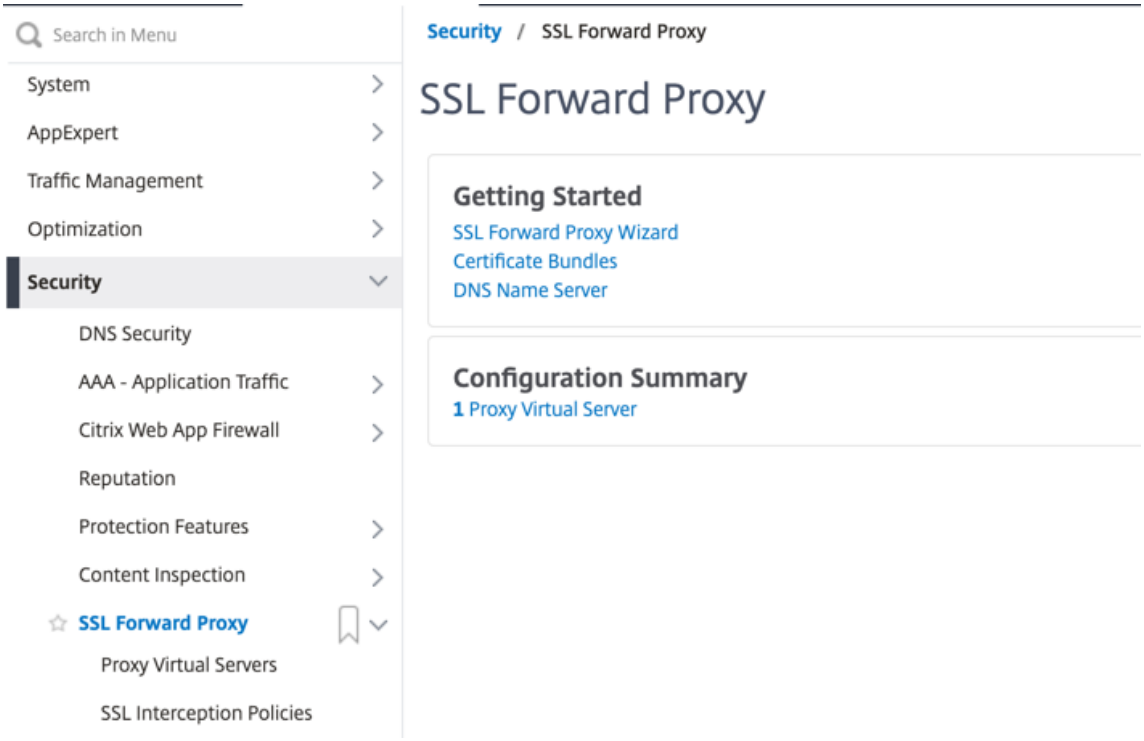
Note: SSL Forward Proxy implements most typical HTTP and HTTPS standards followed by similar products. This implementation is done with no specific browser in mind and is compatible with most common browsers. SSL Forward Proxy has been tested with common browsers and recent versions of Google Chrome, Internet Explorer, and Mozilla Firefox.

SSL forward proxy wizard

The SSL forward proxy wizard provides administrators with a tool for managing the entire SSL forward proxy deployment by using a web browser. It helps guide the customers to bring up an SSL forward

proxy service quickly and helps simplify the configuration by following a sequence of well-defined steps.

1. Navigate to **Security > SSL Forward Proxy**. In **Getting Started**, click **SSL Forward Proxy Wizard**.



2. Follow the steps in the wizard to configure your deployment.

Add a listen policy to the transparent proxy server

1. Navigate to **Security >SSL Forward Proxy** > **Proxy Virtual Servers**. Select the transparent proxy server and click **Edit**.
2. Edit **Basic Settings**, and click **More**.
3. In **Listen priority**, enter 1.
4. In **Listen Policy Expression**, enter the following expression:

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

This expression assumes standard ports for HTTP and HTTPS traffic. If you have configured different ports, for example 8080 for HTTP or 8443 for HTTPS, modify the expression to reflect those ports.

Limitations

SSL forward proxy is not supported in a cluster setup, in admin partitions, and on a Citrix ADC FIPS appliance.

Proxy modes

September 14, 2021

The Citrix ADC appliance acts as a client's proxy to connect to the internet and SaaS applications. As a proxy, it accepts all the traffic and determines the traffic's protocol. Unless the traffic is HTTP or SSL, it is forwarded to the destination as is. When the appliance receives a request from a client, it intercepts the request and performs some actions, such as user authentication, site categorization, and redirection. It uses policies to determine which traffic to allow and which traffic to block.

The appliance maintains two different sessions, one between the client and the proxy and the other between the proxy and the origin server. The proxy relies on customer defined policies to allow or block HTTP and HTTPS traffic. Therefore, it is important that you define policies to bypass sensitive data, such as financial information. The appliance offers a rich set of Layer 4 to Layer 7 traffic attributes and user-identity attributes to create traffic management policies.

For SSL traffic, the proxy verifies the origin server's certificate and establishes a legitimate connection with the server. It then emulates the server certificate, signs it using a CA certificate installed on Citrix ADC, and presents the created server certificate to the client. You must add the CA certificate as a trusted certificate to the client's browser for the SSL session to be successfully established.

The appliance supports transparent and explicit proxy modes. In explicit proxy mode, the client must specify an IP address in their browser, unless the organization pushes the setting onto the client's device. This address is the IP address of a proxy server that is configured on the ADC appliance. All client requests are sent to this IP address. For explicit proxy, you must configure a content switching virtual server of type PROXY and specify an IP address and a valid port number.

Transparent proxy, as the name implies, is transparent to the client. That is, the clients might not be aware that a proxy server is mediating their requests. The ADC appliance is configured in an inline deployment, and transparently accepts all HTTP and HTTPS traffic. For transparent proxy, you must configure a content switching virtual server of type PROXY, with asterisks (* *) as the IP address and port. When using the **SSL Forward Proxy Wizard** in the GUI, you do not have to specify an IP address and port.

> **Note**
>
> To intercept protocols other than HTTP and HTTPS in transparent proxy mode, you must add a

> listen policy and bind it to the proxy server.

Configure SSL forward proxy by using the CLI

At the command prompt, type:

```
add cs vserver <name> PROXY <ipaddress> <port>
<!--NeedCopy-->
```

Arguments:

Name:

Name for the proxy server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CS virtual server is created.

The following requirement applies only to the CLI:

If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my server” or ‘my server’).

This argument is mandatory. Maximum Length: 127

IPAddress:

IP address of the proxy server.

Port:

Port number for the proxy server. Minimum value: 1

Example for explicit proxy:

```
add cs vserver swgVS PROXY 192.0.2.100 80
<!--NeedCopy-->
```

Example for transparent proxy:

```
add cs vserver swgVS PROXY * *
<!--NeedCopy-->
```

Add a listen policy to the transparent proxy server by using the GUI

1. Navigate to **Security** > **SSL Forward Proxy** > **Proxy Virtual Servers**. Select the transparent proxy server and click **Edit**.
2. Edit **Basic Settings**, and click **More**.

3. In **Listen priority**, enter 1.
4. In **Listen Policy Expression**, enter the following expression:

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

> **Note**
>
> This expression assumes standard ports for HTTP and HTTPS traffic. If you have configured different ports, for example 8080 for HTTP or 8443 for HTTPS, modify the preceding expression to specify those ports.

SSL interception

September 14, 2021

A Citrix ADC appliance configured for SSL interception acts as a proxy. It can intercept and decrypt SSL/TLS traffic, inspect the unencrypted request, and enable an admin to enforce compliance rules and security checks. SSL interception uses a policy that specifies which traffic to intercept, block, or allow. For example, traffic to and from financial websites, such as banks, must not be intercepted, but other traffic can be intercepted, and blacklisted sites can be identified and blocked. Citrix recommends that you configure one generic policy to intercept traffic and more specific policies to bypass some traffic.

The client and the proxy establish an HTTPS/TLS handshake. The proxy establishes another HTTPS/TLS handshake with the server and receives the server certificate. The proxy verifies the server certificate on behalf of the client, and also checks the validity of the server certificate by using the Online Certificate Status Protocol (OCSP). It regenerates the server certificate, signs it by using the key of the CA certificate installed on the appliance, and presents it to the client. Therefore, one certificate is used between the client and the Citrix ADC appliance, and another certificate between the appliance and the back-end server.

> **Important**
>
> The CA certificate that is used to sign the server certificate must be preinstalled on all the client devices, so that the regenerated server certificate is trusted by the client.

For intercepted HTTPS traffic, the proxy server decrypts the outbound traffic, accesses the clear text HTTP request, and can use any Layer 7 application to process the traffic, such as by looking into the plain text URL and allowing or blocking access based on the corporate policy and URL reputation. If the policy decision is to allow access to the origin server, the proxy server forwards the re-encrypted

request to the destination service (on the origin server). The proxy decrypts the response from the origin server, accesses the clear text HTTP response, and optionally applies any policies to the response. The proxy then reencrypts the response and forwards it to the client. If the policy decision is to block the request to the origin server, the proxy can send an error response, such as HTTP 403, to the client.

To perform SSL interception, in addition to the proxy server configured earlier, you must configure the following on the ADC appliance:

- SSL profile
- SSL policy
- CA certificate store
- SSL-error autolearning and caching

> **Note:**
>
> HTTP/2 traffic is not intercepted by the SSL Interception feature.

SSL interception certificate store

An SSL certificate, which is a part of any SSL transaction, is a digital data form (X509) that identifies a company (domain) or an individual. An SSL certificate is issued by a certificate authority (CA). A CA can be private or public. Certificates issued by public CAs, such as Verisign, are trusted by applications that conduct SSL transactions. These applications maintain a list of CAs that they trust.

As a forward proxy, the ADC appliance performs encryption and decryption of traffic between a client and a server. It acts as a server to the client (user) and as a client to the server. Before an appliance can process HTTPS traffic, it must validate the identity of a server to prevent any fraudulent transactions. Therefore, as a client to the origin server, the appliance must verify the origin server certificate before accepting it. To verify a server certificate, all the certificates (for example, root and intermediate certificates) that are used to sign and issue the server certificate must be present on the appliance. A default set of CA certificates is preinstalled on an appliance. The appliance can use these certificates to verify almost all the common origin-server certificates. This default set cannot be modified. However, if your deployment requires more CA certificates, you can create a bundle of such certificates and import the bundle to the appliance. A bundle can also contain a single certificate.

When you import a certificate bundle to the appliance, the appliance downloads the bundle from the remote location and, after verifying that the bundle contains only certificates, installs it on the appliance. You must apply a certificate bundle before you can use it to validate a server certificate. You can also export a certificate bundle for editing or to store it in an offline location as a backup.

Import and apply a CA certificate bundle on the appliance by using the CLI

At the command prompt, type:


```
import ssl certBundle <name> <src>
apply ssl certBundle <name>
<!--NeedCopy-->
```

```
show ssl certBundle
<!--NeedCopy-->
```

ARGUMENTS:

Name:

Name to assign to the imported certificate bundle. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the CLI:

If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my file” or ‘my file’).

Maximum Length: 31

src:

URL specifying the protocol, host, and path, including file name, to the certificate bundle to be imported or exported. For example, `http://www.example.com/cert_bundle_file`.

NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access.

Maximum Length: 2047

Example:

```
import ssl certbundle swg-certbundle http://www.example.com/cert_bundle
apply ssl certBundle swg-certbundle
<!--NeedCopy-->
```

```
show ssl certbundle

            Name : swg-certbundle(Inuse)

            URL : http://www.example.com/cert_bundle

    Done
<!--NeedCopy-->
```


Import and apply a CA certificate bundle on the appliance by using the GUI

1. Navigate to **Security > SSL Forward Proxy > Getting Started > Certificate Bundles**.
2. Do one of the following:
   - Select a certificate bundle from the list.
   - To add a certificate bundle, click "+" and specify a name and source URL. Click **OK**.
3. Click **OK**.

Remove a CA certificate bundle from the appliance by using the CLI

At the command prompt, type:

```
remove certBundle <cert bundle name>
<!--NeedCopy-->
```

Example:

```
remove certBundle mytest-cacert
<!--NeedCopy-->
```

Export a CA certificate bundle from the appliance by using the CLI

At the command prompt, type:

```
export certBundle <cert bundle name> <Path to export>
<!--NeedCopy-->
```

ARGUMENTS:

Name:

Name to assign to the imported certificate bundle. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the CLI:

If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file').

Maximum Length: 31

src:

URL specifying the protocol, host, and path, including file name, to the certificate bundle to be imported or exported. For example, `http://www.example.com/cert_bundle_file`.

NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access.

Maximum Length: 2047

Example:

```
export certBundle mytest-cacert http://192.0.2.20/
<!--NeedCopy-->
```

Import, apply, and verify a CA certificate bundle from the Mozilla CA certificate store

At the command prompt, type:

```
> import certbundle mozilla_public_ca https://curl.haxx.se/ca/cacert.
   pem
Done
<!--NeedCopy-->
```

To apply the bundle, type:

```
> apply certbundle mozilla_public_ca
Done
<!--NeedCopy-->
```

To verify the certificate bundle in use, type:

```
> sh certbundle | grep mozilla
                Name : mozilla_public_ca (Inuse)
<!--NeedCopy-->
```

Limitations

- Certificate bundles are not supported in a cluster setup, or on a partitioned appliance.
- TLSv1.3 protocol is not supported with SSL Forward Proxy.

SSL policy infrastructure for SSL interception

A policy acts like a filter on incoming traffic. Policies on the ADC appliance help define how to manage proxied connections and requests. The processing is based on the actions that are configured for that policy. That is, data in connection requests is compared to a rule specified in the policy, and the action is applied to connections that match the rule (expression). After defining an action to assign to

the policy and create the policy, you must bind it to a proxy server, so that it applies to traffic flowing through that proxy server.

An SSL policy for SSL interception evaluates incoming traffic and applies a predefined action to requests that match a rule (expression). A decision to intercept, bypass, or reset a connection is made based on the defined SSL policy. You can configure one of three actions for a policy—INTERCEPT, BYPASS, or RESET. You must specify an action when you create a policy. To put a policy into effect, you must bind it to a proxy server on the appliance. To specify that a policy is intended for SSL interception, you must specify the type (bind point) as INTERCEPT_REQ when you bind the policy to a proxy server. When unbinding a policy, you must specify the type as INTERCEPT_REQ.

> **Note**:
>
> The proxy server cannot make a decision to intercept unless you specify a policy.

Traffic interception can be based on any SSL handshake attribute. The most commonly used is the SSL domain. The SSL domain is usually indicated by the attributes of the SSL handshake. It can be the Server Name Indicator value extracted from the SSL Client Hello message, if present, or the Server Alternate Name (SAN) value extracted from the origin server certificate. The SSL interception policy presents a special attribute, DETECTED_DOMAIN. This attribute makes it easier for the customers to author interception policies based on the SSL domain from the origin server certificate. The customer can match the domain name against a string, URL list (URL set or `patset`), or a URL category derived from the domain.

Create an SSL policy by using the CLI

At the command prompt, type:

```
add ssl policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

Examples:

The following examples are for policies with expressions that use the `detected_domain` attribute to check for a domain name.

Do not intercept traffic to a financial institution, such as XYZBANK

```
add ssl policy pol1 -rule client.ssl.detected_domain.contains("XYZBANK"
    ) -action BYPASS
<!--NeedCopy-->
```

Do not allow a user to connect to YouTube from the corporate network

```
add ssl policy pol2 -rule client.ssl.client.ssl.detected_domain.
    url_categorize(0,0).category.eq ("YouTube") -action RESET
```



```
<!--NeedCopy-->
```

Intercept all user traffic

```
add ssl policy pol3 -rule true -action INTERCEPT
<!--NeedCopy-->
```

If the customer doesn't want to use the detected_domain, they can use any of the SSL handshake attributes to extract and infer the domain.

For example, a domain name is not found in the SNI extension of the client hello message. The domain name must be taken from the origin server certificate. The following examples are for policies with expressions that check for a domain name in the subject name of the origin server certificate.

Intercept all user traffic to any Yahoo domain

```
add ssl policy pol4 -rule client.ssl.origin_server_cert.subject.
   contains("yahoo") -action INTERCEPT
<!--NeedCopy-->
```

Intercept all user traffic for the category "Shopping/Retail"

```
add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
   subject.URL_CATEGORIZE(0,0).CATEGORY.eq("Shopping/Retail") -action
   INTERCEPT
<!--NeedCopy-->
```

Intercept all user traffic to an uncategorized URL

```
add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
   subject.url_categorize(0,0).category.eq("Uncategorized") -action
   INTERCEPT
<!--NeedCopy-->
```

The following examples are for policies that match the domain against an entry in a URL set.

Intercept all user traffic if the domain name in SNI matches an entry in the URL set "top100"

```
add ssl policy pol_url_set -rule client.ssl.client_hello.SNI.
   URLSET_MATCHES_ANY("top100") -action INTERCEPT
<!--NeedCopy-->
```

Intercept all user traffic of the domain name if the origin server certificate matches an entry in the URL set "top100"


```
add ssl policy pol_url_set  -rule client.ssl.origin_server_cert.subject
   .URLSET_MATCHES_ANY("top100") -action INTERCEPT
<!--NeedCopy-->
```

Create an SSL policy to a proxy server by using the GUI

1. Navigate to **Traffic Management > SSL > Policies**.
2. On the **SSL Policies** tab, click **Add** and specify the following parameters:
   - Policy name
   - Policy action – Select from intercept, bypass, or reset.
   - Expression
3. Click **Create**.

Bind an SSL policy to a proxy server by using the CLI

At the command prompt, type:

```
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type  INTERCEPT_REQ
<!--NeedCopy-->
```

Example:

```
bind ssl vserver <name> -policyName pol1 -priority 10 -type
   INTERCEPT_REQ
<!--NeedCopy-->
```

Bind an SSL policy to a proxy server by using the GUI

1. Navigate to **Security > SSL Forward Proxy > Proxy Virtual Servers**.
2. Select a virtual server and click **Edit**.
3. In **Advanced Settings**, click **SSL Policies**.
4. Click inside the **SSL Policy** box.
5. In **Select Policy**, select a policy to bind.
6. In **Type**, select **INTERCEPT_REQ**.
7. Click **Bind** and then click **OK**.

Unbind an SSL policy to a proxy server by using the CLI

At the command prompt, type:


```
unbind ssl vserver <vServerName> -policyName <string> -type
    INTERCEPT_REQ
<!--NeedCopy-->
```

SSL expressions used in SSL policies

Expression	Description
CLIENT.SSL.CLIENT_HELLO.SNI.*	Returns the SNI extension in a string format. Evaluate the string to see if it contains the specified text. Example: client.ssl.client_hello.sni.contains("xyz.com")
CLIENT.SSL.ORIGIN_SERVER_CERT.*	Returns a certificate, received from a back-end server, in a string format. Evaluate the string to see if it contains the specified text. Example: client.ssl.origin_server_cert.subject.contains("xyz.com")
CLIENT.SSL.DETECTED_DOMAIN.*	Returns a domain, either from the SNI extension or from the origin server certificate, in a string format. Evaluate the string to see if it contains the specified text. Example: client.ssl.detected_domain.contains("xyz.com")

SSL error autolearning

The appliance adds a domain to the SSL bypass list if learning mode is on. The learning mode is based on the SSL alert message received from either a client or an origin server. That is, learning depends the client or server sending an alert message. There is no learning if an alert message is not sent. The appliance learns if any of the following conditions are met:

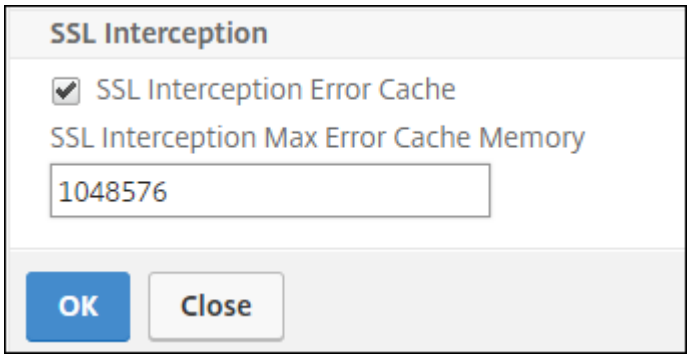
1. A request for a client certificate is received from the server.
2. Any one of the following alerts is received as part of the handshake:
   - BAD_CERTIFICATE
   - UNSUPPORTED_CERTIFICATE
   - CERTIFICATE_REVOKED
   - CERTIFICATE_EXPIRED

- CERTIFICATE_UNKNOWN
- UNKNOWN_CA (If a client uses pinning, it sends this alert message if it receives a server certificate.)
- HANDSHAKE_FAILURE

To enable learning, you must enable the error cache and specify the memory reserved for learning.

Enable learning by using the GUI

1. Navigate to **Traffic Management > SSL**.
2. In **Settings**, click **Change advanced SSL settings**.
3. In **SSL Interception**, select **SSL Interception Error Cache**.
4. In **SSL Interception Max Error Cache Memory**, specify the memory (in bytes) to reserve.



5. Click **OK**.

Enable learning by using the CLI

At the command prompt type:

```
set ssl parameter -ssliErrorCache ( ENABLED | DISABLED ) -
    ssliMaxErrorCacheMem <positive_integer>
<!--NeedCopy-->
```

Arguments:

ssliErrorCache:

Enable or disable dynamic learning, and cache the learned information to make subsequent decisions to intercept or bypass requests. When enabled, the appliance performs a cache lookup to decide whether to bypass the request.

Possible values: ENABLED, DISABLED

Default value: DISABLED

ssliMaxErrorCacheMem:

Specify the maximum memory, in bytes, that can be used to cache the learned data. This memory is used as an LRU cache so that the old entries are replaced with new entries after the set memory limit is exhausted. A value of 0 decides the limit automatically.

Default value: 0

Minimum value: 0

Maximum value: 4294967294

SSL profile

An SSL profile is a collection of SSL settings, such as ciphers and protocols. A profile is helpful if you have common settings for different servers. Instead of specifying the same settings for each server, you can create a profile, specify the settings in the profile, and then bind the profile to different servers. If a custom front-end SSL profile is not created, the default front-end profile is bound to client-side entities. This profile enables you to configure settings for managing the client-side connections.

For SSL interception, you must create an SSL profile and enable SSL interception in the profile. A default cipher group is bound to this profile, but you can configure more ciphers to suit your deployment. Bind an SSL interception CA certificate to this profile and then bind the profile to a proxy server. For SSL interception, the essential parameters in a profile are the ones used for the following actions:

- Check the OCSP status of the origin server certificate.
- Trigger client renegotiation if the origin server requests renegotiation.
- Verify the origin server certificate before reusing the front-end SSL session.

Use the default back-end profile when communicating with the origin servers. Set any server-side parameters, such as cipher suites, in the default back-end profile. A custom back-end profile is not supported.

For examples of the most commonly used SSL settings, see “Sample Profile” at the end of this section.

Cipher/protocol support differs on the internal and external network. In the following tables, the connection between the users and an ADC appliance is the internal network. The external network is between the appliance and the internet.

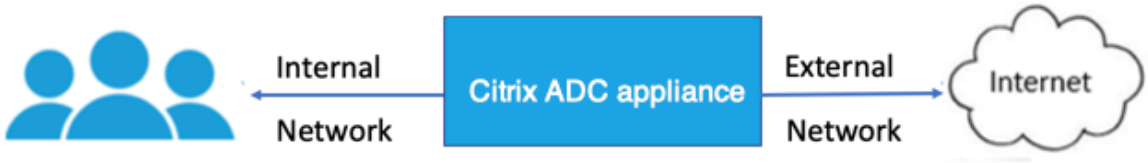


Table 1: Cipher/protocol support matrix for the internal network

See Table 1-Support on virtual server/frontend service/internal service in Ciphers available on the Citrix ADC appliances.

Table 2: Cipher/protocol support matrix for the external network

See Table 2-Support on back-end services in Ciphers available on the Citrix ADC appliances.

Add an SSL profile and enable SSL interception by using the CLI

At the command prompt, type:

```
add ssl profile <name> -sslinterception ENABLED -ssliReneg ( ENABLED |
 DISABLED )-ssliOCSPCheck ( ENABLED | DISABLED )-ssliMaxSessPerServer <
positive_integer>
```

Arguments:

sslInterception:

Enable or disable interception of SSL sessions.

Possible values: ENABLED, DISABLED

Default value: DISABLED

ssliReneg:

Enable or disable triggering client renegotiation when a renegotiation request is received from the origin server.

Possible values: ENABLED, DISABLED

Default value: ENABLED

ssliOCSPCheck:

Enable or disable OCSP check for an origin-server certificate.

Possible values: ENABLED, DISABLED

Default value: ENABLED

ssliMaxSessPerServer:

Maximum number of SSL sessions to be cached per dynamic origin server. A unique SSL session is created for each SNI extension received from the client in a client hello message. The matching session is used for server-session reuse.

Default value: 10

Minimum value: 1

Maximum value: 1000

Example:


```
add ssl profile swg_ssl_profile  -sslinterception ENABLED

Done

sh ssl profile swg_ssl_profile

1)    Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                    .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                NO

                Session Reuse: ENABLED
                    Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                    Ephemeral RSA: ENABLED
                    Refresh Count: 0

                Deny SSL Renegotiation
                    ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED

                Push Encryption Trigger: Always

                PUSH encryption trigger timeout:             1 ms

                SNI: DISABLED
```



```
40
41                 OCSP Stapling: DISABLED
42
43                 Strict Host Header check for SNI enabled SSL sessions:
                                    NO
44
45                 Push flag:             0x0 (Auto)
46
47                 SSL quantum size:                              8 kB
48
49                 Encryption trigger timeout           100 mS
50
51                 Encryption trigger packet count:                45
52
53                 Subject/Issuer Name Insertion Format: Unicode
54
55                 SSL Interception: ENABLED
56
57                 SSL Interception OCSP Check: ENABLED
58
59                 SSL Interception End to End Renegotiation: ENABLED
60
61                 SSL Interception Server Cert Verification for Client
                      Reuse: ENABLED
62
63                 SSL Interception Maximum Reuse Sessions per Server:  10
64
65                 Session Ticket: DISABLED              Session Ticket
                      Lifetime: 300 (secs)
66
67                 HSTS: DISABLED
68
69                 HSTS IncludeSubDomains: NO
70
71                 HSTS Max-Age: 0
72
73                 ECC Curve: P_256, P_384, P_224, P_521
74
75 1)             Cipher Name: DEFAULT Priority :1
76
77                   Description: Predefined Cipher Alias
78
79 Done
80 <!--NeedCopy-->
```


Bind an SSL interception CA certificate to an SSL profile by using the CLI

At the command prompt, type:

```
bind ssl profile <name> -ssliCACertkey <ssli-ca-cert>
```

Example:

```
bind ssl profile swg_ssl_profile -ssliCACertkey swg_ca_cert

Done

sh ssl profile swg_ssl_profile

1)            Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED              TLSv1.0: ENABLED  TLSv1
                   .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                NO

                Session Reuse: ENABLED
                   Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                   Ephemeral RSA: ENABLED
                   Refresh Count: 0

                Deny SSL Renegotiation
                   ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED
```



```
34
35                  Push Encryption Trigger: Always
36
37                  PUSH encryption trigger timeout:             1 ms
38
39                  SNI: DISABLED
40
41                  OCSP Stapling: DISABLED
42
43                  Strict Host Header check for SNI enabled SSL sessions:
                                NO
44
45                  Push flag:            0x0 (Auto)
46
47                  SSL quantum size:                            8 kB
48
49                  Encryption trigger timeout           100 mS
50
51                  Encryption trigger packet count:               45
52
53                  Subject/Issuer Name Insertion Format: Unicode
54
55                  SSL Interception: ENABLED
56
57                  SSL Interception OCSP Check: ENABLED
58
59                  SSL Interception End to End Renegotiation: ENABLED
60
61                  SSL Interception Server Cert Verification for Client
                       Reuse: ENABLED
62
63                  SSL Interception Maximum Reuse Sessions per Server:  10
64
65                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
66
67                  HSTS: DISABLED
68
69                  HSTS IncludeSubDomains: NO
70
71                  HSTS Max-Age: 0
72
73                  ECC Curve: P_256, P_384, P_224, P_521
74
75 1)            Cipher Name: DEFAULT Priority :1
```



```
76
77                  Description: Predefined Cipher Alias
78
79 1)            SSL Interception CA CertKey Name: swg_ca_cert
80
81 Done
82 <!--NeedCopy-->
```

Bind an SSL interception CA certificate to an SSL profile by using the GUI

1. Navigate to **System** > **Profiles** > **SSL Profile**.
2. Click **Add**.
3. Specify a name for the profile.
4. Enable **SSL Sessions Interception**.
5. Click **OK**.
6. In **Advanced Settings**, click **Certificate Key**.
7. Specify an SSL interception CA certificate key to bind to the profile.
8. Click **Select** and then click **Bind**.
9. Optionally, configure ciphers to suit your deployment.
    - Click the edit icon, and then click **Add**.
    - Select one or more cipher groups, and click the right arrow.
    - Click **OK**.
10. Click **Done**.

Bind an SSL profile to a proxy server by using the GUI

1. Navigate to **Security** >**SSL Forward Proxy** > **Proxy Virtual Servers**, and add a server or select a server to modify.
2. In **SSL Profile**, click the edit icon.
3. In the **SSL Profile** list, select the SSL profile that you created earlier.
4. Click **OK**.
5. Click **Done**.

Sample Profile:

```
1 Name: swg_ssl_profile (Front-End)
2
```



```
3                   SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                        .1: ENABLED  TLSv1.2: ENABLED
4
5                   Client Auth: DISABLED
6
7                   Use only bound CA certificates: DISABLED
8
9                   Strict CA checks:                                 NO
10
11                  Session Reuse: ENABLED
                        Timeout: 120 seconds
12
13                  DH: DISABLED
14
15                  DH Private-Key Exponent Size Limit: DISABLED
                        Ephemeral RSA: ENABLED
                        Refresh Count: 0
16
17                  Deny SSL Renegotiation
                        ALL
18
19                  Non FIPS Ciphers: DISABLED
20
21                  Cipher Redirect: DISABLED
22
23                  SSL Redirect: DISABLED
24
25                  Send Close-Notify: YES
26
27                  Strict Sig-Digest Check: DISABLED
28
29                  Push Encryption Trigger: Always
30
31                  PUSH encryption trigger timeout:              1 ms
32
33                  SNI: DISABLED
34
35                  OCSP Stapling: DISABLED
36
37                  Strict Host Header check for SNI enabled SSL sessions:
                                          NO
38
39                  Push flag:             0x0 (Auto)
40
41                  SSL quantum size:                             8 kB
```



```
42
43                  Encryption trigger timeout           100 mS
44
45                  Encryption trigger packet count:               45
46
47                  Subject/Issuer Name Insertion Format: Unicode
48
49                  SSL Interception: ENABLED
50
51                  SSL Interception OCSP Check: ENABLED
52
53                  SSL Interception End to End Renegotiation: ENABLED
54
55                  SSL Interception Maximum Reuse Sessions per Server:  10
56
57                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
58
59                  HSTS: DISABLED
60
61                  HSTS IncludeSubDomains: NO
62
63                  HSTS Max-Age: 0
64
65                  ECC Curve: P_256, P_384, P_224, P_521
66
67 1)             Cipher Name: DEFAULT Priority :1
68
69                  Description: Predefined Cipher Alias
70
71 1)             SSL Interception CA CertKey Name: swg_ca_cert
72 <!--NeedCopy-->
```

User identity management

September 14, 2021

An increasing number of security breaches and the growing popularity of mobile devices has emphasized the need to ensure that use of the external internet is compliant with the corporate policies. Only authorized users must be allowed access to external resources provisioned by the corporate personnel. Identity Management makes it possible by verifying the identity of a person or a device. It does not determine what tasks the individual can take or what files the individual can see.

An SSL forward proxy deployment identifies the user before allowing access to the internet. All requests and responses from the user are inspected. User activity is logged, and records are exported to the Citrix Application Delivery Management (ADM) for reporting. In Citrix ADM, you can view the statistics about the user activities, transactions, and bandwidth consumption.

By default, only the user's IP address is saved, but you can configure the feature to record more details about the user. You can use this identity information to create richer internet usage policies for specific users.

The Citrix ADC appliance supports the following authentication modes for an explicit-proxy configuration.

- **Lightweight Directory Access Protocol (LDAP)**. Authenticates the user through an external LDAP authentication server. For more information, see LDAP Authentication Policies.
- **RADIUS**. Authenticates the user through an external RADIUS server. For more information, see RADIUS Authentication Policies.
- **TACACS+**. Authenticates the user through an external Terminal Access Controller Access-Control System (TACACS) authentication server. For more information, see Authentication Policies.
- **Negotiate**. Authenticates the user through a Kerberos authentication server. If there is an error in Kerberos authentication, the appliance uses NTLM authentication. For more information, see Negotiate Authentication Policies.

For transparent proxy, only IP-based LDAP authentication is supported. When a client request is received, the proxy authenticates the user by checking an entry for the client IP address in the active directory. It then creates a session based on the user IP address. However, if you configure the ssoNameAttribute in an LDAP action, a session is created by using the user name instead of the IP address. Classic policies are not supported for authentication in a transparent proxy setup.

> **Note**
>
> For explicit proxy, you must set the LDAP login name to **sAMAccountName**. For transparent proxy, you must set the LDAP login name to **networkAddress** and attribute1 to **sAMAccountName**.

Example for explicit proxy:

```
1 add authentication ldapAction swg-auth-action-explicit -serverIP
     10.105.157.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "
     CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword
     freebsd123$ -ldapLoginName sAMAccountName
2 <!--NeedCopy-->
```

Example for transparent proxy:


```
add authentication ldapAction swg-auth-action-explicit -serverIP
    10.105.157.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "
    CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword
    freebsd123$ -ldapLoginName networkAddress -authentication disable -
    Attribute1 sAMAccountName
<!--NeedCopy-->
```

Set up user authentication by using the CLI

At the command prompt type:

```
add authentication vserver <vserver name> SSL

bind ssl vserver <vserver name> -certkeyName <certkey name>

add authentication ldapAction <action name> -serverIP <ip_addr> -
    ldapBase <string> -ldapBindDn <string> -ldapBindDnPassword -
    ldapLoginName <string>

add authentication Policy <policy name> -rule <expression> -action <
    string>

bind authentication vserver <vserver name> -policy <string> -priority <
    positive_integer>

set cs vserver <name> -authn401 ON -authnVsName <string>
<!--NeedCopy-->
```

Arguments:

Vserver name:

Name of the authentication virtual server to which to bind the policy.

Maximum Length: 127

serviceType:

Protocol type of the authentication virtual server. Always SSL.

Possible values: SSL

Default value: SSL

Action name:

Name for the new LDAP action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@),

equals (=), colon (:), and underscore characters. Cannot be changed after the LDAP action is added. The following requirement applies only to the CLI:

If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my authentication action” or ‘my authentication action’).

Maximum Length: 127

serverIP:

IP address assigned to the LDAP server.

ldapBase:

Base (node) from which to start LDAP searches. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com. Maximum Length: 127

ldapBindDn:

Full distinguished name (DN) that is used to bind to the LDAP server.

Default: cn=Manager,dc=netscaler,dc=com

Maximum Length: 127

ldapBindDnPassword:

Password used to bind to the LDAP server.

Maximum Length: 127

ldapLoginName:

LDAP login name attribute. The Citrix ADC appliance uses the LDAP login name to query external LDAP servers or Active Directories. Maximum Length: 127

Policy name:

Name for the advance AUTHENTICATION policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after an AUTHENTICATION policy is created. The following requirement applies only to the CLI:

If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, “my authentication policy” or ‘my authentication policy’).

Maximum Length: 127

rule:

Name of the rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the AUTHENTICATION server.

Maximum Length: 1499

action:

Name of the authentication action to be performed if the policy matches.

Maximum Length: 127

priority:

Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Policies are evaluated in the order of their priorities, and the first policy that matches the request is applied. Must be unique within the list of policies bound to the authentication virtual server.

Minimum value: 0

Maximum Value: 4294967295

Example:

```
add authentication vserver swg-auth-vs SSL

Done

bind ssl vserver explicit-auth-vs -certkeyName ns-swg-ca-certkey

Done

add authentication ldapAction swg-auth-action-explicit -serverIP
    192.0.2.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "CN=
    Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword zzzzz
     -ldapLoginName sAMAccountName

Done

add authenticationpolicy swg-auth-policy -rule true -action swg-auth-
    action-explicit
Done

bind authentication vserver swg-auth-vs -policy swg-auth-policy -
    priority 1

Done

set cs vserver testswg -authn401 ON -authnVsName swg-auth-vs

Done
<!--NeedCopy-->
```


Enable user name logging by using the CLI

At the command prompt, type:

```
set appflow param -AAAUserName ENABLED
<!--NeedCopy-->
```

Arguments:

AAAUserName

Enable AppFlow authentication, authorization, and auditing user name logging.

Possible values: ENABLED, DISABLED

Default value: DISABLED

Example:

```
set appflow param -AAAUserName ENABLED
<!--NeedCopy-->
```

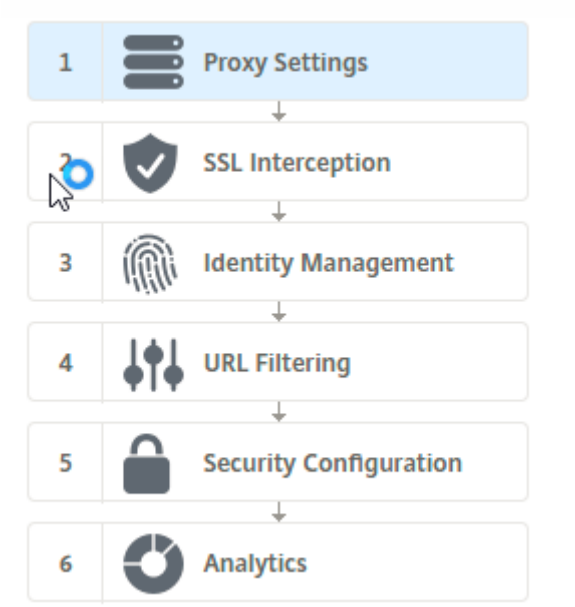
URL filtering

September 14, 2021

URL Filtering provides policy based control of websites by using the information contained in URLs. This feature helps network administrators monitor and control user access to malicious websites on the network.

Get started

If you are a new user and want to configure URL filtering, you must complete the initial SSL forward proxy setup. To get started with URL Filtering, you must first log on to the SSL forward proxy wizard. The wizard takes you through a series of configuration steps before you apply the URL Filtering policies.



> **Note**
>
> Before you begin, be sure you have a valid URL Threat Intelligence feature license installed on your appliance. If you are using a trial version, be sure to purchase a valid license to continue using this feature on the ADC appliance.

Log on to SSL forward proxy wizard

The SSL forward proxy wizard guides you through a series of simplified configuration tasks and the right pane displays the corresponding flow sequence. You can use this wizard to apply URL Filtering policies to a URL list or a predefined list of categories.

Step 1: Configure proxy settings

First configure a proxy server through which the client accesses the gateway. This server is of type SSL, and it operates in explicit or transparent mode. For more information about proxy server configuration, see Proxy Modes.

Step 2: Configure SSL interception

After configuring the proxy server, you must configure the SSL interception proxy to intercept encrypted traffic at the Citrix ADC appliance. In the case of URL filtering, the SSL proxy intercepts the traffic and does not allow blocked URLs while all other traffic can be bypassed. For more information about configuring SSL interception, see SSL Interception.

Step 3: Configure identity management

A user is authenticated before being allowed to log on to the enterprise network. Authentication provides the flexibility to define specific policies for a user or a group of users, based on their roles. For more information about user authentication, see User Identify Management.

Step 4: Configure URL filtering

The administrator can apply a URL filtering policy either by using the URL Categorization feature or by using the URL List feature.

URL Categorization. Controls access to websites and webpages by filtering traffic based on a predefined list of categories.

URL List. Controls access to blacklisted websites and webpages by denying access to URLs that are in a URL set imported into the appliance.

Step 5: Configure security configuration

This step enables you to configure a reputation score and allow users to control access to the websites by denying access if the score is too low. Your reputation score can range from one to four, and you can configure the threshold at which the score becomes unacceptable. For scores that exceed the threshold, you can select a policy action to allow, block, or redirect traffic. For more information, see URL Reputation Score.

Step 6: Configure SSL forward proxy analytics

This step enables you to activate SSL forward proxy analytics for categorizing web traffic, logging URL category in the user transaction logs and viewing traffic analytics. For more information about SSL forward proxy analytics, see Analytics.

Step 7: Click "Done" to complete the initial configuration and continue managing the URL filtering configuration

URL list

September 14, 2021

The URL List feature enables enterprise customers to control access to specific websites and website categories. The feature filters websites by applying a responder policy bound to a URL matching algorithm. The algorithm matches the incoming URL against a URL set consisting of up to one million

(1,000,000) entries. If the incoming URL request matches an entry in the set, the appliance uses the responder policy to evaluate the request (HTTP/HTTPS) and control access to it.

URL set types

Each entry in a URL set can include a URL and, optionally, its metadata (URL category, category groups, or any other related data). For URLs with metadata, the appliance uses a policy expression that evaluates the metadata. For more information, see URL Set.

SSL forward proxy supports custom URL sets. You can also use pattern sets to filter URLs.

Custom URL set. You can create a customized URL set with up to 1,000,000 URL entries and import it as a text file into your appliance.

Pattern set. An ADC appliance can use pattern sets to filter URLs before granting access to websites. A pattern set is a string-matching algorithm that looks for an exact string match between an incoming URL and up to 5000 entries. For more information, see Pattern Set.

Each URL in an imported URL set can have a custom category in the form of URL metadata. Your organization can host the set and configure the ADC appliance to periodically update the set without requiring manual intervention.

After the set is updated, the Citrix ADC appliance automatically detects the metadata. The category is now available as a policy expression for evaluating the URL and applying an action such as allow, block, redirect, or notify the user.

Advanced policy expressions used with URL sets

The following table describes the basic expressions you can use to evaluate incoming traffic.

1. .URLSET_MATCHES_ANY - Evaluates to TRUE if the URL exactly matches any entry in the URL set.
2. .GET_URLSET_METADATA() - The GET_URLSET_METADATA() expression returns the associated metadata if the URL exactly matches any pattern within the URL set. An empty string is returned if there is no match.
3. `.GET_ URLSET_METADATA().EQ(<METADATA)- .GET_ URLSET_METADATA().EQ(<METADATA)`
4. .GET_URLSET_METADATA ().TYPECAST_LIST_T(‘,’).GET(0).EQ() - Evaluates to TRUE if the matched metadata is at the beginning of the category. This pattern can be used to encode separate fields within metadata but match only the first field.
5. HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL) - Joins the host and URL parameters, which can then be used for matching.

Responder action types

> **Note:** In the table, HTTP.REQ.URL is generalized as <URL expression>.

The following table describes the actions that can be applied to incoming internet traffic.

Responder Action	Description
Allow	Allow the request to access the target URL.
Redirect	Redirect the request to the URL specified as the target.
Block	Deny the request.

Prerequisites

Configure a DNS server if you import a URL Set from a host name URL. This configuration is not required if you use an IP address.

At the command prompt, type:

```
add dns nameServer ((<IP> [-local])| <dnsVserverName>)[-state (ENABLED |
DISABLED )] [-type <type>] [-dnsProfileName <string>]
```

Example:

```
add dns nameServer 10.140.50.5
```

Configure a URL list

To configure a URL list, you can use the Citrix SSL forward proxy wizard or the Citrix ADC command-line interface (CLI). On the Citrix ADC appliance, you must first configure the responder policy and then bind the policy to a URL set.

Citrix recommends that you use the Citrix SSL forward proxy wizard as the preferred option to configure a URL list. Use the wizard to bind a responder policy to a URL set. Alternatively, you can bind the policy to a pattern set.

Configure a URL list by using the SSL forward proxy wizard

To configure URL List for HTTPS traffic by using the GUI:

1. Navigate to **Security** > **SSL Forward Proxy** page.
2. In the details pane, do one of the following:
   a) Click **SSL Forward Proxy Wizard**.

b) Select an existing configuration and click **Edit**.

3. In the **URL Filtering** section, click **Edit**.
4. Select the **URL List** check box to enable the feature.
5. Select a **URL List** policy and Click **Bind**.
6. Click **Continue** and then **Done**.

For more information, see How to Create a URL List Policy.

Configure a URL list by using the CLI

To configure a URL list, do the following.

1. Configure a proxy virtual server for HTTP and HTTPS traffic.
2. Configure SSL interception for intercepting HTTPS traffic.
3. Configure a URL list containing a URL set for HTTP traffic.
4. Configure URL list containing URL set for HTTPS traffic.
5. Configure a private URL set.

> **Note**
>
> If you have already configured an ADC appliance, you can skip steps 1 and 2, and configure with step 3.

Configuring a proxy virtual server for Internet traffic

The Citrix ADC appliance supports transparent and explicit proxy virtual servers. To configure a proxy virtual server for internet traffic in explicit mode, do the following:

1. Add a proxy SSL virtual server.
2. Bind a responder policy to the proxy virtual server.

To add a proxy virtual server by using the CLI:

At the command prompt, type:

```
add cs vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

Example:

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
<!--NeedCopy-->
```

To bind a responder policy to a proxy virtual server by using the CLI:


```
bind ssl vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

> **Note**
>
> If you have already configured the SSL interceptor as part of the Citrix ADC configuration, you can skip the following procedure.

Configure SSL interception for HTTPS traffic

To configure SSL interception for HTTPS traffic, do the following:

1. Bind a CA certificate-key pair to the proxy virtual server.
2. Enable the default SSL profile.
3. Create a front-end SSL profile, and bind it to the proxy virtual server and enable SSL interception in the front-end SSL profile.

To bind a CA certificate-key pair to the proxy virtual server by using the CLI:

At the command prompt, type:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
<!--NeedCopy-->
```

To configure a front-end SSL profile by using the CLI:

At the command prompt, type:

```
set ssl parameter -defaultProfile ENABLED

add ssl profile <name> -sslInterception ENABLED -ssliMaxSessPerServer <
    positive_integer>
<!--NeedCopy-->
```

To bind a front-end SSL profile to a proxy virtual server by using the CLI

At the command prompt, type:

```
set ssl vserver <vServer name>  -sslProfile <name>
<!--NeedCopy-->
```

Configure a URL list by importing a URL set for HTTP traffic

For information about how to configure a URL Set for HTTP traffic, see URL Set.

Perform explicit subdomain match

You can now perform an explicit subdomain match for an imported URL set. A new parameter, "subdomainExactMatch" is added to the `import policy URLset` command.

When you enable the parameter, the URL Filtering algorithm performs an explicit subdomain match. For example, if the incoming URL is `news.example.com` and if the entry in the URL set is `example.com`, the algorithm does not match the URLs.

At the command prompt, type:

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-rowSeparator
 <character>] -url [-interval <secs>] [-privateSet][-subdomainExactMatch]
[-canaryUrl <URL>]
```

Example

```
import policy urlset test -url http://10.78.79.80/top-1k.csv -privateSet -
subdomainExactMatch -interval 900
```

Configure a URL set for HTTPS traffic

To configure a URL Set for HTTPS traffic by using the CLI

At the command prompt type:

```
1 add ssl policy <name> -rule <expression> -action <string> [-undefAction
      <string>] [-comment <string>]
2 <!--NeedCopy-->
```

Example:

```
1 add ssl policy pol1 -rule "client.ssl.client_hello.SNI.
     URLSET_MATCHES_ANY("top1m") -action INTERCEPT
2 <!--NeedCopy-->
```

To configure a URL set for HTTPS traffic by using the SSL forward proxy wizard

Citrix recommends that you use the SSL forward proxy wizard as the preferred option to configure a URL list. Use the wizard to import a custom URL set and bind to a responder policy.

1. Navigate to **Security > SSL Forward proxy > URL Filtering > URL Lists**.
2. In the details pane, click **Add**.
3. On the **URL List Policy** page, specify the policy name.
4. Select an option to import a URL set.
5. On the **URL List Policy** tab page, select the **Import URL Set** check box and specify the following URL Set parameters.

a) URL Set Name—Name of the custom URL set.
b) URL—Web address of the location at which to access the URL Set.
c) Overwrite—Overwrite a previously imported URL set.
d) Delimiter—Character sequence that delimits a CSV file record.
e) Row Separator—Row separator used in the CSV file.
f) Interval—Interval in seconds, rounded off to the nearest number of seconds equal to 15 minutes, at which the URL set is updated.
g) Private Set—Option to prevent exporting the URL set.
h) Canary URL—Internal URL for testing whether the content of the URL set is to be kept confidential. The maximum length of the URL is 2047 characters.

6. Select a responder action from the drop-down list.
7. Click **Create** and **Close**.

Configure a private URL set

If you configure a private URL set and keep its contents confidential, the network administrator might not know the blacklisted URLs in the set. For such cases, you can configure a Canary URL and add it to the URL set. Using the Canary URL, the administrator can request the private URL Set to be used for every lookup request. You can refer to the wizard section for descriptions of each parameter.

To import a URL set by using the CLI:

At the command prompt, type:

```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
   rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
   ] [-canaryUrl <URL>]
<!--NeedCopy-->
```

Example:

```
import policy urlset test1 -url http://10.78.79.80/alytra/top-1k.csv -
   private -canaryUrl http://www.in.gr
<!--NeedCopy-->
```

Display imported URL set

You can now display imported URL sets in addition to added URL sets. A new parameter “imported” is added to the `show urlset` command. If you enable this option, the appliance displays all imported URL sets and distinguishes the imported URL sets from the added URL sets.

At the command prompt, type:

```
show policy urlset [<name>] [-imported]
```


Example

```
show policy urlset -imported
```

Configure audit log messaging

Audit logging enables you to review a condition or a situation in any phase of a URL List process. When a Citrix ADC appliance receives an incoming URL, if the responder policy has a URL Set advanced policy expression, the audit log feature collects URL Set information in the URL. It stores the details as a log message for any target allowed by audit logging.

The log message contains the following information:

1. Timestamp.
2. Log message type.
3. The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency).
4. Log message information, such as URL set name, policy action, URL.

To configure audit logging for the URL List feature, you must complete the following tasks:

1. Enable Audit Log.
2. Create Audit Log message action.
3. Set URL List responder policy with Audit Log message action.

For more information, see Audit Logging topic.

URL pattern semantics

September 14, 2021

The following table shows the URL patterns used for specifying the list of pages you to want to filter. For example, the pattern, www.example.com/bar matches only one page at www.example.com/bar. To match all the pages whose URL starts with ‘ www.example.com/bar’, you add an asterisk (*) at the end of the URL.

Semantics for URL pattern to match metadata mapping

The pattern matching semantics is available in a table format. For more information, see Pattern Semantics PDF page.

Mapping URL categories

September 14, 2021

A list of third party categories and category groups. For more information, see URL Category Mapping page.

Use case: URL filtering by using custom URL set

September 14, 2021

If you are an enterprise customer looking to control access to specific websites and website categories, use a custom URL set bound to a responder policy. Your organization's network infrastructure can use a URL filter to block access to malicious or dangerous websites. For example, websites featuring adult, violence, gaming, drugs, politics, or job portals. In addition to filtering the URLs, you can create a customized list of URLs and import it to the ADC appliance. For example, your organization's policies might call for blocking access to certain websites such as social networking, shopping portals, and job portals.

Each URL in the list can have a custom category in the form of metadata. The organization can host the list of URLs as a URL set on the Citrix ADC appliance. Configure the appliance to periodically update the set without requiring manual intervention.

After the set is updated, the Citrix ADC appliance automatically detects the metadata. The responder policy uses the URL metadata (category details) to evaluate the incoming URL and apply an action such as allow, block, redirect, or notify the user.

To do so, configure in your network, you can perform the following tasks:

1. Import a custom URL set
2. Add a custom URL set
3. Configure a custom URL list in the SSL Forward Proxy wizard.

Import a custom URL set by using the CLI

At the command prompt, type:

```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
   rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
   ] [-canaryUrl <URL>]

import policy urlset test1  -url http://10.78.79.80/alytra/top-1k.csv
```



```
<!--NeedCopy-->
```

Add a custom URL set by using the CLI

At the command prompt, type:

```
add urlset <urlset_name>
```

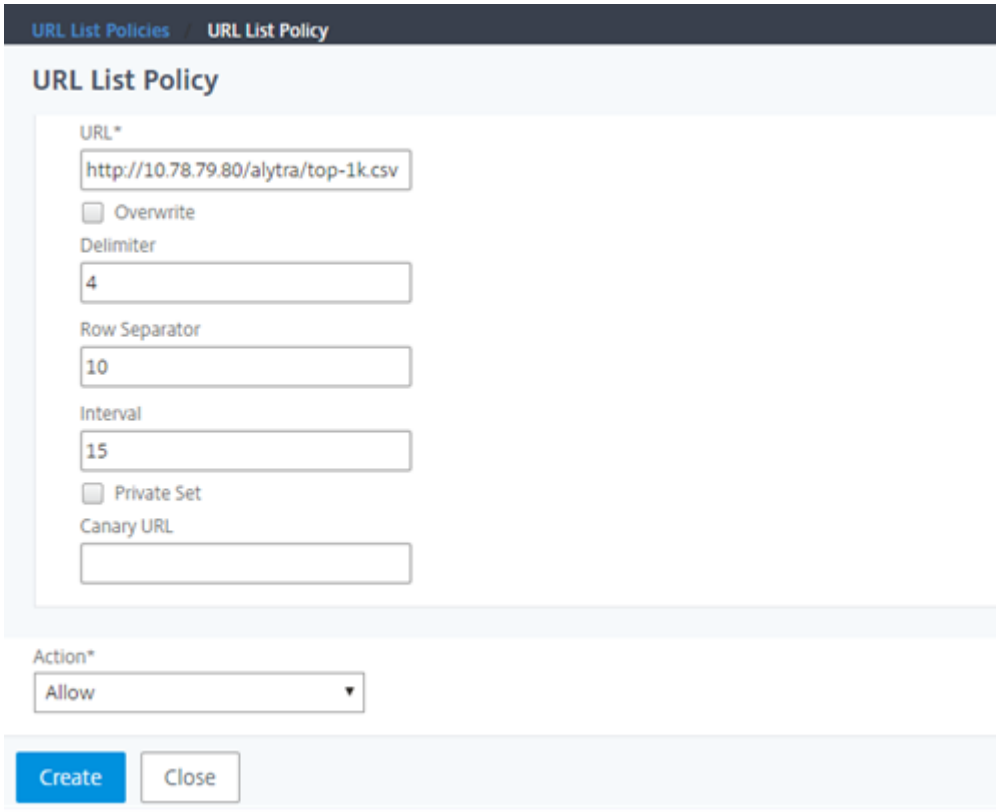
Example:

```
add urlset test1
```

Configure a URL list by using the SSL Forward Proxy wizard

Citrix recommends that you use the SSL Forward Proxy wizard as the preferred option to configure a URL list. Use the wizard to import a custom URL set and bind it to a responder policy.

1. Navigate to **Security > SSL Forward Proxy > URL Filtering > URL Lists**.
2. In the details pane, click **Add**.
3. On the **URL List Policy** page, specify the policy name.
4. Select an option to either import a URL set.
5. In the **URL List Policy** tab page, select the **Import URL Set** check box and specify the following URL Set parameters.
   a) URL Set Name—Name of the custom URL set.
   b) URL—Web address of the location at which to access the URL Set.
   c) Overwrite—Overwrite a previously imported URL set.
   d) Delimiter—Character sequence that delimits a CSV file record.
   e) Row Separator—Row separator used in the CSV file.
   f) Interval—Interval in seconds, rounded off to the nearest 15 minutes, at which the URL set is updated.
   g) Private Set—Option to prevent exporting the URL set.
   h) Canary URL—Internal URL for testing if the content of the URL set is to be kept confidential. The maximum length of the URL is 2047 characters.
6. Select a responder action from the drop-down list.
7. Click **Create** and **Close**.



Metadata semantics for custom URL sets

To import a custom URL set, add the URLs to a text file and bind it to a responder policy to block Social networking URLs.

Following are examples of URLs that you might add to the text file:

cnn.com, News

bbc.com, News

google.com, Search Engine

yahoo.com, Search Engine

facebook.com, Social Media

twitter.com, Social Media

Configure a responder policy to block social media URLs by using the CLI

```
1 add responder action act_url_unauthorized respondwith '"HTTP/1.1 451
     Unavailable For Legal Reasons\r\n\r\nURL is NOT authorized\n"
2
```



```
add responder policy pol_url_meta_match 'HTTP.REQ.HOSTNAME.APPEND(HTTP.
    REQ.URL).GET_URLSET_METADATA("u1").EQ("Social Media")'
    act_url_meta_match
<!--NeedCopy-->
```

URL categorization

September 14, 2021

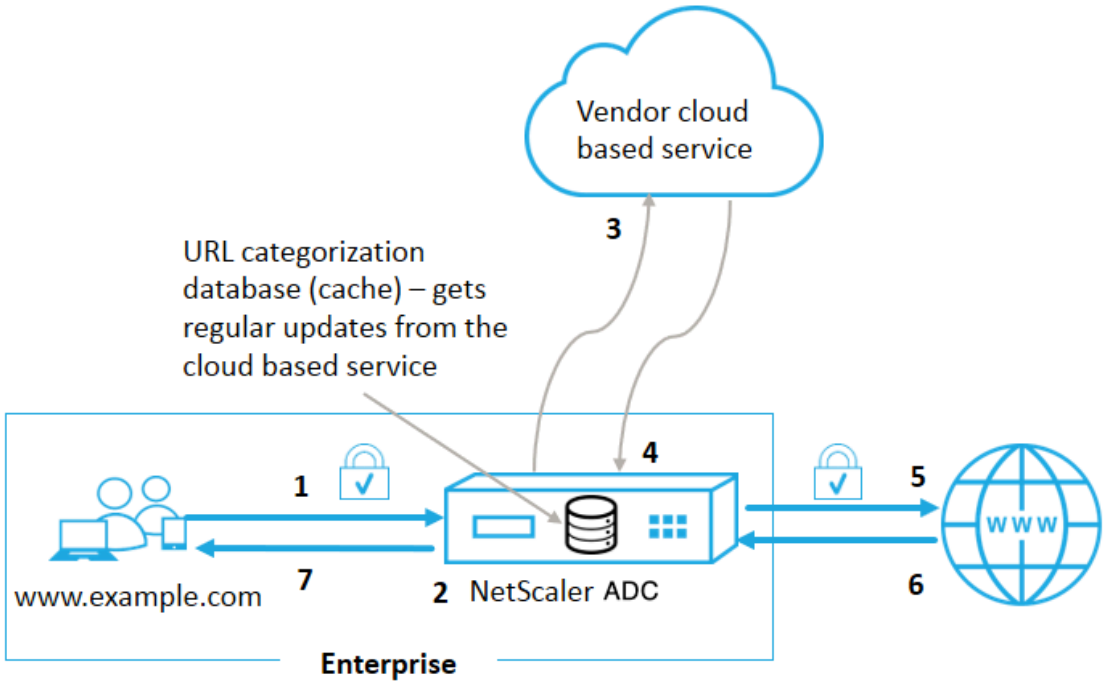
URL Categorization restricts user access to specific websites and website categories. As a subscribed service in collaboration with NetSTAR, the feature enables enterprise customers to filter web traffic using a commercial categorization database. The NetSTAR database has a vast number (billions) of URLs classified into different categories, such as social networking, gambling, adult content, new media, and shopping. In addition to categorization, each URL has a reputation score kept up to date based on the site's historical risk profile. We can use NetSTAR data to filter the traffic by configuring advanced policies based on categories, category groups (such as Terrorism, Illegal drugs), or site-reputation scores.

For example, you might block access to dangerous sites, such as sites known to be infected with malware. You might also selectively restrict access to content such as adult content or entertainment streaming media for enterprise users. You can also capture the user's transactional details and outbound traffic details for monitoring web traffic analytics on the Citrix ADM server.

Citrix ADC uploads or downloads data from the pre-configured NetSTAR device nsv10.netstar-inc.com and incompasshybridpc.netstar-inc.com is used as a cloud host by default for cloud-categorization requests. The appliance uses its NSIP address as a source IP address and 443 as the destination port for communication.

How URL categorization works

The following figure shows how a Citrix ADC URL categorization service is integrated with a commercial URL Categorization database and cloud services for frequent updates.



The components interact as follows:

1. A client sends an internet bound URL request.
2. The SSL forward proxy applies a policy enforcement to the request based on the category details, such as, category, category group, and site-reputation score. The category details are retrieved from the URL categorization database. If the database returns the category details, the process jumps to step 5.
3. If the database misses the categorization details, the request is sent to a cloud-based lookup service maintained by a URL categorization vendor. However, the appliance does not wait for a response, instead, the URL is marked as uncategorized and a policy enforcement is performed (jump to step 5). The appliance continues to monitor the cloud query feedback and updates the cache so that future requests can benefit from the cloud lookup.
4. The ADC appliance receives the URL category details (category, category group, and reputation score) from the cloud-based service and stores it in the categorization database.
5. The policy allows the URL and the request is sent to the origin server. Otherwise, the appliance drops, redirects, or responds with a custom HTML page.
6. The origin server responds with the requested data to the ADC appliance.
7. The appliance sends the response to the client.

Use Case: Internet usage under corporate compliance for enterprises

You can use the URL Filtering feature to detect and implement compliance policies to block sites that violate corporate compliance. For example, sites such as adult, streaming media, social networking which can be deemed nonproductive or consume excess internet bandwidth in an enterprise network. Blocking access to these websites can improve employee productivity, reduce operating costs for bandwidth usage, and reduce the overhead of network consumption.

Prerequisites

The URL Categorization feature works on a Citrix ADC platform only if it has an optional subscription service with URL filtering capabilities and threat intelligence for SSL forward proxy. The subscription allows customers to download the latest threat categorizations for websites and then enforce those categories on the Secure Web Gateway. Before enabling and configuring the feature, you must install the following licenses:

- CNS_WEBF_SSERVER_Retail.lic
- CNS_XXXX_SERVER_PLT_Retail.lic

Where, XXXXX is the platform type, for example: V25000

Responder policy expressions

The following table lists the different policy expressions that you can use to verify if an incoming URL must be allowed, redirected, or blocked.

1. `<text>. URL_CATEGORIZE (<min_reputation>, <max_reputation>)` - Returns a URL_CATEGORY object. If `<min_reputation>` is greater than 0, the returned object does not contain a category with a reputation lower than `<min_reputation>`. If `<max_reputation>` is greater than 0, the returned object does not contain a category with a reputation higher than `<max_reputation>`. If the category fails to resolve in a timely manner, the undef value is returned.
2. `<url_category>. CATEGORY()` - Returns the category string for this object. If the URL does not have a category, or if the URL is malformed, the returned value is “Unknown.”
3. `<url_category>. CATEGORY_GROUP()` - Returns a string identifying the object’s category group. This grouping is a higher level grouping of categories, which is useful in operations that require less detailed information about the URL category. If the URL does not have a category, or if the URL is malformed, the returned value is “Unknown.”
4. `<url_category>. REPUTATION()` - Returns the reputation score as a number from 0 to 5, where 5 indicates the riskiest reputation. If there is the category “Unknown”, the reputation value is 1.

Policy types:

1. Policy to select requests for URLs that are in the Search Engine category - `add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ( "Search Engine" )`
2. Policy to select requests for URLs that are in the Adult category group - `add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY_GROUP().EQ( "Adult" )'`
3. Policy to select requests for Search Engine URLs with a reputation score lower than 4 - `add responder policy p2 'HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL). URL_CATEGORIZE(4,0).HAS_CATEGORY( "Search Engine" )`
4. Policy to select requests for Search Engine and Shopping URLs - `add responder policy p3 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).CATEGORY ().EQ ( "good_categories" )`
5. Policy to select requests for Search Engine URLs with a reputation score equal to or greater than 4 - `add responder policy p5 'CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE (4,0). CATEGORY().EQ( "Search Engines" )`
6. Policy to select requests for URLs that are in the Search Engine category and compare them with a URL Set - `'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0) . CATEGORY().EQ( "Search Engine" )&& HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ. URL).URLSET_MATCHES_ANY( "u1" )'`

Responder policy types

There are two types of policies used in a URL Categorization feature and each of these policy types is explained the following table:

Policy Type	Description
URL Category	Categorize web traffic and based on evaluation result blocks, allows, or redirects traffic.
URL Reputation Score	Determines the reputation score of the website and allows you to control access based on the reputation score threshold level set by the administrator.

Configure URL categorization

To configure URL categorization on a Citrix ADC appliance, do the following:

1. Enable URL filtering.
2. Configure a proxy server for Web traffic.

3. Configure SSL interception for Web traffic in explicit mode.
4. Configure shared memory to limit cache memory.
5. Configure URL categorization parameters.
6. Configure URL categorization by using the Citrix SSL forward proxy wizard.
7. Configure URL categorization parameters by using the SSL forward proxy wizard.
8. Configure seed database path and cloud server name

Step 1: Enabling URL Filtering

To enable URL categorization, enable the URL filtering feature and enable modes for URL categorization.

To enable URL Categorization by using the CLI

At the command prompt, type:

```
enable ns feature URLFiltering
```

```
disable ns feature URLFiltering
```

Step 2: Configure a proxy server for web traffic in explicit mode

The Citrix ADC appliance supports transparent and explicit proxy virtual servers. To configure a proxy virtual server for SSL traffic in explicit mode, do the following:

1. Add a proxy server.
2. Bind an SSL policy to the proxy server.

To add a proxy server by using the CLI

At the command prompt, type:

```
add cs vserver <name> [-td <positive_integer>] <serviceType> [-cltTimeout <
secs>]
```

Example:

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
```

Bind an SSL policy to a proxy virtual server by using the CLI

```
bind ssl vserver <vServerName> -policyName <string> [-priority <positive_integer
>]
```


Step 3: Configure SSL interception for HTTPS traffic

To configure SSL interception for HTTPS traffic, do the following:

1. Bind a CA certificate-key pair to the proxy virtual server.
2. Configure the default SSL profile with SSL parameters.
3. Bind a front-end SSL profile to the proxy virtual server and enable SSL interception in the front-end SSL profile.

To bind a CA certificate-key pair to the proxy virtual server by using the CLI

At the command prompt, type:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -CA -
skipCAName
```

To configure the default SSL profile by using the CLI

At the command prompt, type:

```
set ssl profile <name> -denySSLReneg <denySSLReneg> -sslInterception (
ENABLED | DISABLED)-ssliMaxSessPerServer positive_integer>
```

Bind a front-end SSL profile to a proxy virtual server by using the CLI

At the command prompt, type:

```
set ssl vserver <vServer name> -sslProfile ssl_profile_interception
```

Step 4: Configure shared memory to limit cache memory

To configure shared memory to limit cache memory by using the CLI

At the command prompt, type:

```
set cache parameter [-memLimit <megaBytes>]
```

Where, the memory limit configured for caching is set as 10 MB.

Step 5: Configure URL categorization parameters

To configure the URL categorization parameters by using the CLI

At the command prompt, type:

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>] [-
TimeOfDayToUpdateDB <HH:MM>]
```

Example:

```
set urlfiltering parameter -urlfilt_hours_betweenDB_updates 20
```


Step 6: Configure URL Categorization by using the Citrix SSL forward proxy wizard

1. Log on to the Citrix ADC appliance and navigate to **Security** > **SSL Forward Proxy** page.
2. In the details pane, do one of the following:
   a) Click **SSL Forward Proxy Wizard** to create a new configuration.
   b) Select an existing configuration and click **Edit**.
3. In the **URL Filtering** section, click **Edit**.
4. Select the **URL Categorization** check box to enable the feature.
5. Select a **URL Categorization** policy and Click **Bind**.
6. Click **Continue** and then **Done**.

For more information about URL Categorization policy, see How to Create a URL Categorization Policy.

Step 7: Configuring URL Categorization parameters by using an SSL forward proxy Wizard

1. Log on to **Citrix ADC** appliance and navigate to **Security > URL Filtering**.
2. In the **URL Filtering** page, click **Change URL filtering settings** link.
3. In the **Configuring URL Filtering Params** page, specify the following parameters.
   a) Hours Between DB Updates. URL Filtering hours between database updates. Minimum value: 0 and Maximum value: 720.
   b) Time of Day to Update DB. URL Filtering time of day to update database.
   c) Cloud Host. The URL path of the cloud server.
   d) Seed DB Path. The URL path of the seed database lookup server.
4. Click **OK** and **Close**.

Sample Configuration:

```
enable ns feature LB CS SSL IC RESPONDER AppFlow URLFiltering

enable ns mode FR L3 Edge USNIP PMTUD

set ssl profile ns_default_ssl_profile_frontend -denySSLReneg NONSECURE
    -sslInterception ENABLED -ssliMaxSessPerServer 100

add ssl certKey swg_ca_cert -cert ns_swg_ca.crt -key ns_swg_ca.key

set cache parameter -memLimit 100

add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180

add responder action act1 respondwith "\"HTTP/1.1 200 OK\r\n\r\n\" +
   http.req.url.url_categorize(0,0).reputation + \"\n\""

```



```
15 add responder policy p1 "HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq
   (\"Shopping/Retail\") || HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.
   eq(\"Search Engines & Portals
16
17 \")" act1
18
19 bind cs vserver starcs_PROXY -policyName p1 -priority 10 -
   gotoPriorityExpression END -type REQUEST
20
21 add dns nameServer 10.140.50.5
22
23 set ssl parameter -denySSLReneg NONSECURE -defaultProfile ENABLED -
   sigDigestType RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-
   SHA512 -ssliErrorCache ENABLED
24
25 -ssliMaxErrorCacheMem 100000000
26
27 add ssl policy pol1 -rule "client.ssl.origin_server_cert.subject.
   URL_CATEGORIZE(0,0).CATEGORY.eq(\"Search Engines & Portals\")"" -
   action INTERCEPT
28
29 add ssl policy pol3 -rule "client.ssl.origin_server_cert.subject.ne(\"
   citrix\")" -action INTERCEPT
30
31 add ssl policy swg_pol -rule "client.ssl.client_hello.SNI.
   URL_CATEGORIZE(0,0).CATEGORY.ne(\"Uncategorized\")" -action
   INTERCEPT
32
33 set urlfiltering parameter -HoursBetweenDBUpdates 3 -
   TimeOfDayToUpdateDB 03:00
34 <!--NeedCopy-->
```

Configure seed database path and cloud server name

You can now configure the seed database path and cloud lookup server name for manual setting of the cloud lookup server name and the seed database path. To do this, two new parameters, “CloudHost” and “SeedDBPath”, are added to the URL filtering parameter.

At the command prompt, type:

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>] [-
TimeOfDayToUpdateDB <HH:MM>] [-LocalDatabaseThreads <positive_integer>] [-
CloudHost <string>] [-SeedDBPath <string>]
```

Example:


```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
03:00 -CloudHost localhost -SeedDBPath /mypath
```

The Communication between a Citrix ADC appliance and NetSTAR might require a domain name server. You can test using a simple console or telnet connection from the appliance.

Example:

```
root@ns# telnet nsv10.netstar-inc.com 443
Trying 1.1.1.1...
Connected to nsv10.netstar-inc.com.
Escape character is '^]'.

root@ns# telnet incompasshybridpc.netstar-inc.com 443
Trying 10.10.10.10...
Connected to incompasshybridpc.netstar-inc.com.
Escape character is '^]'.
<!--NeedCopy-->
```

Configure audit log messaging

Audit logging enables you to review a condition or a situation in any phase of the URL Categorization process. When a Citrix ADC appliance receives an incoming URL, if the responder policy has a URL Filtering expression, the audit log feature collects URL Set information in the URL. It stores the information as log messages for any target allowed by audit logging.

- Source IP address (the IP address of the client that made the request).
- Destination IP address (the IP address of the requested server).
- Requested URL containing the schema, the host, and the domain name (http://www.example.com).
- URL category that the URL filtering framework returns.
- URL category group that the URL filtering framework returned.
- URL reputation number that the URL filtering framework returned.
- Audit log action taken by the policy.

To configure audit logging for a URL List feature, you must complete the following tasks:

1. Enable Audit Log.
2. Create Audit Log message action.
3. Set URL List responder policy with Audit Log message action.

For more information, see Audit Logging topic.

Storing failure errors using SYSLOG messaging

At any stage of the URL Filtering process, if there is a system-level failure, the ADC appliance uses the audit log mechanism to store logs in the ns.log file. The errors are stored as text messages in SYSLOG format so that, an administrator can view it later in a chronological order of event occurrence. These logs are also sent to an external SYSLOG server for archival. For more information, see article CTX229399.

For example, if a failure occurs when you initialize the URL Filtering SDK, the error message is stored in the following messaging format.

```
Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: Error initializing
NetStar SDK (SDK error=-1). (status=1).
```

The Citrix ADC appliance stores the error messages under four different failure categories:

- **Download failure**. If an error occurs when you try to download the categorization database.
- **Integration failure**. If an error occurs when you integrate an update into the existing categorization database.
- **Initialization failure**. If an error occurs when you initialize the URL Categorization feature, set categorization parameters, or end a categorization service.
- **Retrieval failure**. If an error occurs when the appliance retrieves the categorization details of the request.

Configure SNMP traps for NetSTAR events

The URL Filtering feature generates SNMP traps, if the following conditions occur:

- NetSTAR database update fails or succeeds.
- NetSTAR SDK initialization fails or succeeds.

The appliance has a set of conditional entities called SNMP alarms. When a condition in the SNMP alarm is met, the appliance generates traps and sends it to a specified trap destination. For example, if the NetSTAR SDK initialization fails, an SNMP OID 1.3.6.1.4.1.5951.1.1.0.183 is generated and sent to the trap destination.

For the appliance to generate traps, you must first enable and configure SNMP alarms. Then, you specify the trap destination to which the appliance sends the generated trap messages

Enable an SNMP alarm

The Citrix ADC appliance generates traps only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them.

When you enable an SNMP alarm, the URL filtering feature generates trap messages when a success or failure event occurs. Some alarms are enabled by default.

To enable an SNMP alarm by using the command line interface:

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
enable snmp alarm <trapName>
show snmp alarm <trapName>
```

To enable an SNMP alarm by using the Citrix ADC GUI

1. Navigate to **System** > **SNMP** > **Alarms**, and select the alarm.
2. Click **Actions** and select **Enable**.

Configure SNMP alarm by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue
 <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

Example:

```
set snmp alarm URL-FIL-DB-UPDATE-STATUS -state ENABLED
set snmp alarm URL-FIL-INIT-SDK -state ENABLED
```

Configure SNMP alarms by using the GUI

Navigate to **System** > **SNMP** > **Alarms**, select an alarm, and configure the alarm parameters.

For more information about SNMP traps, see SNMP topic

URL reputation score

September 14, 2021

The URL Categorization feature provides policy-based control to restrict blacklisted URLs. You can control access to websites based on URL category, reputation score, or URL category and reputation score. If network administrators monitor a user accessing highly risky websites, they can use a responder policy bound to the URL reputation score to block such risky websites.

Upon receiving an incoming URL request, the appliance retrieves the category and reputation score from the URL categorization database. Based on the reputation score returned by the database, the

appliance assigns a reputation rating for websites. The value can range from 1 to 4, where 4 is the riskiest type of websites, as shown in the following table.

URL Reputation Rating	Reputation Comment
1	Clean site
2	Unknown site
3	Potentially dangerous or affiliated to a dangerous site
4	Malicious site

Use Case: Filtering by URL reputation score

Consider an enterprise organization with a network administrator monitoring user transactions and network bandwidth consumption. If malware can enter the network, the administrator must enhance the data security and control access to malicious and dangerous websites accessing the network. To protect the network against such threats, the administrator can configure the URL filtering feature to allow or deny access by URL reputation score.

For more information about monitoring outbound traffic and user activities on the network, see Analytics.

If an employee of the organization tries to access a social networking website, the ADC appliance receives a URL request. It queries the URL Categorization database to retrieve the URL category as social networking and a reputation score 3, which indicates a potentially dangerous website. The appliance then checks the security policy configured by the administrator, such as block access to sites with a reputation rating of 3 or more. It then applies the policy action to control access to the website.

To implement this feature, you must configure the URL reputation score and security threshold levels by using the SSL Forward Proxy wizard.

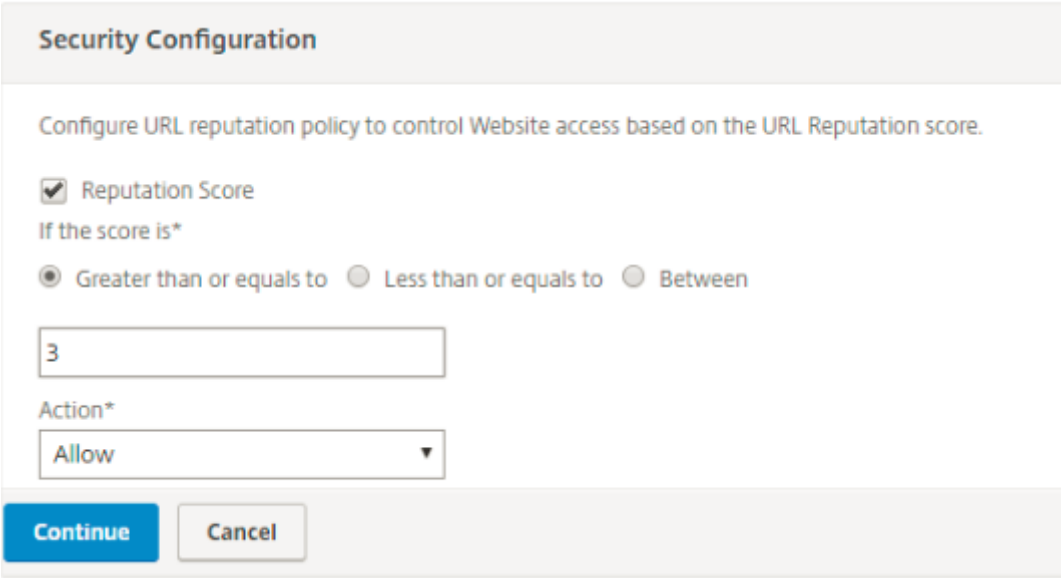
Configure reputation score by using the GUI

Citrix recommends that you use the SSL forward proxy wizard to configure the reputation score and security levels. Based on the configured threshold, you can select a policy action to allow, block, or redirect traffic.

1. Navigate to **Security** > **SSL Forward Proxy**.
2. In the details pane, click **SSL Forward Proxy Wizard**.
3. In the details page, specify the proxy server settings.
4. Click **Continue** to specify other settings such as SSL interception and identify management.

5. Click **Continue** to access the **Security Configuration** section.
6. In the **Security Configuration** section, select the **Reputation Score** check box to control access based on URL reputation score.
7. Select the security level and specify the reputation score threshold value:
   a) Greater than or equals to—Allow or block a website if the threshold value is greater than or equal to N, where N ranges from one to four.
   b) Less than or equals to— Allow or block a website if the threshold value is less than or equal to N, where N ranges from one to four.
   c) In between— Allow or block a website if the threshold value is between N1 and N2 and the range is from one to four.
8. Select a responder action from the drop-down list.
9. Click **Continue** and **Close**.

The following image shows the **Security Configuration** section on the SSL Forward Proxy wizard. Enable the URL Reputation Score option to configure the policy settings.



Analytics

September 14, 2021

In the Citrix ADC appliance, all the user records and subsequent records are logged. When you integrate Citrix Application Delivery Management (ADM) with the Citrix ADC appliance, the logged user activity and the subsequent records in the appliance are exported to Citrix ADM using the `logstream` feature.

Citrix ADM collates and presents information on the activities of users, such as, websites visited and

the bandwidth spent. It also reports bandwidth use and detected threats, such as malware and phishing sites. You can use these key metrics to monitor your network and take corrective actions with the Citrix SWG appliance. For more information, see Citrix SSL Forward Proxy Analytics.

To integrate Citrix ADC appliance with Citrix ADM:

1. In the Citrix ADC appliance, while configuring the SSL forward proxy feature, enable Analytics and provide the details of the Citrix ADM instance that you want to use for analytics.
2. In Citrix ADM, add the Citrix ADC appliance as an instance to Citrix ADM. For more information see Add Instances to Citrix ADM.

Use case: Making an enterprise network secure by using ICAP for remote malware inspection

September 14, 2021

The Citrix ADC appliance acts as a proxy and intercepts all the client traffic. The appliance uses policies to evaluate the traffic and forwards client requests to the origin server on which the resource resides. The appliance decrypts the response from the origin server and forwards the plain text content to the ICAP server for an antimalware check. The ICAP server responds with a message indicating "No adaptation required," or error, or modified request. Depending on the response from the ICAP server, the content requested is either forwarded to the client, or an appropriate message is sent.

For this use case, you must perform some general configuration, proxy and SSL interception related configuration, and ICAP configuration on the Citrix ADC appliance.

General configuration

Configure the following entities:

- NSIP address
- Subnet IP (SNIP) address
- DNS name server
- CA certificate-key pair to sign the server certificate for SSL interception

Proxy server and SSL interception configuration

Configure the following entities:

- Proxy server in explicit mode to intercept all outbound HTTP and HTTPS traffic.
- SSL profile to define SSL settings, such as ciphers and parameters, for connections.

- SSL policy to define rules for intercepting traffic. Set to true to intercept all client requests.

For more details, see the following topics:

- Proxy modes
- SSL interception

In the following sample configuration, the antimalware detection service resides at `www.example.com`.

Sample general configuration:

```
add dns nameServer 203.0.113.2

add ssl certKey ns-swg-ca-certkey -cert ns_swg_ca.crt -key ns_swg_ca.key
<!--NeedCopy-->
```

Sample proxy server and SSL interception configuration:

```
add cs vserver explicitswg PROXY 192.0.2.100 80 -Authn401 ENABLED -authnVsName explicit-auth-vs

set ssl parameter -defaultProfile ENABLED

add ssl profile swg_profile -sslInterception ENABLED

bind ssl profile swg_profile -ssliCACertkey ns-swg-ca-certkey

set ssl vserver explicitswg -sslProfile swg_profile

add ssl policy ssli-pol_ssli -rule true -action INTERCEPT

bind ssl vserver explicitswg -policyName ssli-pol_ssli -priority 100 -type INTERCEPT_REQ
<!--NeedCopy-->
```

Sample ICAP Configuration:

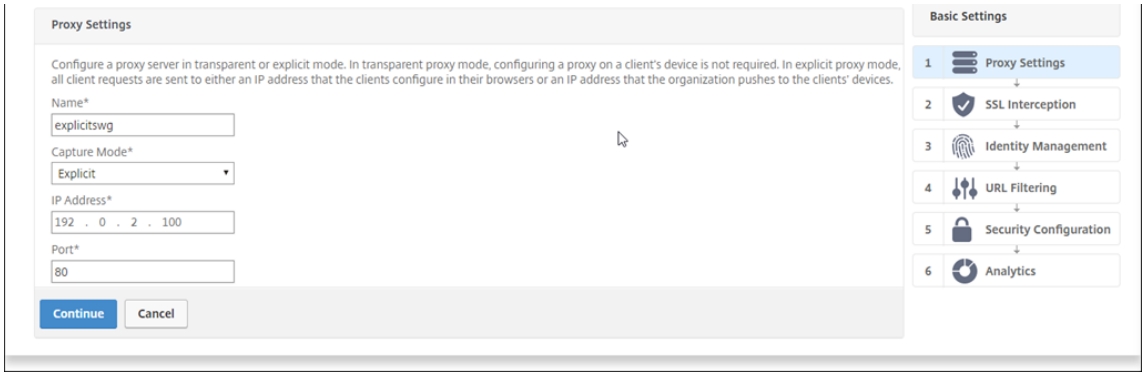
```
add service icap_svc 203.0.113.225 TCP 1344

enable ns feature contentinspection

add icapprofile icapprofile1 -uri /example.com -Mode RESMOD

add contentInspection action CiRemoteAction -type ICAP -serverName icap_svc -icapProfileName icapprofile1
```



```
8
9  add contentInspection policy CiPolicy -rule "HTTP.REQ.METHOD.NE("
       CONNECT")" -action CiRemoteAction
10
11 bind cs vserver explicitswg -policyName  CiPolicy -priority 200 -type
       response
12 <!--NeedCopy-->
```

Configure the proxy settings

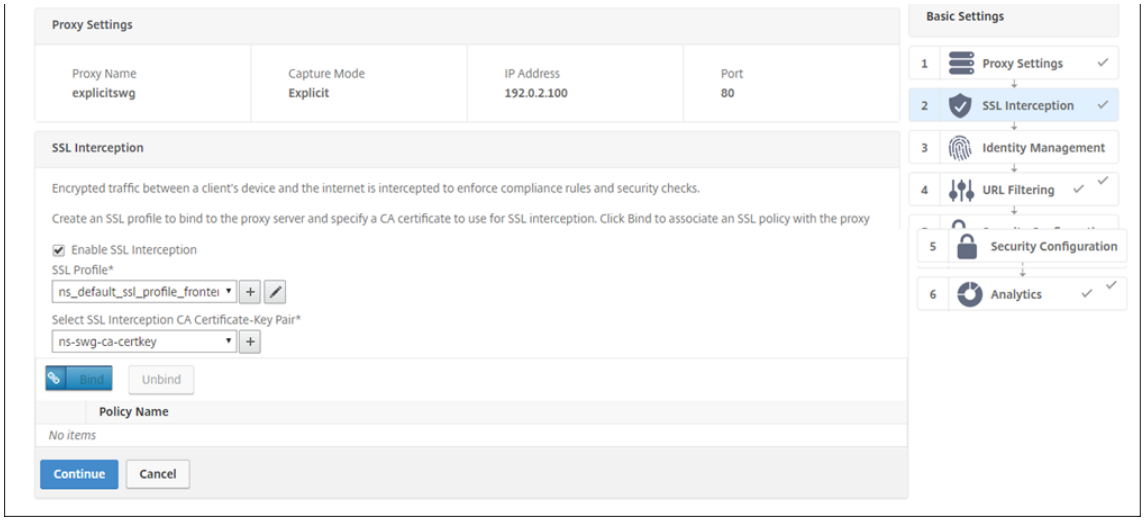
1. Navigate to **Security** > **SSL Forward Proxy** > **SSL Forward Proxy Wizard**.
2. Click **Get Started** and then click **Continue**.
3. In the **Proxy Settings** dialog box, enter a name for the explicit proxy server.
4. For **Capture Mode,** select **Explicit**.
5. Enter an IP address and port number.



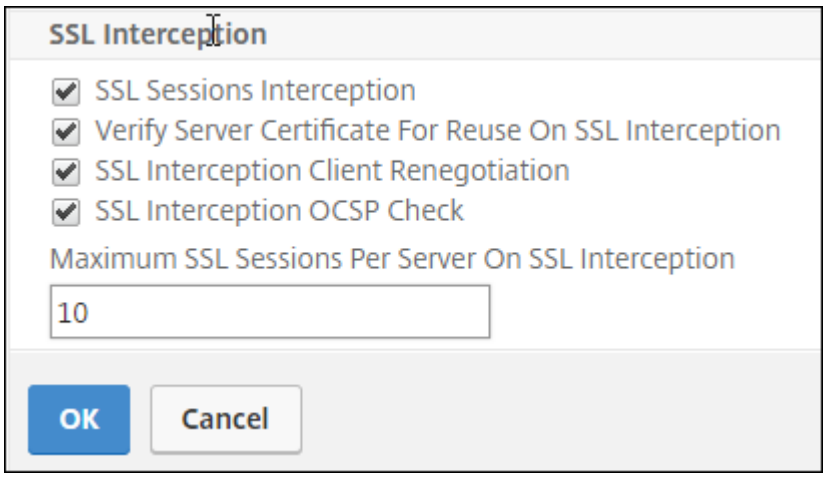
6. Click **Continue**.

Configure the SSL interception settings

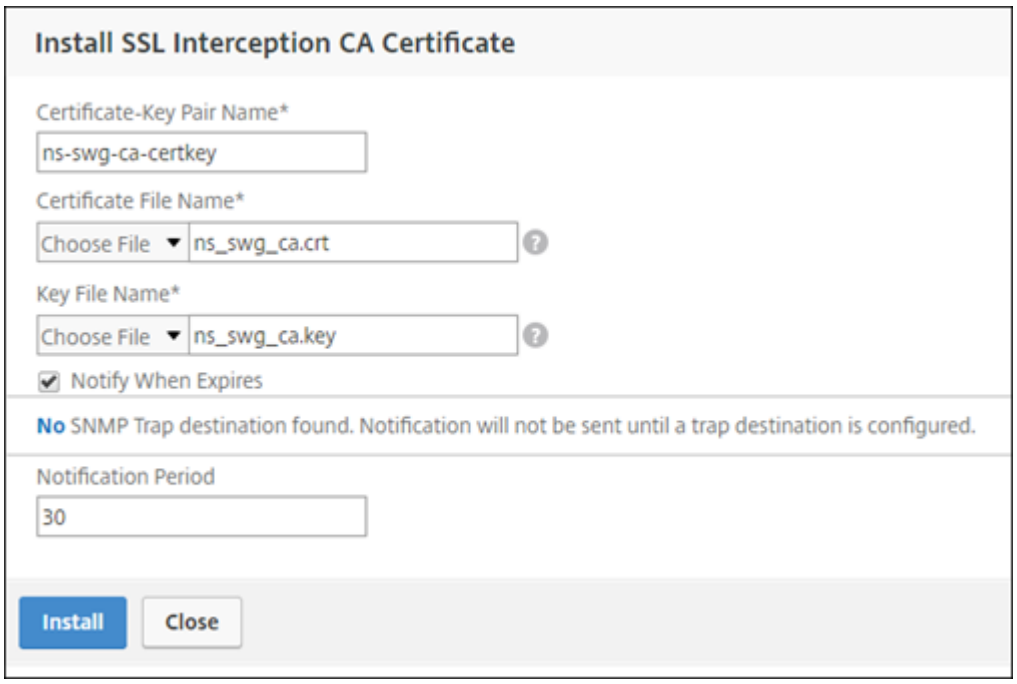
1. Select **Enable SSL Interception**.



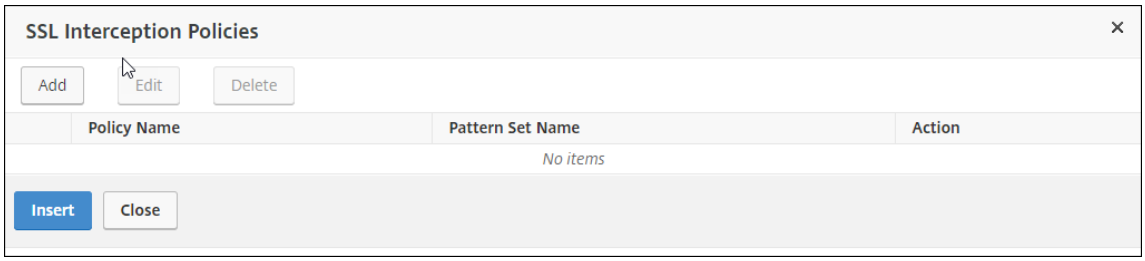
2. In **SSL Profile**, select an existing profile or click "+" to add a new front-end SSL profile. Enable **SSL Sessions Interception** in this profile. If you select an existing profile, skip the next step.



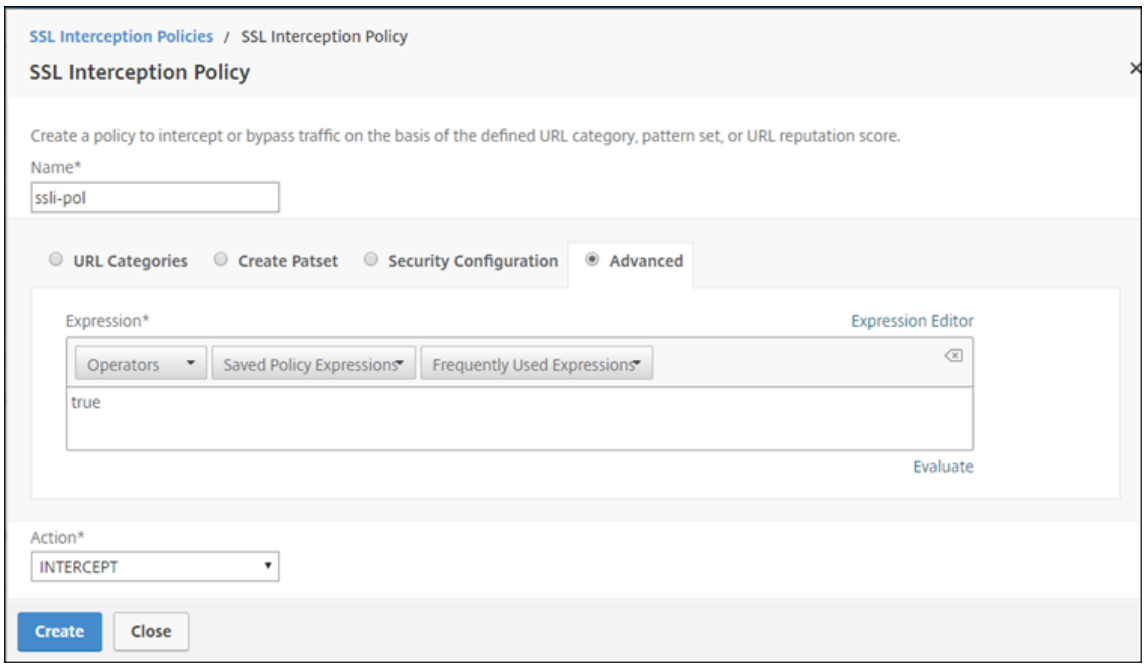
3. Click **OK** and then click **Done**.
4. In **Select SSL interception CA Certificate-Key Pair**, select an existing certificate or click "+" to install a CA certificate-key pair for SSL interception. If you select an existing certificate, skip the next step.



5. Click **Install** and then click **Close**.
6. Add a policy to intercept all the traffic. Click **Bind**. Click **Add** to add a new policy or select an existing policy. If you select an existing policy, click **Insert**, and skip the next three steps.



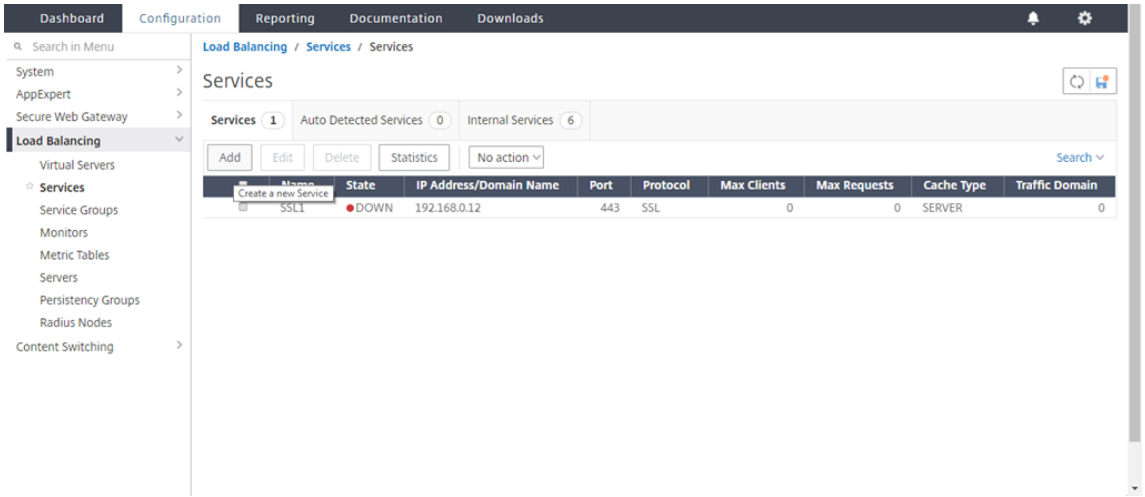
7. Enter a name for the policy and select **Advanced**. In the Expression editor, enter true.
8. For **Action**, select **INTERCEPT**.



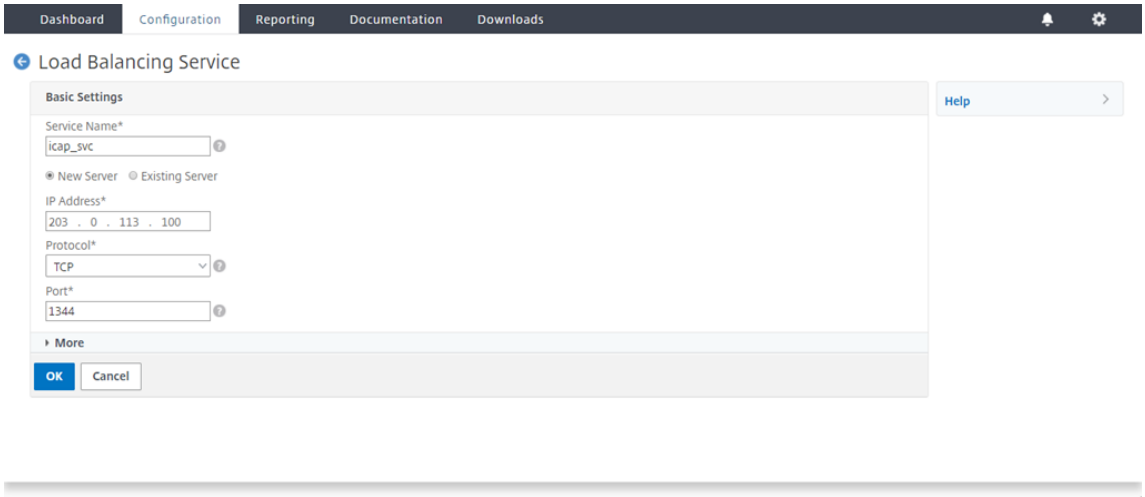
9. Click **Create**.
10. Click **Continue** four times, and then click **Done**.

Configure the ICAP settings

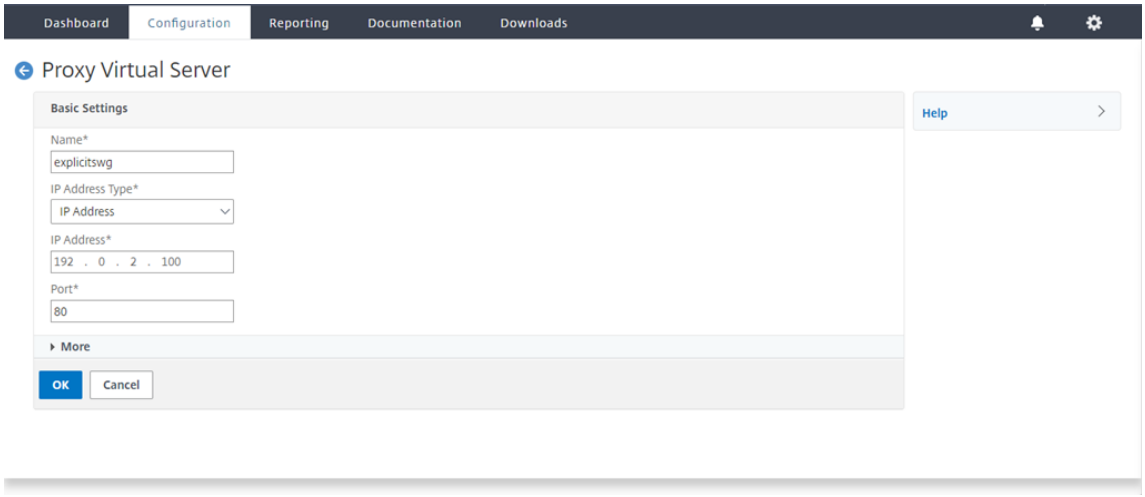
1. Navigate to **Load Balancing** > **Services** and click **Add**.



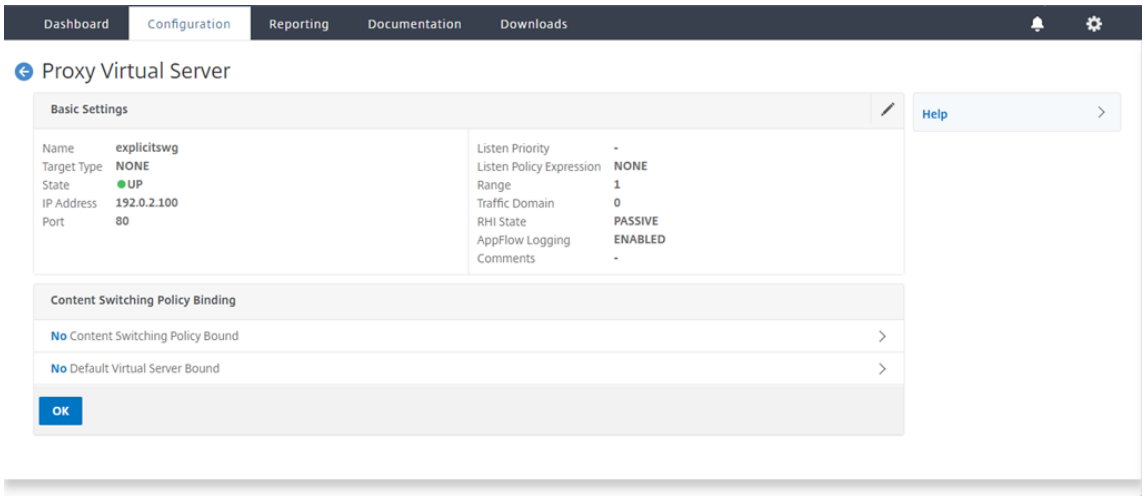
2. Type a name and IP address. In **Protocol**, select **TCP**. In **Port**, type **1344**. Click **OK**.



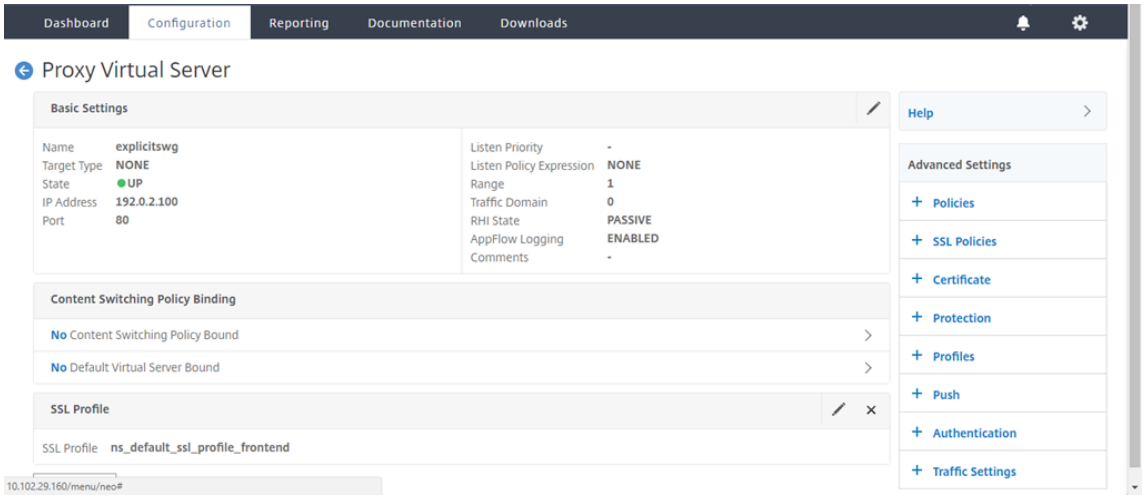
3. Navigate to **SSL Forward Proxy** > **Proxy Virtual Servers**. Add a proxy virtual server or select a virtual server and click **Edit**. After entering details, click **OK**.



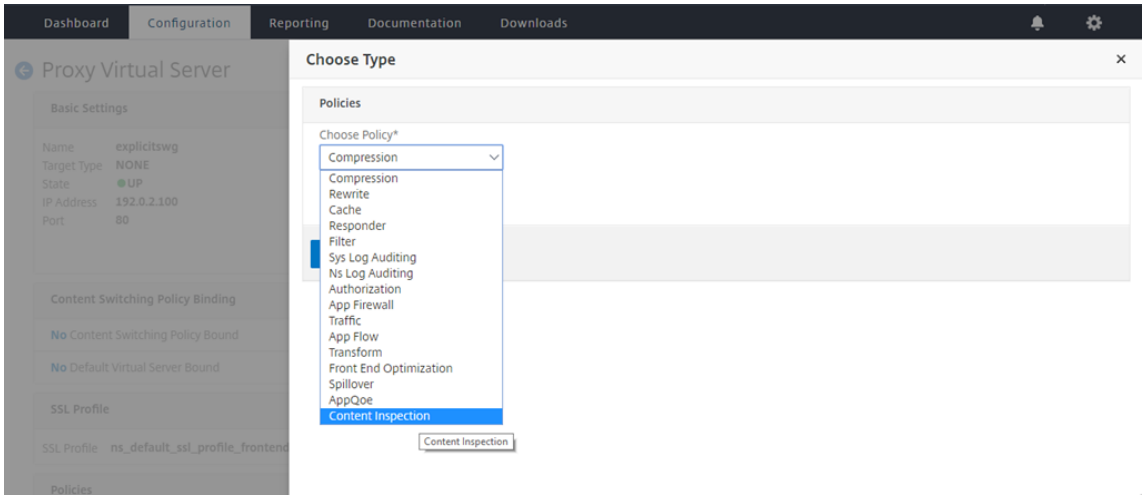
Click **OK** again.



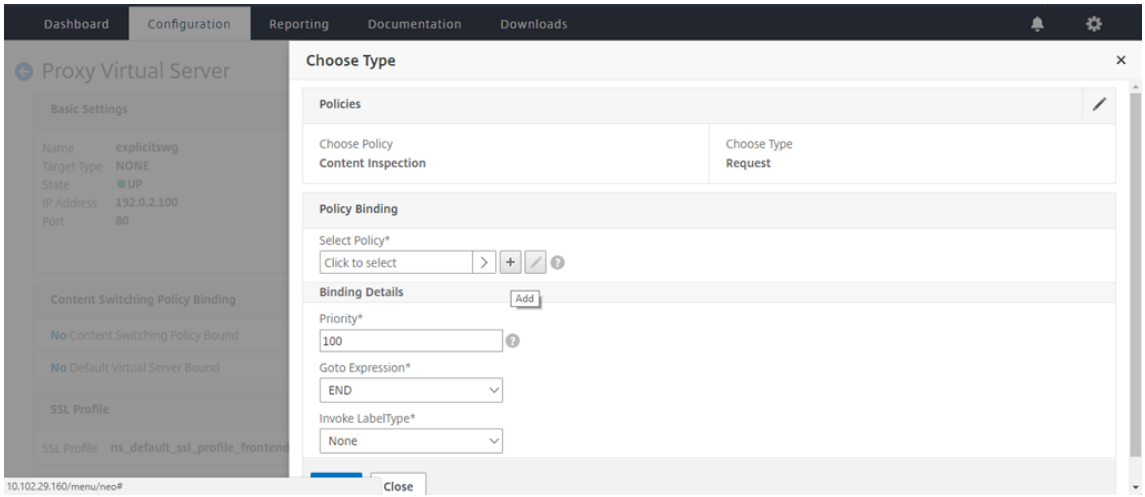
4. In **Advanced Settings**, click **Policies**.



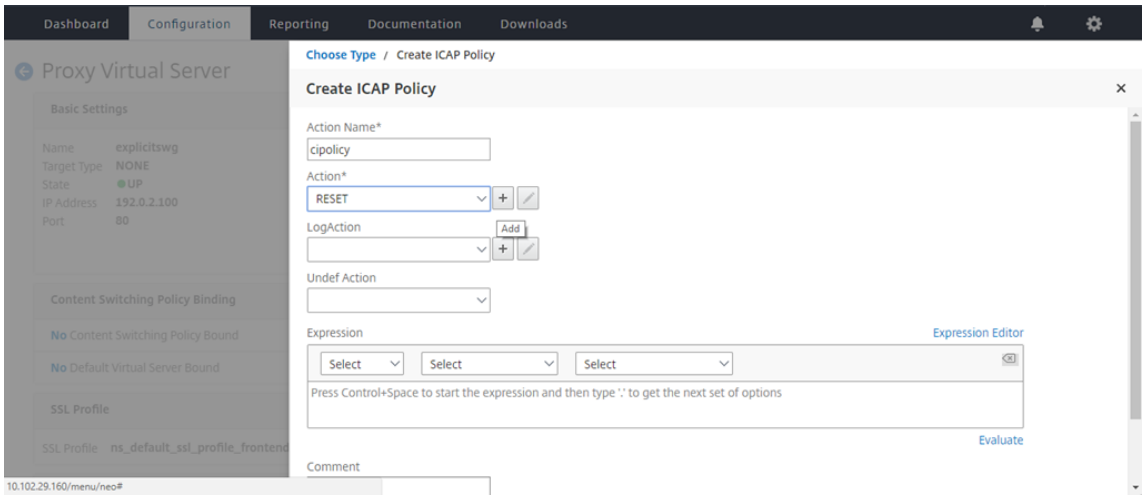
5. In **Choose Policy**, select **Content Inspection**. Click **Continue**.



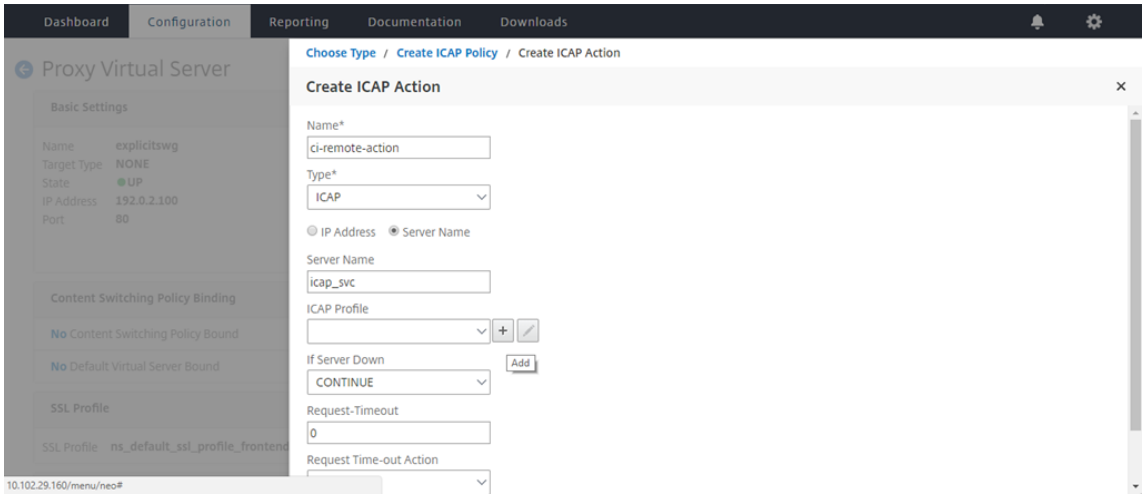
6. In **Select Policy**, click the “+” sign to add a policy.



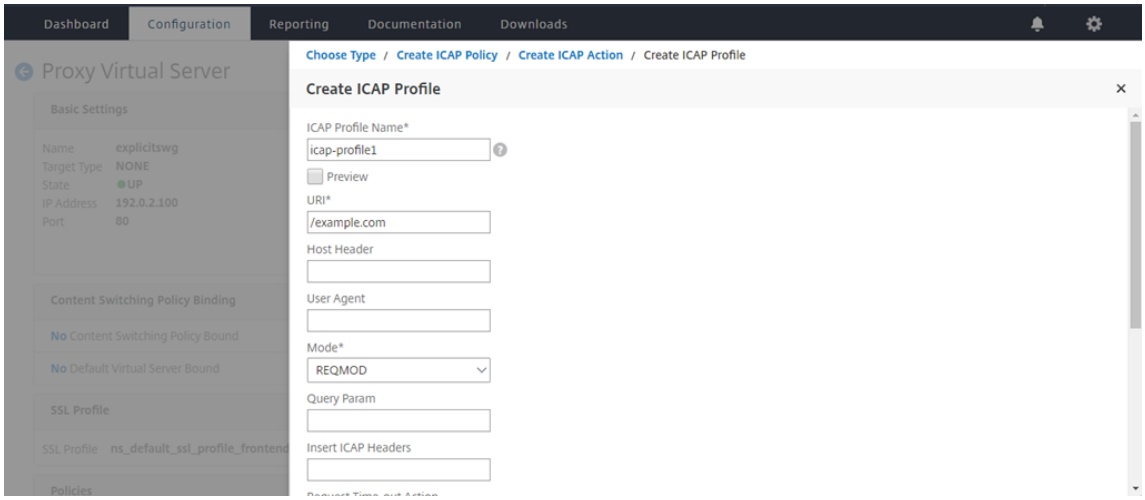
7. Enter a name for the policy. In **Action**, click the “+” sign to add an action.



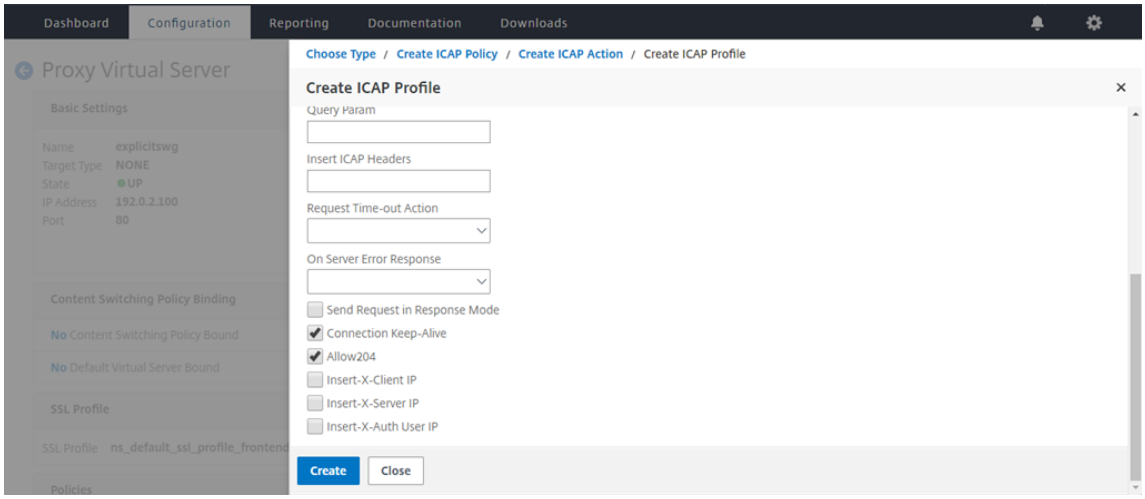
8. Type a name for the action. In **Server Name**, type the name of the TCP service created earlier. In **ICAP Profile**, click the “+” sign to add an ICAP profile.



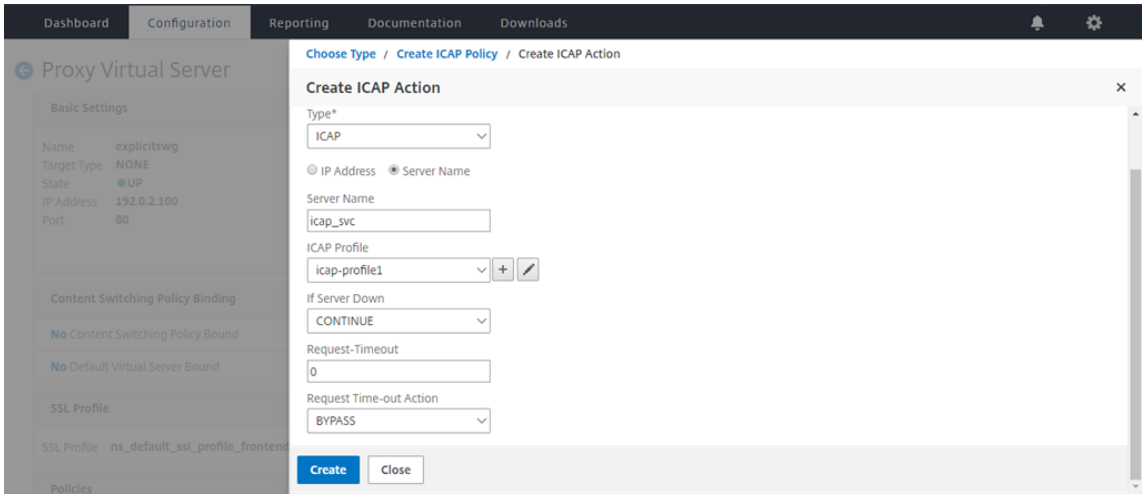
9. Type a profile name, URI. In **Mode**, select **REQMOD**.



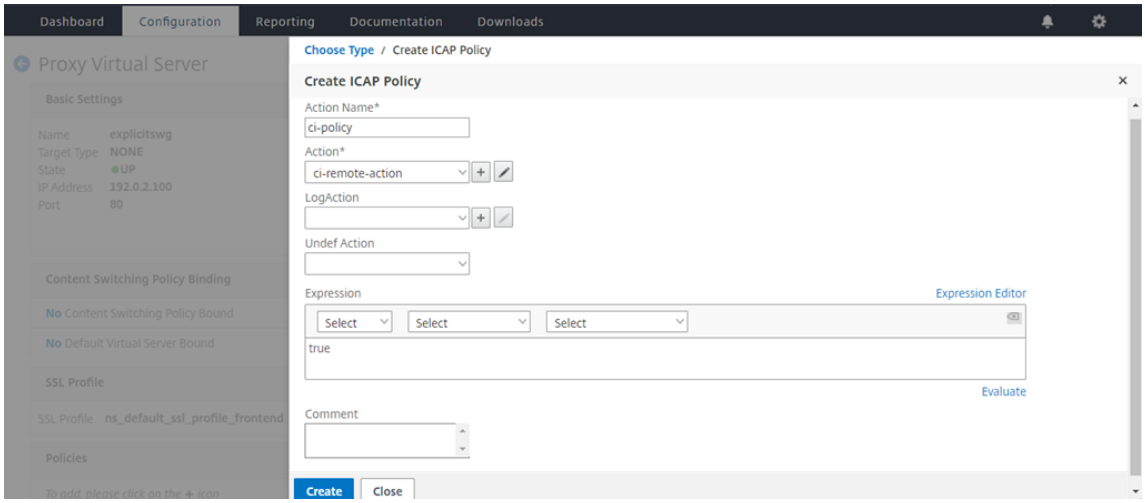
10. Click **Create**.



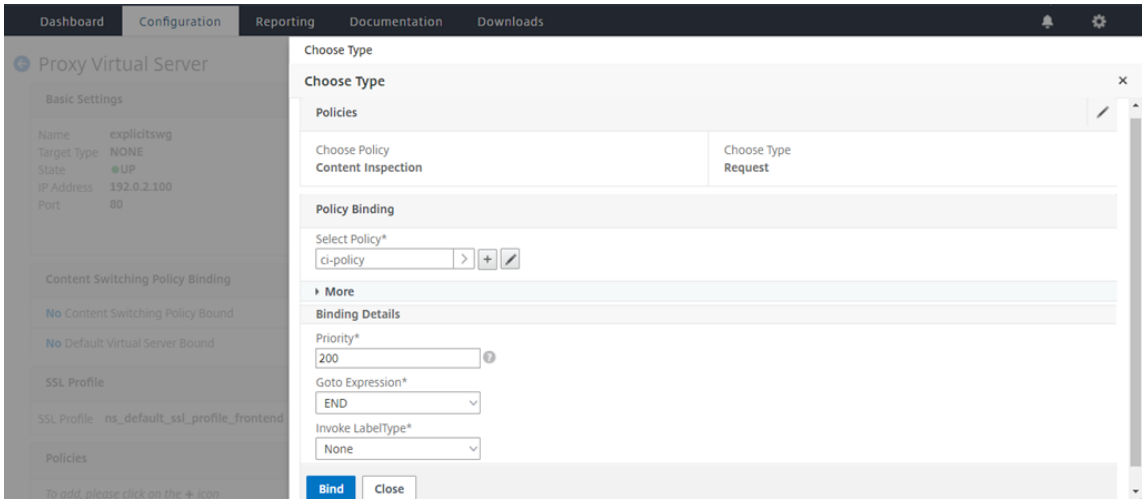
11. In the **Create ICAP Action** page, click **Create**.



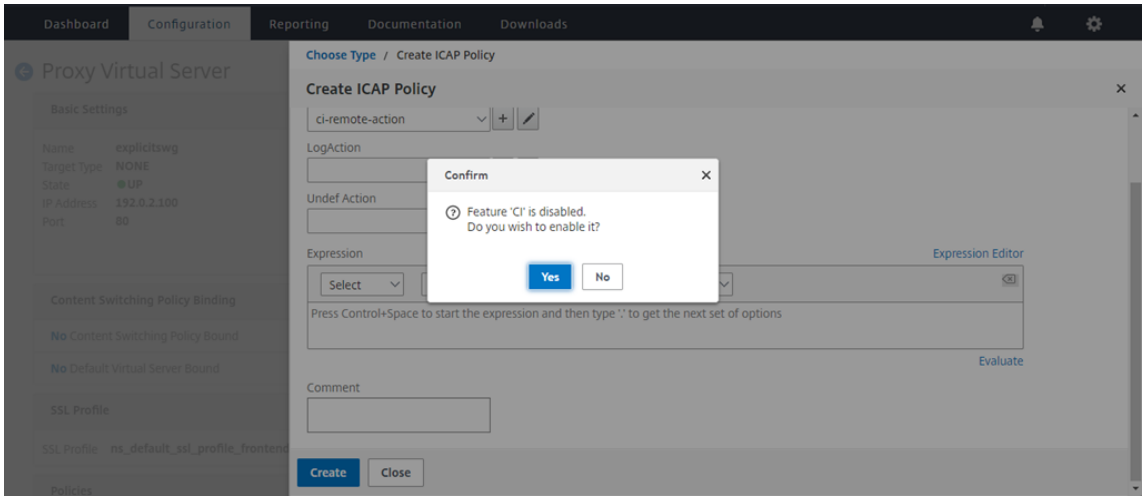
12. In the **Create ICAP Policy** page, enter true in the **Expression Editor**. Then, click **Create**.



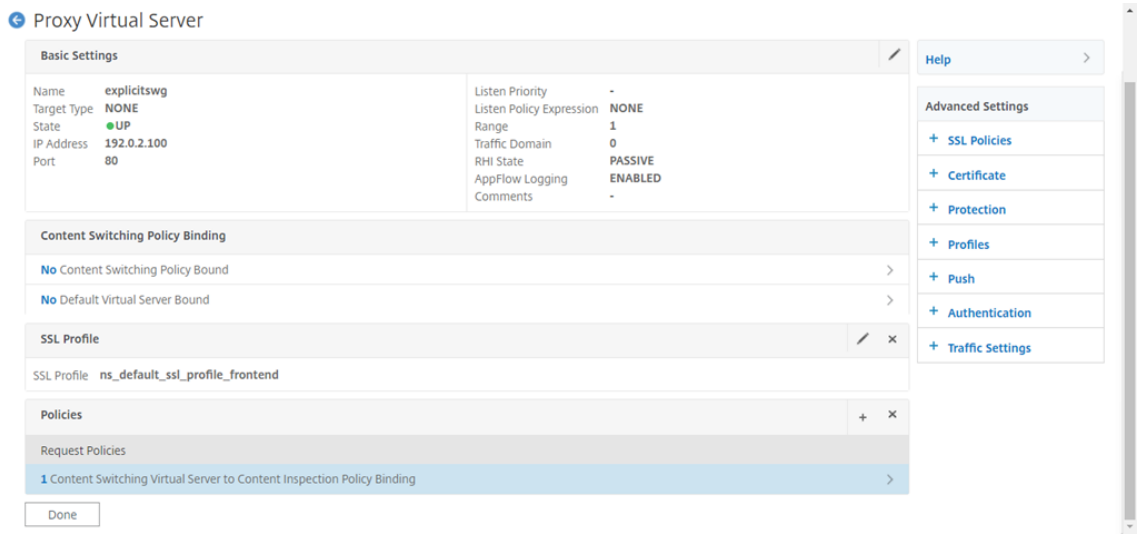
13. Click **Bind**.



14. When prompted to enable the content inspection feature, select **Yes**.



15. Click **Done**.



Sample ICAP transactions between the Citrix ADC appliance and the ICAP server in RESPMOD

Request from the Citrix ADC appliance to the ICAP server:

```
RESPMOD icap://10.106.137.15:1344/resp ICAP/1.0

Host: 10.106.137.15

Connection: Keep-Alive

Encapsulated: res-hdr=0, res-body=282

HTTP/1.1 200 OK

Date: Fri, 01 Dec 2017 11:55:18 GMT

Server: Apache/2.2.21 (Fedora)

Last-Modified: Fri, 01 Dec 2017 11:16:16 GMT

ETag: "20169-45-55f457f42aee4"

Accept-Ranges: bytes

Content-Length: 69

Keep-Alive: timeout=15, max=100

```



```
Content-Type: text/plain; charset=UTF-8

X5O!P%@AP[4PZX54(P^)7CC)7 }
 $EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
<!--NeedCopy-->
```

Response from the ICAP server to the Citrix ADC appliance:

```
ICAP/1.0 200 OK

Connection: keep-alive

Date: Fri, 01 Dec, 2017 11:40:42 GMT

Encapsulated: res-hdr=0, res-body=224

Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$

ISTag: "9.8-13.815.00-3.100.1027-1.0"

X-Virus-ID: Eicar_test_file

X-Infection-Found: Type=0; Resolution=2; Threat=Eicar_test_file;

HTTP/1.1 403 Forbidden

Date: Fri, 01 Dec, 2017 11:40:42 GMT

Cache-Control: no-cache

Content-Type: text/html; charset=UTF-8

Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$

Content-Length: 5688

<html><head><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset
   =UTF-8"/>

…

…

</body></html>
<!--NeedCopy-->
```


How-to articles

September 14, 2021

Following are some configuration instructions or functional use cases available as “How to” articles to help you manage your SSL forward proxy deployment.

URL filtering

How to create a URL categorization policy

How to create a URL list policy

How to allow an exceptional URL

How to block adult category websites

Security

September 14, 2021

The following topics cover configuration and installation information for Citrix ADC security features. Most of these features are policy based.

Content Filtering	Blocks inappropriate HTML requests, preventing the requests from reaching the Web servers.
Surge Protection	Detects any rapid rise in connection attempts and adjusts the rate at which connections are allowed to proceed to the server, preventing server overload.
DNS Security Options	Simplified UI wizard to create policies to protect against DNS attacks.

Content filtering

September 16, 2021

> **Warning:**
>
> Filter actions are deprecated from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use the Responder, Rewrite, or Content Switching features. However, Filter actions are removed and no longer available on the Citrix ADC appliance release 13.1 onwards. For more information, see Responder, Rewrite, or Content Switching topics.

Content filtering can do some of the same tasks as the Citrix Web App Firewall, and is a less CPU-intensive tool. It is limited, however, to examining the header portion of the HTTP request or response and to performing a few simple actions on connections that match. If you have a complex website that makes extensive use of scripts and accesses back-end databases, the Application Firewall might be the better tool for protecting that website. For more information about the Citrix Web App Firewall, see Application Firewall.

Content filtering is based on regular expressions that you can apply to either HTTP requests or HTTP responses. To block requests from a particular site, for example, you can use an expression that compares each request's URL to the URL specified in the expression. The expression is part of a policy, which also specifies an action to be performed on requests or responses that match the expression. For example, an action might drop a request or reset the connection.

Following are some examples of things you can do with content filtering policies:

- Prevent users from accessing certain parts of your websites unless they are connecting from authorized locations.
- Prevent inappropriate HTTP headers from being sent to your Web server, possibly breaching security.
- Redirect specified requests to a different server or service.

To configure content filtering, once you have made sure that the feature is enabled, you configure filtering actions for your servers to perform on selected connections (unless the predefined actions are adequate for your purposes). Then you can configure policies to apply the actions to selected connections. Your policies can use predefined expressions, or you can create your own. To activate the policies you configured, you bind them either globally or to specific virtual servers.

Enabling content filtering

September 14, 2021

By default, content filtering is enabled on Citrix ADC appliances running the Citrix ADC operating system 8.0 or above. If you are upgrading an existing appliance from an operating system version earlier than 8.0, you must update the licenses before you can use content filtering, and you might need to enable the content filtering feature itself manually.

Enable content filtering by using the CLI

At the command prompt, type the following commands to enable content filtering and verify the configuration:

```
-  enable ns feature ContentFiltering
-  show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature ContentFiltering
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   OFF
 .
 .
 .
 .
 11)    Http DoS Protection             HDOSP                OFF
 12)    Content Filtering               CF                   ON
 .
 .
 23)    HTML Injection                  HTMLInjection        ON
 24)    Citrix ADC Push                  push                 OFF
 Done
<!--NeedCopy-->
```

Enable content by filtering by using the GUI

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Configure basic features.
3. In the Configure Basic Features pane, select the Content Filter check box, and then click OK.

Configure a content filtering action

September 14, 2021

After you enable the content filtering feature, you create one or more actions to tell your Citrix ADC appliance how to handle the connections it receives.

Content filtering supports the following actions for HTTP requests:

- **Add:** Adds the specified HTTP header before sending the request to the web server.
- **Reset:** Terminates the connection, sending the appropriate termination notice to the user's browser.
- **Forward:** Redirects the request to the designated service.
- **Drop:** Silently deletes the request, without sending a response to the user's browser.
- **Corrupt:** Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the request to the server.

Content filtering supports the following actions for HTTP responses:

- **Add:** Adds the specified HTTP header before sending the response to the user's browser.
- **ErrorCode:** Returns the designated HTTP error code to the user's browser.
- **Corrupt:** Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the response to the user's browser.

Configure a content filtering action by using the CLI

At the command prompt, type the following commands to configure a Content Filtering action and verify the configuration:

```
1 -  add filter action <name> <qualifier> [<serviceName>] [<value>] [<
     respCode>] [<page>]
2 -  show filter action <name>
3 <!--NeedCopy-->
```

Example:

```
1 > add filter action act_drop Drop
2  Done
3 > show filter action act_drop
4 1)      Name: act_drop   Filter Type: drop
5  Done
6 <!--NeedCopy-->
```


Configure a content filtering action by using the GUI

1. Navigate to **Security > Protection Features > Filter**.
2. In the details pane, do one of the following:
   - To create a new action, click **Add**.
   - To modify an existing action, select the action, and then click **Open**.
3. In the **Add Filter Action** or **Configure Filter Action** dialog box, specify values for the parameters:
   - Action Name*—name
   - Qualifier*—qualifier (Determines which of the following parameters you can configure)
   - Service Name—service name
   - HeaderName:Value—value
   - Response Code—`respcode`
   - Response Page—page
4. Fill in any other required information. For example, if you are configuring an action to send an HTTP error code, you must choose the appropriate error code from a drop-down list. If necessary, you can then modify the text of the error message, which is displayed beneath the drop-down list.
5. Click Create or OK, and then click Close. The Actions list displays the action you configured, and a message in the status bar indicates that your action has been created.

Configure a content filtering policy

September 14, 2021

To implement content filtering, you must configure at least one policy to tell your Citrix ADC appliance how to distinguish the connections you want to filter. You must first have configured at least one filtering action, because when you configure a policy, you associate it with an action.

Content filtering policies examine a combination of one or more of the following elements to select requests or responses for filtering:

- **URL:** The URL in the HTTP request.
- **URL query:** Only the query portion of the URL, which is the portion after the query (?) symbol.
- **URL token:** Only the tokens in the URL, if any, which are the parts that begin with an ampersand (&) and consist of the token name, followed by an equals sign (=), followed by the token value.
- **HTTP method:** The HTTP method used in the request, which is usually GET or POST, but can be any of the eight defined HTTP methods.
- **HTTP version:** The HTTP version in the request, which is usually HTTP 1.1.

- **Standard HTTP header:** Any of the standard HTTP headers defined in the HTTP 1.1 specification.
- **Standard HTTP header value:** The value portion of the HTTP header, which is the portion after the colon and space (:).
- **Custom HTTP header:** A non-standard HTTP header issued by your website or that appears in a user request.
- **Custom header value:** The value portion of the custom HTTP header, which (as with the standard HTTP header) is the portion after the colon and space (:).
- **Client Source IP:** The IP from which the client request was sent.

Content filtering policies use the simpler of two Citrix ADC expressions languages, called classic expressions. For a complete description of classic expressions, how they work, and how to configure them manually, see "Policies and Expressions."

Note: Users who are not experienced in configuring policies at the Citrix ADC command line will usually find using the configuration utility considerably easier.

Configure a content filtering policy by using the CLI

At the command prompt, type the following commands to configure a content filtering policy and verify the configuration:

```
1 -  add filter policy <name> -rule <expression> (-reqAction <action> | -
     resAction <string>
2 -  show filter policy <name>
3 <!--NeedCopy-->
```

Example:

```
1 > add filter policy cf-pol -rule "REQ.HTTP.URL CONTAINS http://abc.com"
      -reqaction DROP
2  Done
3 > show filter policy cf-pol
4 1)      Name: cf-pol    Rule: REQ.HTTP.URL CONTAINS http://abc.com
5         Request action: DROP
6         Response action:
7         Hits: 0
8  Done
9 <!--NeedCopy-->
```


Configure a content filtering policy by using the GUI

1. Navigate to Security > Protection Features > Filter.
2. Navigate to Protection Features > Filter.
3. In the details pane, to create a new policy, click Add.
4. If you are creating a new policy, in the Create Filter Policy dialog box, in the Filter Name text box, type a name for your new policy.
5. Select either Request Action or Response Action to activate the drop-down list to the right of that item.
6. Click the down arrow to the right of the drop-down list and select the action to be performed on the request or response. The default choices are RESET and DROP. Any other actions you have created will also appear in this list.

   Note: You can also click New to create a new Content Filtering action, or Modify to modify an existing Content Filtering action. You can only modify actions you created; the default actions are read-only.
7. If you want to use a predefined expression (or named expression) to define your policy, choose one from the Named Expressions list.
   a) Click the down arrow to the right of the first Named Expressions drop-down list, and choose the category of named expressions that contains the named expression you want to use.
   b) Click the down arrow to the right of the second Named Expressions drop-down list, and choose the named expression you want. As you choose a named expression, the regular expression definition of that named expression appears in the Preview Expression pane beneath the Named Expression list boxes.
   c) Click Add Expression to add that named expression to the Expression list.
   Note: You must perform either this step or step 7, but not both.
8. If you want to create a new expression to define your policy, use the Expression Editor.
   a) Click the Add button. The Add Expression dialog box appears.
   b) In the Add Expression dialog box, choose the type of connection you want to filter. The Flow Type is set to REQ by default, which tells the Citrix ADC appliance to look at incoming connections, or requests. If you want to filter outgoing connections (responses), you click the right arrow beside the drop-down list and choose RES.
   c) If the Protocol is not already set to HTTP, click the down arrow to the right of the Protocol drop-down list and choose HTTP.
   Note: In the Citrix ADC classic expressions language, “HTTP” includes HTTPS requests, as well.
   d) Click the down arrow to the right of the Qualifier drop-down list, and then choose a qualifier for your expression. Your choices are:
      - **METHOD:** The HTTP method used in the request.
      - **URL:** The contents of the URL header.

- **URLTOKENS:** The URL tokens in the HTTP header.
- **VERSION:** The HTTP version of the connection.
- **HEADER:** The header portion of the HTTP request.
- **URLLEN:** The length of the contents of the URL header.
- **URLQUERY:** The query portion of the contents of the URL header.
- **URLQUERYLEN:** The length of the query portion of the URL header.

  The contents of the remaining list boxes change to the choices appropriate to the Qualifier you pick. For example, if you choose HEADER, a text field labeled Header Name* appears below the Flow Type list box.

e) Click the down arrow to the right of the Operator drop-down list, and choose an operator for your expression. Your choices will vary depending on the Protocol you chose in the preceding step. The following list includes all of the operators:
   - **==:** Matches the following text string exactly.
   - **!=:** Does not exactly match the following text string.
   - **>:** Is greater than the following integer.
   - **CONTAINS:** Contains the following text string.
   - **CONTENTS:** The contents of the designated header, URL, or URL query.
   - **EXISTS:** The specified header or query exists.
   - **NOTCONTAINS:** Does not contain the following text string.
   - **NOTEXISTS:** The specified header or query does not exist.

f) If the Value text box is visible, type the appropriate string or number. If you are testing a string in any way, type the string into the Value text box. If you are testing an integer in any way, type the integer into the Value text box.

g) If you chose HEADER as the Protocol, type the header you want in the Header Name* text box.

h) Click OK to add your expression to the Expressions list.

i) Repeat steps B through H to create any additional expressions you want for your profile.

j) Click Close to close the Expressions Editor.

9. If you created a new expression, in the Expression frame select an option from the Match Any Expression drop-down list. Your choices are:
   - Match Any Expression. If a request matches any expression in the Expressions list, the request matches this policy.
   - Match All Expressions If a request matches all expressions in the Expressions list, the request matches this policy. If it does not match all of them, it does not match this policy.
   - Tabular Expression Switches the Expressions list to a tabular format with three columns. In the first column you can place a BEGIN [(] operator. The second column contains the expressions you have selected or created. In the third column, you can place any of the other operators in the following list, to create complex policy groups in which each group can be configured for match any expression or match all expressions.

- The AND [&&] operator tells the appliance to require that a request match both the current expression and the following expression.
- The OR [||] operator tells the appliance to require that a request match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.
- The END [)] operator tells the appliance that this is the last expression in this expression group or policy.
  Note: The Tabular format allows you to create a complex policy that contains both “Match Any Expression” and “Match All Expressions” on a per-expression basis. You are not limited to just one or the other.
- Advanced Free-Form Switches off the Expressions Editor entirely and modifies the Expressions list into a text area. In the text area, you can type the PCRE-format regular expression of your choice to define this policy. This is both the most powerful and the most difficult method of creating a policy, and is recommended only for those thoroughly familiar with the Citrix ADC appliance and PCRE-format regular expressions.
  Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that is what you want.

10. Repeat steps 6 through 8 to add any additional expressions you want to the Expressions list. You can mix named expressions and expressions created in the Expressions Editor. To the Citrix ADC appliance, they are all the same.
11. Click Create to create your new policy. Your new policy appears in the Policies pane list.
12. Click **Close**. To create additional Content Filtering policies, repeat the previous procedure. To remove a Content Filtering policy, select the policy in the **Policies** tab and click **Remove**.

Binding a content filtering policy

September 14, 2021

You must bind each content filtering policy to put it into effect. You can bind policies globally or to a particular virtual server. Globally bound policies are evaluated each time traffic directed to any virtual server matches the policy. Policies bound to a specific virtual server are evaluated only when that virtual server receives traffic that matches the policy.

Bind a policy to a virtual server by using the CLI

At the command prompt, type the following commands to bind a policy to a virtual server and verify the configuration:


```
- bind lb vserver <name>@ -policyName <string> -priority <
   positive_integer>
- show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver vs-loadbal
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

 Done
<!--NeedCopy-->
```

Globally bind a policy by using the CLI

At the command prompt, type the following commands to globally bind a policy and verify the configuration:

```
- bind filter global (<policyName> [-priority <positive_integer>]) [-
   state ( ENABLED | DISABLED )]
- show filter global
<!--NeedCopy-->
```

Example:


```
 bind filter global cf-pol -priority 1
 Done show filter global
1)      Policy Name: cf-pol     Priority: 1
 Done
<!--NeedCopy-->
```

Bind a policy to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. In the details pane, select the virtual server to which you want to bind the content filtering policy from the list, and click **Open**.
3. In the **Configure Virtual Server** (Load Balancing) dialog box, select the **Policies** tab, and then select the check box in the Active column of the filter policy that you want to bind to the virtual server.
4. Click **OK**. The policies you have bound display a check mark and the word Yes in the Policies Bound column of the Policies tab.

Globally bind a policy by using the GUI

1. Navigate to **Security > Protection Features > Filter**.
2. In the details pane, in the **Policies** tab, select the policy that you want to bind, and then click **Global Bindings**.
3. In the Bind/Unbind Filter Policies dialog box, in the **Policy Name** drop-down list, select a policy, and then click **Add**. The policy is added to the Configured list.

   > **Note**
   >
   > To select multiple policies from the list, drag the Ctrl key, then click each policy you want.

4. Click **OK**, and then click **Close**. The policies you have bound display a check mark and the word Yes in the Globally Bound column of the Policies tab.

Configuring content filtering for a commonly used deployment scenario

September 14, 2021

This example provides instructions for using the configuration utility to implement a content filtering policy in which, if a requested URL contains root.exe or cmd.exe, the content filtering policy `filter-CF-nimda` is evaluated and the connection is reset.

To configure this content filtering policy, you must do the following:

- Enable content filtering
- Configure content filtering policy
- Bind content filtering policy globally or to a virtual server
- Verify the configuration

Note: Since this example uses a default content filtering action, you do not need to create a separate content filtering action.

Enable content filtering

1. In the navigation pane, expand System, and click Settings.
2. In the details pane, under Modes & Features, click Change Basic Features.
3. In the Configure Basic Features dialog box, select the Content Filtering check box, and then click OK.
4. In the Enable/Disable feature(s) dialog box, click Yes. A message appears in the status bar, stating that the selected feature is enabled.

Configure the content filtering policy `filter-CF-nimda`

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, click Add. The Create Filter Policy dialog box appears.
3. In the Create Filter Policy dialog box, in the Filter Name text box, type the name `filter-CF-nimda`.
4. Select the Request Action option, and in the drop-down list, select RESET.
5. In the Expression frame, select Match Any Expression from the drop-down list, and then click Add.
6. In the Add Expression dialog box, Expression Type drop-down list, select General.
7. In the Flow Type drop-down list, select REQ.
8. In the Protocol drop-down list, select HTTP.
9. In the Qualifier drop-down list, select URL.
10. In the Operator drop-down list, select CONTAINS.
11. In the Value text box, type cmd.exe, and then click OK. The expression is added in the Expression text box.
12. To create another expression, repeat Steps 7 through 11, but in the Value text box, type root.exe. Then click OK, and finally click Close.
13. Click Create on the Create Filter Policy dialog box. The filter policy `filter-CF-nimda` appears in the Filter list.
14. Click Close.

Globally bind the content filtering policy

1. Navigate to Security > Protection Features > Filter. The Filter page appears in the right pane.
2. In the details pane, Policies tab, select the policy that you want to bind and click Global Bindings. The Bind/Unbind Filter Policies dialog box appears.
3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select the policy `filter-CF-nimda`, and click Add. The policy is added to the Configured list.
4. Click OK, and then click Close. The policy you have bound displays a check mark and Yes in the Globally Bound column of the Policies tab.

Bind the content filtering policy to a virtual server

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane virtual servers list, select vserver-CF-1 to which you want to bind the content filtering policy and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab.
4. In the Active column, select the check box for the policy `filter-CF-nimda`, and then click OK. Your content filtering policy is now active, and must be filtering requests. If it is functioning correctly, the select counter is incremented every time there is a request for a URL containing either root.exe or cmd.exe. This allows you to confirm that your content filtering policy is working. The content filtering policy is bound to the virtual server.

Verify the content filtering configuration by using the command line interface

At the command prompt, type the following command to verify the content filtering configuration:

```
show filter policy filter-CF-nimda
```

Example:

```
1 sh filter policy filter-CF-nimda
2         Name: filter-CF-nimda   Rule: REQ.HTTP.URL CONTAINS cmd.exe ||
             REQ.HTTP.URL CONTAINS root.exe
3         Request action: RESET
4         Response action:
5         Hits: 0
6 Done
7 <!--NeedCopy-->
```

> **Note**
>
> The select counter displays an integer that denotes the number of times the `filter-CF-nimda` policy is evaluated. In the preceding steps, the select counter is set to zero because no requests

> for a URL containing either cmd.exe or root.exe has been made yet. If you want to see the counter increment in real time, you can simply request a URL that contains either of these strings.

Verify the content filtering configuration by using the GUI

1. Navigate to **Security > Protection Features > Filter**.
2. In the details pane, select the filter policy `filter-CF-nimda`. The bottom of the pane must display the following:

```
    **Request Action:**

    RESET

    **Rule:**

    REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe

    **Hits:**

    0
<!--NeedCopy-->
```

Troubleshooting

September 14, 2021

If the content filtering feature does not work as expected after you have configured it, you can use some common tools to access Citrix ADC resources and diagnose the problem.

Resources for troubleshooting

You can use the following tools and resources to troubleshoot most Content Filtering issues on a Citrix ADC appliance:

- The Wireshark application customized for the Citrix ADC trace files
- Trace files recorded when accessing the resource
- The configuration files
- The ns.log file
- The `iehttpheaders`, or a Fiddler trace or a similar utility

Troubleshooting content filtering issues

To troubleshoot a content filtering issue, proceed as follows:

- Verify that the feature is enabled.
- Verify that the content filtering policy is configured correctly. Pay special attention to the expression that evaluates the incoming requests.

  > **Note**
  >
  > Most content filtering issues are caused by incorrect configuration, and the error is most often in the policy configuration.

- Check the policy's select counter to verify that it is incrementing. If it is not, the policy is not getting evaluated.
- If the policy is getting evaluated and the required filtering is still not performed, you need to look into the policy expressions and action.
- If the policy's expression seems valid, test it by assigning a simple NSTRUE value to see if the evaluation of the expression is creating any issue.
- Reevaluate whether the filtering must be based on the request or the response.
- Verify that the action is configured correctly. For example, if a custom action is used to corrupt a header in the request, verify that the header name in the action is correct. If you are not sure about the header name, start a browser with `iehttpheaders` or a similar utility, and then verify the headers in the request. When this feature is used, you can use ns trace to find out if appropriate action is performed when the packets leave Citrix ADC appliance.
- An `iehttpheaders` or Fiddler trace can help you find header options and names, client-side request headers, and response headers recorded on the client.
- To check the modifications made to the request header, record an ns trace on the Citrix ADC appliance or a Wireshark trace on the server.
- If none of the above measures resolves the issue, verify that the connection has not become untrackable, which can happen in certain circumstances. If a connection becomes untrackable, the appliance does not perform any application-level processing of the requests. In that event, contact Citrix Technical Support.

Surge protection

September 14, 2021

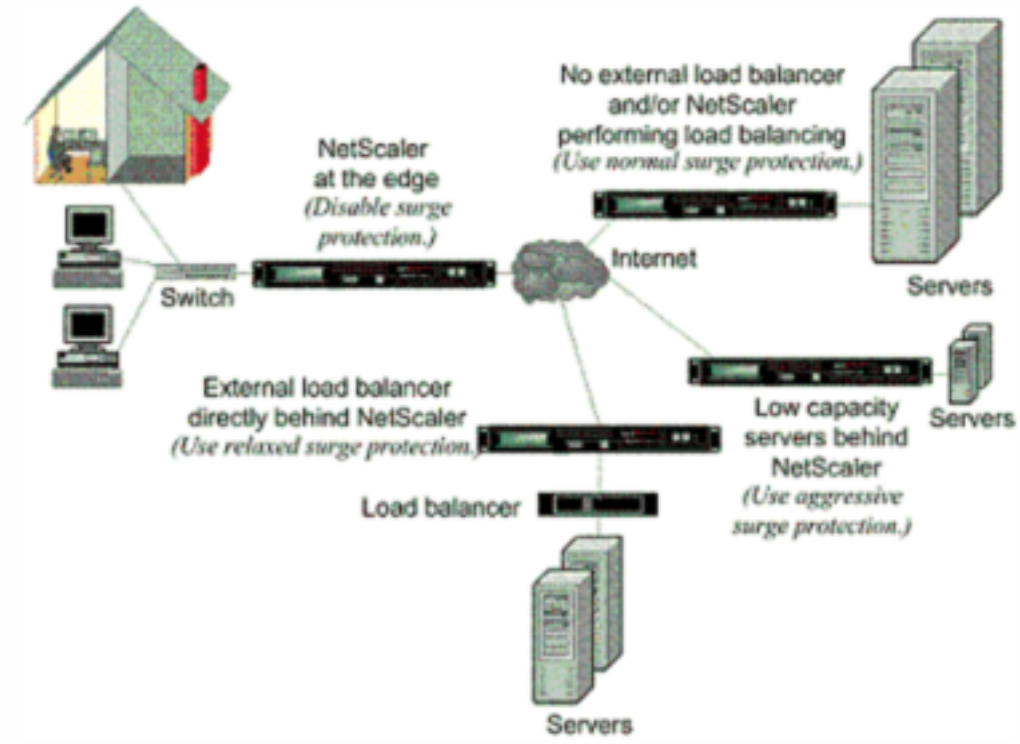
When a surge in client requests overloads a server, server response becomes slow, and the server is unable to respond to new requests. The Surge Protection feature ensures that connections to the server occur at a rate that the server can handle. The response rate depends on how surge protection is configured. The Citrix ADC appliance also tracks the number of connections to the server, and uses that information to adjust the rate at which it opens new server connections.

Surge protection is enabled by default. If you do not want to use surge protection, as is the case with some special configurations, you must disable it.

The default surge protection settings are sufficient for most uses, but you can configure surge protection to tune it for your needs. First, you can set the throttle value to tell it how aggressively to manage connection attempts. Second you can set the base threshold value to control the maximum number of concurrent connections that the Citrix ADC appliance allows before triggering surge protection. (The default base threshold value is set by the throttle value, but after setting the throttle value you can change it to any number you want.)

The following figure illustrates how surge protection is configured to handle traffic to a website.

Figure 1. A Functional Illustration of Citrix ADC Surge Protection



Note

If the Citrix ADC appliance is installed at the edge of the network, where it interacts with network devices on the client side of the Internet, the surge protection feature must be disabled. Surge protection must also be disabled if you enable USIP (Using Source IP) mode on your appliance.

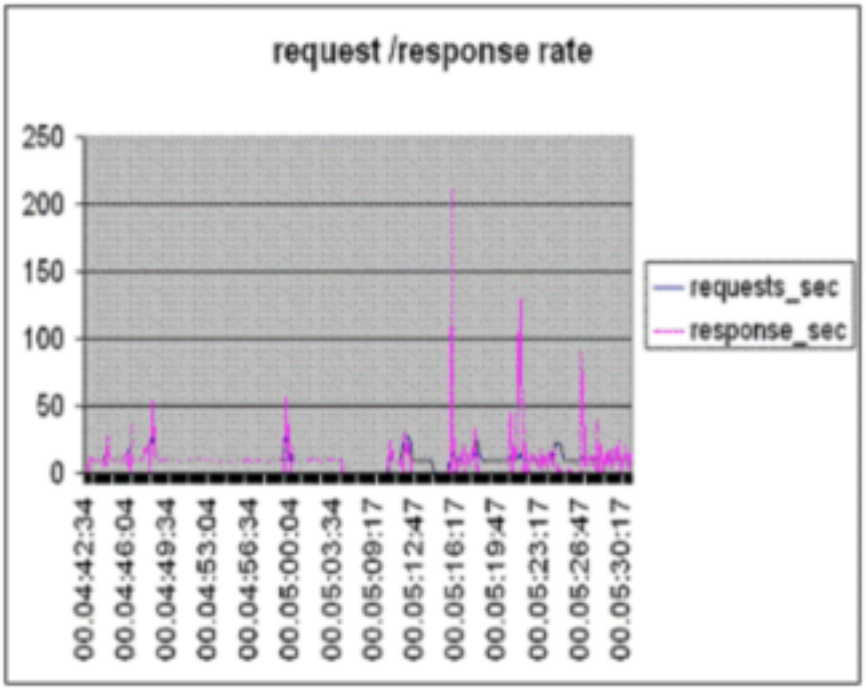
The following example and illustration show the request and response rates for two cases. In one case, surge protection is disabled, and in the other it is enabled.

When surge protection is disabled and a surge in requests occurs, the server accepts as many requests as it can process concurrently, and then begins to drop requests. As the server becomes more overloaded, it goes down and the response rate is reduced to zero. When the server recovers from the crash, several minutes later, it sends resets for all pending requests, which are abnormal behavior, and also responds to new requests with resets. The process repeats for each surge in requests. Therefore, a server that is under DDoS attack and receives multiple surges of requests can become unavailable to legitimate users.

When surge protection is enabled and a surge in requests occurs, surge protection manages the rate of requests to the server, sending requests to the server only as fast as the server can handle those requests. This enables the server to respond to each request correctly in the order it was received. When the surge is over, the backlogged requests are cleared as fast as the server can handle them, until the request rate matches the response rate.

The following figure compares the request and response scenarios when surge protection is enabled to that when it is disabled.

Figure 2. Request/Response Rate with and without Surge Protection



Disable and reenable surge protection

September 14, 2021

The surge protection feature is enabled by default. When surge protection is enabled, it is active for any service that you add.

Disable or reenable surge protection by using the CLI

At the command prompt, type one of the following sets of commands to disable or reenable surge protection and verify the configuration:

```
- disable ns feature SurgeProtection
- show ns feature
- enable ns feature SurgeProtection
- show ns feature
<!--NeedCopy-->
```

Example:

```
disable ns feature SurgeProtection
 Done show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   OFF
 .
 .
 .
 23)    HTML Injection                  HTMLInjection        ON
 24)    Citrix ADC Push                  push                 OFF
 Done
<!--NeedCopy-->
```

```
 enable ns feature SurgeProtection
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
```



```
10   .
11   .
12
13   23)    HTML Injection                HTMLInjection        ON
14   24)    Citrix ADC Push                push                 OFF
15   Done
16  >
17  <!--NeedCopy-->
```

Disable or reenable surge protection by using the GUI

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Change Advanced Features**.
3. In the **Configure Advanced Features** dialog box, clear the selection from the **Surge Protection** check box to disable the surge protection feature, or select the check box to enable the feature.
4. Click **OK**.
5. In the Enable/Disable Features dialog box, click Yes. A message appears in the status bar, stating that the feature has been enabled or disabled.

Disable or reenable surge protection for a particular service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services. The list of configured services is displayed in the details pane.
2. In the details pane, select the service for which you want to disable or reenable the surge protection feature, and then click Open.
3. In the Configure Service dialog box, click the Advanced tab and scroll down.
4. In the Others frame, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
5. Click OK. A message appears in the status bar, stating that the feature has been enabled or disabled.
   Note: Surge protection works only when both the feature and the service setting are enabled.

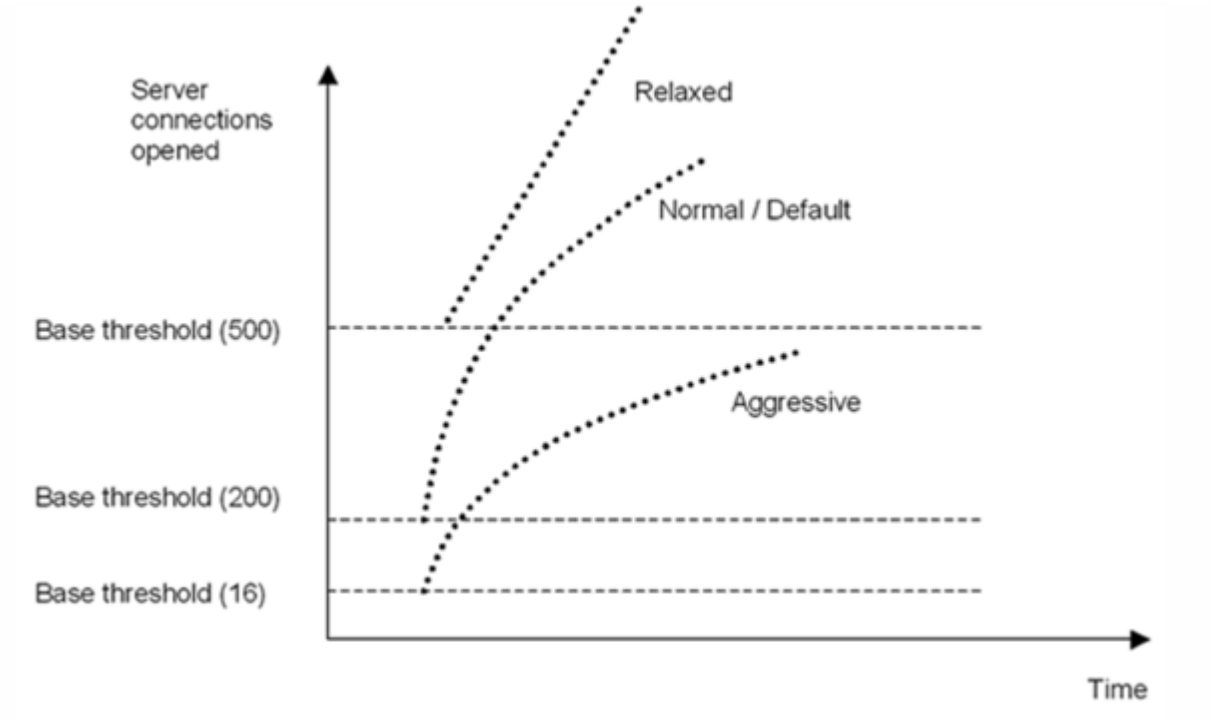
Set thresholds for surge protection

September 14, 2021

To set the rate at which the Citrix ADC appliance opens connections to the server, you must configure the threshold and throttle values for surge protection.

The following figure shows the surge protection curves that result from setting the throttle rate to relaxed, normal, or aggressive. Depending on the configuration of the server capacity, you can set base threshold values to generate appropriate surge protection curves.

Figure 1. Surge Protection Curves

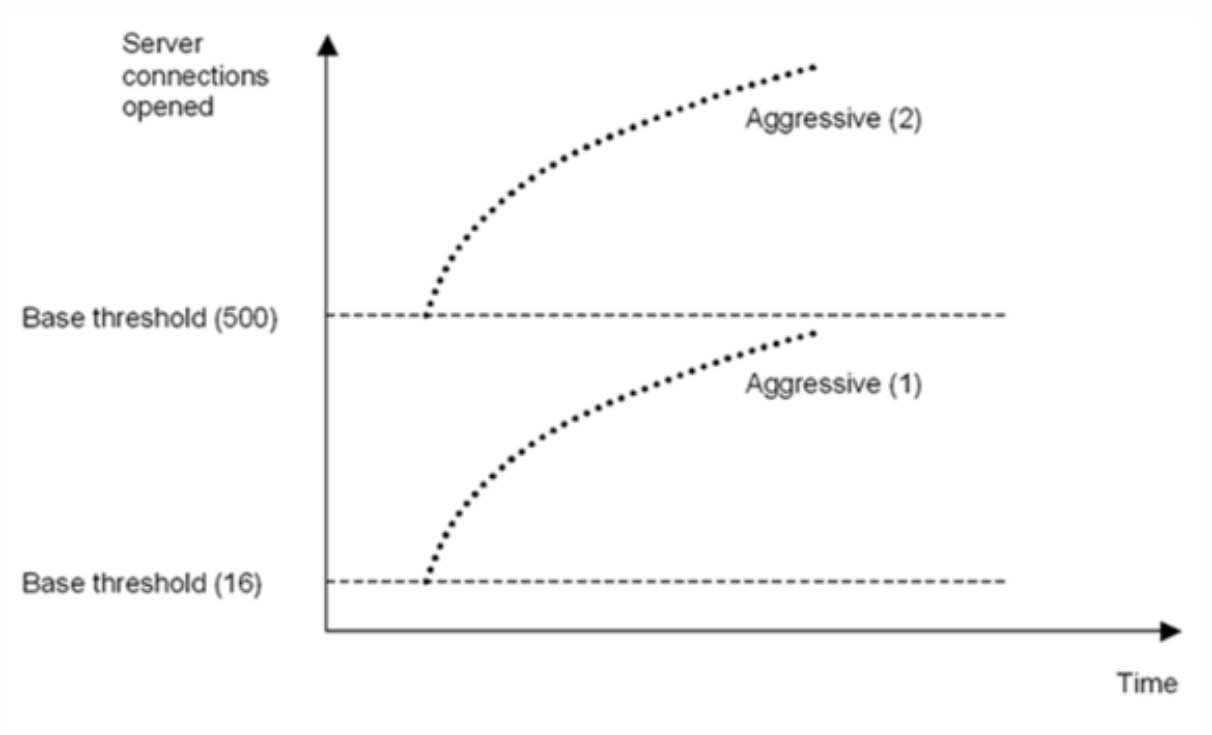


Your configuration settings affect the behavior of surge protection in the following manner:

- If you do not specify a throttle rate, it is set to normal (the default value), and the base threshold is set to 200, as shown in the preceding figure.
- If you specify a throttle rate (aggressive, normal, or relaxed) without specifying a base threshold, the curve reflects the default values of the base threshold for that throttle rate. For example, if you set the throttle rate to relaxed, the resulting curve will have the base threshold value of 500.
- If you specify only the base threshold, the entire surge protection curve shifts up or down, depending on the value you specify, as shown in the figure that follows.
- If you specify both a base threshold and a throttle rate, the resulting surge protection curve is based on the set throttle rate and adjusted according to the value set for the base threshold.

In the following figure, the lower curve (Aggressive 1) results when the throttle rate is set to aggressive but the base threshold is not set. The upper curve (Aggressive 2) results when the base threshold is set to 500, but the throttle rate is not set. The second upper curve (Aggressive 2) also results when the base threshold is set to 500, and the throttle rate is set to aggressive.

Figure 2. Aggressive Rate with the Default or a Set Base Threshold



Set the threshold for surge protection by using the GUI

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Global System Settings.
3. If you want to set a base threshold different from the default for the throttle rate, in the Configure Global Settings dialog box, Base Threshold text box, enter the maximum number of concurrent server connections allowed before surge protection is triggered. The base threshold is the maximum number of server connections that can be open before surge protection is activated. The maximum value for this setting is 32,767 server connections. The default setting for this value is controlled by the throttle rate you choose in the next step.

   Note: If you do not set an explicit value here, the default value will be used.
4. In the Throttle drop-down list, select a throttle rate. The throttle is the rate at which the Citrix ADC appliance allows connections to the server to be opened. The throttle can be set to the following values:

   - **Aggressive:** Choose this option when the connection-handling and surge-handling capacity of the server is low and the connection needs to be managed carefully. When you set the throttle to aggressive, the base threshold is set to a default value of 16, which means that surge protection is triggered whenever there are 17 or more concurrent connections to the server.

- **Normal:** Choose this option when there is no external load balancer behind the Citrix ADC appliance or downstream. The base threshold is set to a value of 200, which means that surge protection is triggered whenever there are 201 or more concurrent connections to the server. Normal is the default throttle option.
- **Relaxed:** Choose this option when the Citrix ADC appliance is performing load balancing between a large number of Web servers, and can therefore handle a high number of concurrent connections. The base threshold is set to a value of 500, which means that surge protection is triggered only when there are 501 or more concurrent connections to the server.

5. Click OK. A message appears in the status bar, stating that the global settings are configured.

Flush the surge queue

September 14, 2021

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the Citrix ADC appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between a client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups

bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed. The other requests in the surge queue of the load balancing virtual server are not flushed.

Note:

> - You cannot flush the surge queues of cache redirection, authentication, VPN, or GSLB virtual servers or GSLB services.
> - Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

Flush a surge queue by using the CLI

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while running the command, the surge queue of the specified entity is flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while running the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member <serverName> and <port> without specifying the name of the service group <name> and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you run the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
```

Examples

1. `flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80`

The preceding command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10

2. `flush ns surgeQ`

   The preceding command flushes all the surge queues on the appliance.

Flush a surge queue by using the GUI

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

DNS security options

September 14, 2021

You can now configure the DNS security options from the Add DNS Security Profile page in the Citrix ADC GUI. To configure the DNS security options from the Citrix ADC CLI or the NITRO API, use the AppExpert components. For instructions, see the NITRO API documentation and the Citrix ADC Command Reference Guide.

One option, cache poisoning protection, is enabled by default and cannot be disabled. You can apply the other options to all DNS endpoints or to specific DNS virtual servers in your deployment, as shown in the following table:

Security option	Can be applied to all DNS endpoints?	Can be applied to specific DNS virtual servers?
DNS DDoS protection	Yes	Yes
Manage exceptions – whitelist/blacklist servers	Yes	Yes
Prevent random subdomain attacks	Yes	Yes
Bypass the cache	Yes	No
Enforce DNS transactions over TCP	Yes	Yes
Provide root details in the DNS response	Yes	No

Cache poisoning protection

A cache poisoning attack redirects users from legitimate sites to malicious websites.

For example, the attacker replaces a genuine IP address in the DNS cache with a fake IP address that they control. When the server responds to requests from these IP addresses, the cache is poisoned. Subsequent requests for the addresses of the domain are redirected to the attacker's site.

The Cache Poisoning Protection option prevents insertion of corrupt data into the database that caches DNS server requests and responses. This feature is inbuilt in the Citrix ADC appliances and is always enabled.

DNS DDoS protection

You can configure the DNS DDoS Protection option for each type of request that you suspect might be used in a DDoS attack. For each type, the appliance drops any requests received after a threshold value for the number of requests received in a specified time period (time slice) is exceeded. You can also configure this option to log a warning to the SYSLOG server. For example:

- **DROP:** - Select this option to DROP requests without logging. Assume that you have enabled A record protection with threshold value 15, time slice as 1 second, and chosen DROP. When the incoming requests exceed 15 queries in 1 second, then the packets start getting dropped.
- **WARN:** - Select this option to LOG and DROP requests. Assume that you have enabled A record protection with threshold value 15, time slice as 1 second, and chosen WARN. When the incoming requests exceed 15 queries in 1 second, a warning message is logged indicating a threat and then the packets are dropped. Citrix recommends you to set threshold values for WARN smaller than the threshold value of DROP for a record type. Such a setting helps administrators identify an attack by logging a warning message before the actual attack happens and Citrix ADC starts dropping incoming requests.

Set a threshold for incoming traffic by using the GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profile** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
4. Expand **DNS DDoS Protection**.
    a) Select the record type and enter the threshold limit and the time slice value.
    b) Select **DROP** or **WARN**.
    c) Repeat steps a and b for each of the other record types that you want to protect against.
5. Click **Submit**.

Manage exceptions – allowlist/blocklist servers

Manage exceptions enables you to add exceptions either to block list or allow list domain name and IP addresses. For example:

- When a particular IP address is identified posting an attack, such IP address can be added to the block list.
- When administrators find that there is an unexpectedly high number of requests for a particular domain name, then that domain name can be added to the block list.
- `NXDomains` and some of the existent domains which can consume the server resources can be blacklisted.
- When administrators allow list domain names or IP addresses, queries or requests only from these domains or IP addresses are answered and all others are dropped.

Create an allow list or a block list by using the GUI

1. Navigate to **Configuration > Security > DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
   a) Expand **Manage exceptions – Whitelist/Blacklist Servers**.
   b) Select **Block** to block queries from blacklisted domains/addresses, or select **Allow** only to allow queries from whitelisted domains/addresses.
   c) In the **Domain name / IP Address** box, enter the domain names, IP addresses, or IP address ranges. Use commas to separate the entries.
   Note: If you select **Advanced Option**, you can use the “start with,” “contains,” and “ends with” options to set the criteria.
   For example, you can set criteria to block a DNS query that starts with “image” or ends with “.co.ru” or contains “mobile sites.”
4. Click **Submit**.

Prevent random subdomain attacks

In random subdomain attacks, queries are sent to random, nonexistent subdomains of legitimate domains. This action increases the load on the DNS resolvers and servers. As a result, they can become overloaded and slow down.

The Prevent Random Subdomain Attacks option directs the DNS responder to drop DNS queries that exceed a specified length.

Assume that example.com is a domain name owned by you and hence the resolution request comes to your DNS server. The attacker can append a random subdomain to example.com and send a request. Based on the specified query length and the FQDN, the random queries are dropped.

For example, if the query is www.image987trending.example.com, it is dropped if the query length is set to 20.

Specify a DNS query length by using the GUI

1. Navigate to **Configuration > Security > DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
   a) Expand **Prevent Random Subdomain Attacks**.
   b) Enter the numerical value for the query length.
4. Click **Submit**.

Bypassing the cache

During an attack, the data that is already cached must be protected. To protect the cache, new requests for certain domains or record types or response codes can be sent to the origin servers instead of cached.

The Bypassing the cache option directs the Citrix ADC appliance to bypass the cache for specified domains, record types, or response codes when an attack is detected.

Bypass the cache for specified domains or record types or response types by using the GUI

1. Navigate to **Configuration > Security > DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, expand **Bypassing the cache** and enter the domain names. Optionally, choose the record types or the response types for which the cache has to be bypassed.
   - Click **Domains** and enter the domain names. Use commas to separate the entries.
   - Click **Record Types** and choose the record types.
   - Click **Response Types** and choose the response type.
4. Click **Submit**.

Enforce DNS transactions over TCP

Some DNS attacks can be prevented if the transactions are forced to use TCP instead of UDP. For example, during a bot attack, the client sends a flood of queries but cannot handle responses. If the use of TCP is enforced for these transactions, then the bots cannot understand the responses and therefore cannot send requests over TCP.

Force domains or record types to operate at the TCP level by using the GUI

1. Navigate to **Configuration > Security > DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, expand **Enforce DNS Transactions over TCP** and enter the domain names and / or choose the record types for which the DNS transactions must be enforced over TCP.
   - Click **Domains** and enter the domain names. Use commas to separate the entries.
   - Click **Record Types** and choose the record types.
4. Click **Submit**.

Provide root details in the DNS response

In some attacks, the attacker sends a flood of queries for unrelated domains that are not configured or cached on the Citrix ADC appliance. If the `dnsRootReferral` parameter is ENABLED, it exposes all the root servers.

The Provide Root Details in the DNS Response option directs the Citrix ADC appliance to restrict access to root referrals for a query that is not configured or cached. The appliance sends a blank response.

The Provide Root Details in the DNS Response option can also mitigate or block amplification attacks. When the dnsRootReferral parameter is DISABLED, there are no root referrals in the Citrix ADC responses and hence they do not get amplified.

Enable or disable access to the root server by using the GUI

1. Navigate to **Configuration > Security > DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
   a) Expand **Provide Root Details in the DNS Response**.
   b) Click **ON** or **OFF** to allow or restrict access to the root server.
4. Click **Submit**.

System

September 14, 2021

This section provides system-level information of the Citrix ADC. This includes a detailed explanation of system-level features, the scenarios in which the features can be used, the configuration steps, and examples to help you better understand the features.

- Basic Operations
- Authentication and Authorization
- TCP Configurations
- HTTP Configurations
- SNMP
- Audit Logging
- Web Server Logging
- Call Home
- Reporting Tool
- CloudBridge Connector
- High Availability
- TCP Optimization

System base operations

September 14, 2021

The following configurations enable you to perform system base operations on a Citrix ADC appliance.

How to view, save, and clear Citrix ADC configuration

Citrix ADC configurations are stored in the `/nsconfig/ns.conf directory`. For configurations to be available across sessions, you must save the configuration after every configuration change.

View running configuration by using the command interface

At the command prompt, type:

```
show ns runningConfig
```

View running configuration by using the GUI

1. Navigate to **System** > **Diagnostics** and, in the **View Configuration group**, click **Running Configuration.**

View the difference between the two configuration files by using the command interface

At the command prompt, type:

```
diff ns config <configfile> <configfile2>
```


View the difference between the two configuration files by using the GUI

1. Navigate to **System** > **Diagnostics** and, in the **View Configuration group**, click **Configuration difference**.

Save Citrix ADC configurations by using the command interface

At the command prompt, type:

```
save ns config
```

Save Citrix ADC configurations by using the GUI

1. On the **Configuration** tab, in the top-right corner, click the **Save** icon.

View saved configurations by using the command interface

At the command prompt, type:

```
show ns ns.conf
```

View saved configurations by using the GUI

Navigate to **System** > **Diagnostics** and, in the **View Configuration** group, click **Saved Configuration**.

Clear Citrix ADC configuration by using the command interface

You have the following three options for clearing the Citrix ADC configuration.

Basic level. Clearing your configuration at the basic level clears all settings except the following:

- `Nsroot` password
- Time Zone
- NTP server
- ADM server connect
- License fie information
- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions
- Feature and mode settings
- Default administrator password (`nsroot`)

Extended level. Clearing your configuration at the extended level clears all settings except the following:

- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions

Feature and mode settings revert to their default values.

Full level. Clearing your configuration at the full level returns all settings to their factory default values. However, the NSIP and default gateway are not changed, because changing them can cause the appliance to lose network connectivity.

At the command prompt, type:

```
clear ns config -force
```

Example: To forcefully clear the basic configurations on an appliance.

```
clear ns config -force basic
```

Clear Citrix ADC configuration by using the GUI

Navigate to **System** > **Diagnostics** and, in the Maintenance group, click **Clear Configuration** and select the configuration level to be cleared from the appliance.

How to restart or shut down appliance for unsaved Citrix ADC configurations

The Citrix ADC appliance can be remotely restarted or shut down from the available user interfaces. When you restart or shut down a standalone Citrix ADC appliance, the unsaved configurations (configurations performed since the last
save ns config command was issued) are lost.

In a high availability setup, when the primary appliance is rebooted or shut down, the secondary appliance takes over and becomes the primary. The unsaved configurations from the old primary are available on the new primary appliance.

You can also restart the appliance by only rebooting the Citrix ADC software and not rebooting the underlying operating system. This is called a warm reboot. For example, when you add a new license or change the IP address, you can warm reboot the Citrix ADC appliance for these changes to take place.

> **Note:**
>
> You can perform warm reboot only on a standalone Citrix ADC appliance.

Restart the appliance by using the command interface

At the command prompt, type:


```
reboot [-warm]
```

Restart a Citrix ADC appliance by using the GUI

1. In the configuration page, click **Reboot**.
2. When prompted to reboot, select **Save configuration** to make sure that you do not lose any configurations.

> **Note:**
>
> You can perform a warm reboot by selecting Warm reboot.

Shut down an appliance by using the command interface

At the command prompt, type:

- `shutdown -p now`: Shuts down the software and switches off the Citrix ADC. To restart Citrix ADC MPX, press the AC power switch. To Restart Citrix ADC VPX, restart the VPX instance.
- `shutdown -h now`: Shuts down the software and leaves the Citrix ADC switched on. Press any key to restart the Citrix ADC. This command does not switch off the Citrix ADC. Therefore, do not switch off the AC power or remove the AC power cables.

> **Note:**
>
> You cannot shut down an appliance through the Citrix ADC GUI.

How to synchronize system clock with servers on the network

You can configure your Citrix ADC appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network.

You can configure clock synchronization on your appliance by adding NTP server entries to the ntp.conf file from either the GUI or the command line interface, or by manually modifying the ntp.conf file and then starting the NTP daemon (NTPD). The clock synchronization configuration does not change if the appliance is restarted, upgraded, or downgraded. However, the configuration does not get propagated to the secondary Citrix ADC in a high availability setup.

Citrix ADC GUI allows you to configure the time zone and the NTP server IP address required for clock synchronization on the first-time-user (FTU) screen.

> **Note:**
>
> If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site, <http://www.ntp.org>, under Public Time Servers List. Before configuring

> your Citrix ADC to use a public NTP server, be sure to read the Rules of Engagement page (link included on all Public Time Servers pages).
>
> In Citrix ADC release 11, the NTP version has been updated from 4.2.6p3 to 4.2.8p2.

Pre-requisite

To configure clock synchronization, you must configure the following entities:

1. NTP servers
2. NTP synchronization.

Add an NTP server by using the command interface

At the command prompt, type the following commands to add an NTP server and verify the configuration:

- `add ntp server (<serverIP> | <serverName>)[-minpoll <positive_integer>] [-maxpoll <positive_integer>]`
- `show ntp server`

Example:

```
add ntp server 10.102.29.30 -minpoll 6 -maxpoll 11
```

Add an NTP server by using the GUI

Navigate to **System** > **NTP Servers**, and create the NTP server.

Enable NTP synchronization by using the command interface

When you enable NTP synchronization, the Citrix ADC starts the NTP daemon and uses the NTP server entries in the ntp.conf file to synchronize its local time setting. If you do not want to synchronize the appliance time with the other servers in the network, you can disable NTP synchronization, which stops the NTP daemon (NTPD).

At the command prompt, type one of the following commands:

```
enable ntp sync
```

Enable NTP synchronization by using the GUI

Navigate to **System** > **NTP Servers**, click **Action** and select **NTP Synchronization**.

Configure clock synchronization to edit a ntp.conf file by using the GUI

1. Log on to the command line interface.
2. Switch to the shell prompt.
3. Copy the /etc/ntp.conf file to /nsconfig/ntp.conf, unless the /nsconfig directory already contains an ntp.conf file.
4. For each NTP server you want to add, you must add the following two lines to the /nsconfig/ntp.conf file:

```
server <IP address for NTP server> iburst

restrict <IP address for NTP server> mask <netmask> nomodify notrap nopeer
noquery
```

```
1  > Note:
2  >
3  > For security reasons, there should be a corresponding restrict entry
       for each server entry.
4
5  Example
6
7  In the following example, an administrator has inserted # characters to
       “comment out” an existing NTP entry, and then added an entry:
8
9  `#server 1.2.3.4 iburst`
10
11 `#restrict 1.2.3.4 mask 55.255.255.255 nomodify notrap nopeer noquery`
12
13 `server 10.102.29.160 iburst`
14
15 `restrict 10.102.29.160 mask 255.255.255.255 nomodify notrap nopeer
       noquery`
```

1. If the /nsconfig directory does not contain a file named rc.netscaler, create the file.
2. Add the following entry to /nsconfig/rc.netscaler: /bin/sh /etc/ntpd_ctl full_start

   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   This process runs every time the Citrix ADC is restarted.
3. Restart the Citrix ADC appliance to enable clock synchronization. Or, to start the time synchronization process without restarting the appliance, enter the following commands at the shell prompt:

- `rm /etc/ntp.conf`
- `ln -s /nsconfig/ntp.conf /etc/ntp.conf`
- `/bin/sh /etc/ntpd_ctl full_start`

How to configure session timeout for idle client connections

A session timeout interval is provided to restrict the time duration for which a session (GUI, CLI, or API) remains active when not in use. For the Citrix ADC, the system session timeout can be configured at the following levels:

- **User level timeout.** Applicable to the specific user.

Interface type	Time out configuration
GUI	Navigate to **System > User Administration > Users**, select a user, and edit the user's timeout setting.
CLI	At the command prompt, enter the following command: `set system user <name> -timeout <secs>`

- **User group level timeout.** Applicable to all users in the group.

Interface type	Time out configuration
GUI	Navigate to **System > User Administration > Groups**, select a group, and edit the group's timeout setting.
CLI	At the command prompt, enter the following command:`set system group <groupName> -timeout <secs>`

- **Global system timeout.** Applicable to all users and users from groups who do not have a time-out configured.

Interface type	Time out configuration
GUI	Navigate to **System > Settings**, click Change global system settings, and update the timeout value as required.

Interface type	Time out configuration
CLI	At the command prompt, enter the following command: `set system parameter -timeout <secs>`

```
The timeout value specified for a user has the highest priority. If timeout is not configured for the user, the timeout configured for a member group is considered. If timeout is not specified for a group (or the user does not belong to a group), the globally configured timeout value is considered. If timeout is not configured at any level, the default value of 900 seconds is set as the system session timeout.

Additionally, you can specify timeout durations for each of the interfaces you are accessing. However, the timeout value specified for a specific interface is restricted to the timeout value configured for the user that is accessing the interface. For example, let us consider an user "publicadmin" who has a timeout value of 20 minutes. Now, when accessing an interface, the user must specify a timeout value that is within 20 minutes.

> **Note:**
>
> You can choose to keep a check on the minimum and maximum timeout values by specifying the timeout as restricted (in CLI by specifying the *restrictedTimeout* parameter). This parameter is provided to account for previous Citrix ADC versions where the timeout value was not restricted.
```

- When enabled, the minimum configurable timeout value is 5 minutes (300 secs) and the maximum value is 1 day (86400 secs). If the timeout value is already configured to a value larger than 1 day, when this parameter is enabled, you are prompted to change it. If you do not change the value, the timeout value will automatically be reconfigured to the default timeout duration of 15 minutes (900 secs) on the next reboot. The same will happen is the configured timeout value is less than 5 minutes.
- When disabled, the configured timeout durations are considered.
- **Timeout duration at each interface:**

Interface type	Time out configuration
CLI	Specify the timeout value on the command prompt by using the following command: `set cli mode -timeout <secs>`
API	Specify the timeout value in the login payload.

How to set system date and time to synchronize clock with a time server

To change the system date and time, you must use the shell interface to the underlying FreeBSD OS. However, to view the system date and time, you can use the command line interface or the GUI.

View system date and time by using the command interface

At the command prompt, type:

```
show ns config
```

View system date and time by using the GUI

Navigate to **System** and select the **System Information** tab to view the system date.

How to configure HTTP and HTTPS management ports for internal services

In a single-IP mode deployment of a Citrix ADC appliance, a single IP address is used as NSIP, SNIP, and VIP addresses. This single IP address uses different port numbers to function as NSIP, SNIP, and VIP addresses.

Port numbers 80 and 443 are well-known ports for HTTP and HTTPS services. Earlier, port 80 and 443 of the Citrix ADC IP address (NSIP) were dedicated ports for internal HTTP and HTTPS management services. Because these ports were reserved for internal services, you cannot use these well-known ports for providing HTTP and HTTPS data services from a VIP address, which has the same address as the NSIP address in a single-IP mode deployment.

To address this requirement, you can now configure ports for internal HTTP and HTTPS management services (of the NSIP address) other than port 80 and 443.

The following lists the default port numbers for internal HTTP and HTTPS management services in Citrix ADC MPX, VPX, and CPX appliances:

- Citrix ADC MPX and VPX appliances: 80 (HTTP) and 443 (HTTPS)
- Citrix ADC CPX appliances: 9080 (HTTP) and 9443 (HTTPS)

Configure HTTP and HTTPS management ports by using the command interface

You can configure an HTTP and an HTTPS port to any value on the Citrix ADC appliance to support the HTTP and HTTPS management service. However, by default, the Citrix ADC appliance uses 80 and 443 ports for HTTP and HTTPS connection.

At the command prompt, type:

```
set ns param -mgmtHttpPort<port>
```

Example:

```
set ns param -mgmtHttpPort 2000
```

To configure an HTTPS port by using the command interface

At the command prompt, type:

```
set ns param -mgmtHttpsPort<port>
```

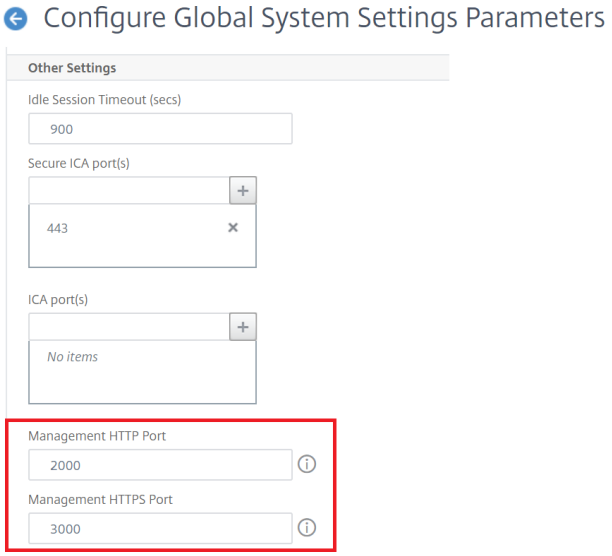
Example:

```
set ns param -mgmtHttpsPort 3000
```

Configure HTTP and HTTPS management ports by using the GUI

Follow the steps given below to configure HTTP and HTTPS port values:

1. Navigate to **System** > **settings** > **Change global system settings**.
2. In **Configure Global System Settings Parameters** page, under **Other Setting** section, set the following parameters.
   a) Management HTTP Port. set port value to 2000. Default = 80, Min = 1, Max = 65534.
   b) Management HTTPS Port. set port value to 3000. Default = 443, Min = 1, Max = 65534.



How to allocate extra management CPU for data processing and monitoring

If you need better performance for configuration and monitoring of a Citrix ADC MPX appliance, you can allocate an extra management CPU from the appliance's packet engine pool. This feature is supported on certain Citrix ADC MPX models and all VPX models except the VPX instances that run on Citrix ADC SDX appliances. It affects the output of the stat system CPU and stat system commands.

Supported Citrix ADC MPX models:

- 25xxx
- 22xxx
- 14xxx
- 115xx
- 15xxx
- 26xxx

> **Note:**
>
> For Citrix ADC MPX 26xxx models with more than 20 cores, the mandatory extra management CPU feature is enabled by default. For Citrix ADC VPX models, a license that supports at least 12 vCPUs is required to enable this feature.

Allocate an extra management CPU by using the command interface

At the command prompt, type one of the following commands:

- `enable extramgmtcpu`
- `disable extramgmtcpu`

> **Note:**
>
> After you enable and disable this feature, the Citrix ADC appliance displays a warning to restart the appliance, for the changes to take effect.

To show the configured and effective state of an extra management CPU.

At the command prompt, type:

```
1 `show extramgmtcpu`
```

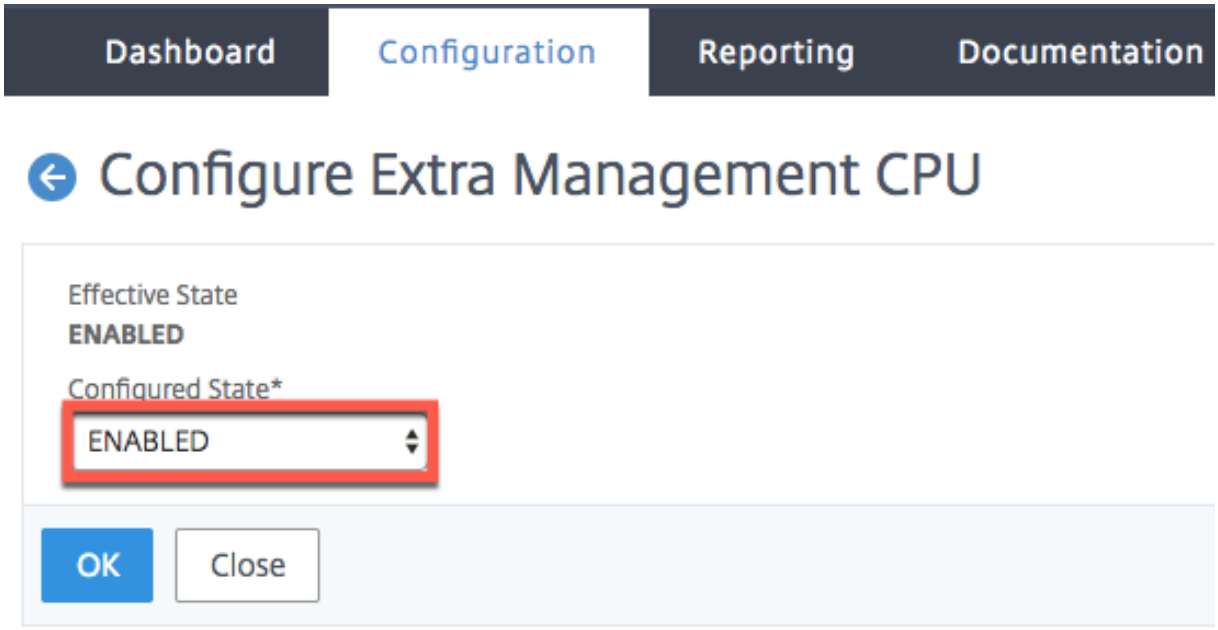
Example:

```
> show extramgmtcpu ConfiguredState: ENABLED EffectiveState: ENABLED
```

> **Note:**
>
> In this example, the show command is entered before restarting the appliance.

Allocate an extra management CPU by using the GUI

To allocate an extra management CPU by using the GUI, navigate to **System** > **Settings** and click **Configure Extra Management CPU**. From the **Configured State** drop-down menu, select **Enabled** and then select **OK**.



To check CPU usage, go to **System** > **Settings** > **Dashboard**.

Configure an extra management CPU by using the NITRO API

Use the following NITRO methods and formats to enable, disable, and show an extra management CPU.

To enable an extra management CPU:

```
HTTP Method: POST
```

```
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=enable
```

```
Payload: { "systemextramgmtcpu":{ } }
```

```
curl -v -X POST -H "Content-Type: application/json"-u nsroot:nsroot http://10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=enable -d '{ "systemextramgmtcpu":{ } } '
```

To disable an extra management CPU

```
HTTP Method: POST
```

```
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=disable
```

```
Payload: { "systemextramgmtcpu":{ } }
```



```
curl -v -X POST -H "Content-Type: application/json"-u nsroot:nsroot http://10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=disable -d '{ "systemextramgmtcpu":{ } } '
```

To show an extra management CPU

```
HTTP Method: GET
```

```
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu
```

Example:

```
curl -v -X GET -H "Content-Type: application/json"-u nsroot:nsroot http://10.102.201.92/nitro/v1/config/systemextramgmtcpu
```

Statistics and monitoring before and after adding extra management CPU

The following examples show the differences in the output of the stat system CPU and stat system commands before and after adding an extra management CPU.

```
stat system cpu
```

This command displays statistics of CPUs.

Here is a sample output before adding an extra management CPU on one of the supported models.

Example

```
```
> stat system cpu

CPU statistics

ID   Usage

8   1

7   1

11   2

1   1

6   1

9   1

3   1
```

 
```

```
21
22 5               1
23
24 4               1
25
26 10              1
27
28 2               1
29 <!--NeedCopy--> ```
```

Here is the output after adding an extra management CPU on the same MPX appliance.

```
1  ```
2  > stat system cpu
3
4  CPU statistics
5
6  ID          Usage
7
8  9               1
9
10 7               1
11
12 5               1
13
14 8               1
15
16 11              2
17
18 10              1
19
20 6               1
21
22 4               1
23
24 3               1
25
26 2               1
27 <!--NeedCopy--> ```
```

`stat system`

This command displays CPU use. In the following example, the output before adding an extra management CPU on one of the supported models is:

Mgmt Additional-CPU usage (%) 0.00

Example

```
```
> stat system

Citrix ADC Executive View

System Information:

Up since   Wed Oct 11 11:17:54 2017

/flash Used (%)   0

Packet CPU usage (%)   1.30

Management CPU usage (%)   4.00

Mgmt CPU0 usage (%)   4.00

Mgmt Additional-CPU usage (%)   0.00

Memory usage (MB)   2167

InUse Memory (%)   5.76

/var Used (%)   0
<!--NeedCopy--> ```
```

In the following example, the output after adding an extra management CPU on the same MPX appliance is:

Mgmt Additional-CPU usage (%) 0.80

```
```    > stat system

Citrix ADC Executive View

System Information:

Up since        Wed Oct 11 11:55:56 2017

/flash Used (%)                          0

Packet CPU usage (%)                 1.20

Management CPU usage (%)             5.70
```



```
14
15 Mgmt CPU0 usage (%)                10.60
16
17 Mgmt Additional-CPU  usage (%)      0.80
18
19 Memory usage (MB)                   1970
20
21 InUse Memory (%)                    5.75
22
23 /var Used (%)                          0
24
25 <!--NeedCopy--> ```
```

How to backup and restore your appliance to recover lost configuration

When your appliance gets corrupted or needs an upgrade, you can back up your system configuration. The backup procedure is done either through the Citrix CLI or GUI interface. The appliance also enables you to import the backup file from an external source. However, you can do this only through the GUI interface and there is no support through the CLI interface.

Points to remember

You must remember the following points when your backup and restore your appliance.

- There must be a support for network configuration on a new platform.
- The new platform build must be same as the backup file or a later version.

Back up a Citrix ADC appliance

Depending on data and backup requirement, you can create a “basic” backup or a “full” backup.

- **Basic backup.** You can perform this type of backup if you want to back up files that constantly change. The files that you can back up are in the following table.

For information about the basic backup details, see the Table topic.

- **Full backup.** In addition to the files that are backed up by a basic backup, a full backup has less frequently updated files. The files that are backed up when you use the “Full” backup option are:

Directory	Sub-Directory or Files
nsconfig	ssl*, license*, fips*

Directory	Sub-Directory or Files
/var/	netscaler/ssl/*, wi/java_home/jre/lib/security/cacerts/*, wi/java_home/lib/security/cacerts/*

The backed-up data is stored as a compressed TAR file in the `/var/ns_sys_backup/` directory. To avoid issues due to non-availability of disk space, you can store up to 50 backup files in this directory. You can use the `rm system backup` command to delete existing backup files and create more backups.

> **Note:**
>
> When the backup operation is in progress, do not run commands that affect the configuration.
>
> If a file that is required to be backed up is not available, the operation skips that file.

Back up a Citrix ADC appliance using command interface

Follow the procedure given below to back up a Citrix ADC appliance by using the Citrix ADC command interface.

At the command prompt, do the following:

1. Save the Citrix ADC configurations.

```
save ns config
```

1. Create the backup file.

```
create system backup [<fileName>] -level <basic ¦ full> -comment <string>
```

> **Note:**
>
> If the file name is not specified, the appliance creates a TAR file with the following naming convention:`backup_<level>_<nsip_address>_<date-timestamp>.tgz`.

Example: To back up the full appliance using the default naming convention for the backup file.

```
> create system backup -level full
```

1. Verify that the backup file was created.

```
show system backup
```

You can view the properties of a specific backup file by using the `fileName` parameter.

Restore a Citrix ADC appliance by using the command interface

> **Important:**
>
> You cannot successfully restore your appliance, if you rename or modify your backup file.

When you restore your appliance, the restore operation untars the backup file from the `/var/ns_sys_backup/` directory. Once the files are untarred, the files are copied to the respective directories.

Restore the Citrix ADC from a local backup file by using the command interface

> **Note:**
>
> Citrix recommends you to back up the current configuration before restoring a previous configuration. However, if you do not want the restore command to automatically create a backup of the current configuration, use the `-skipBackup` parameter.

At the command prompt, do the following:

1. Obtain a list of the backup files available on the appliance.

   ```
   show system backup
   ```

2. Restore the appliance by specifying one of the backup files.

   ```
   restore system backup <filename> [-skipBackup]
   ```

   Example: To restore by using a full backup of an appliance

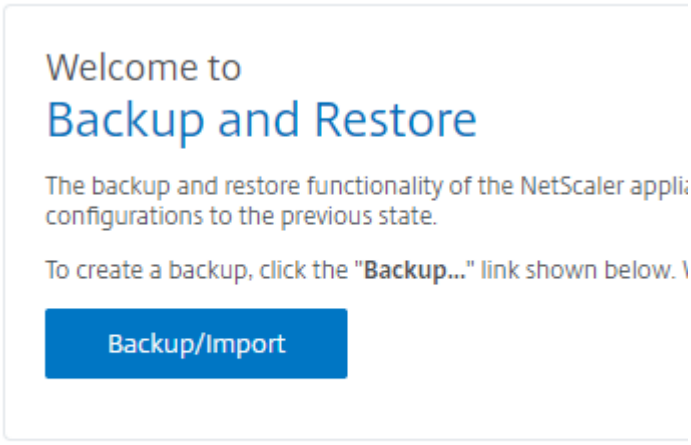
   ```
   > restore system backup backup_full_<nsip_address>_<date-timestamp>.tgz
   ```

3. Reboot the appliance.

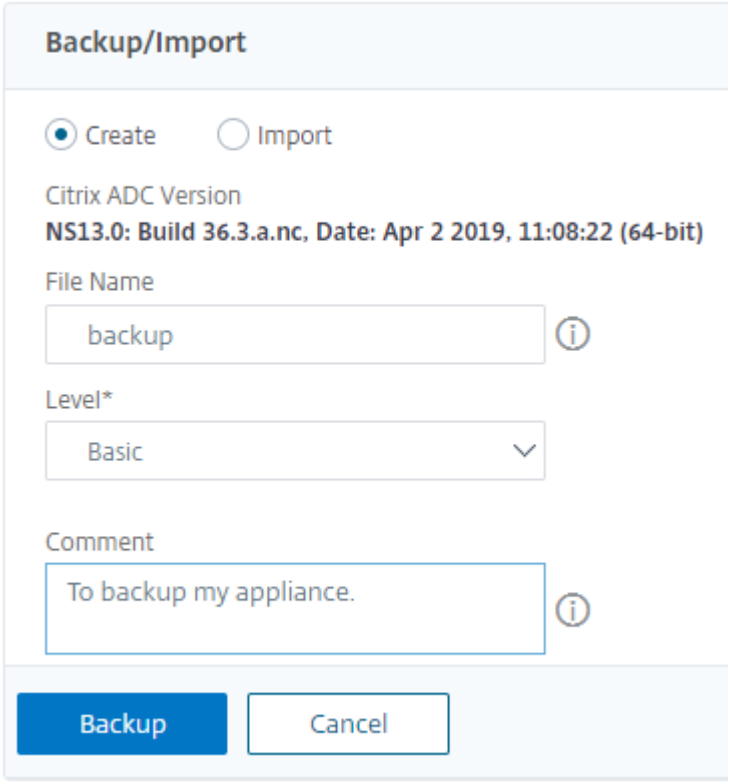
   ```
   reboot
   ```

Backup and restore a Citrix ADC appliance by using the GUI

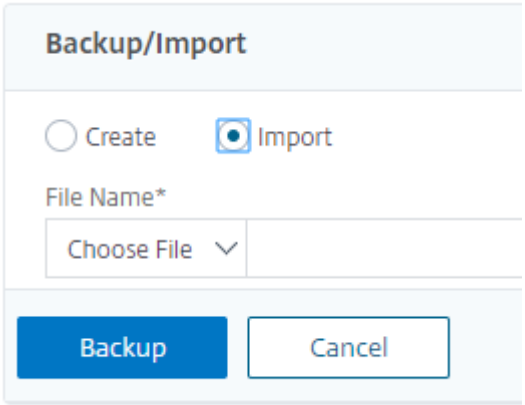
1. Navigate to **System** > **Backup and Restore**.



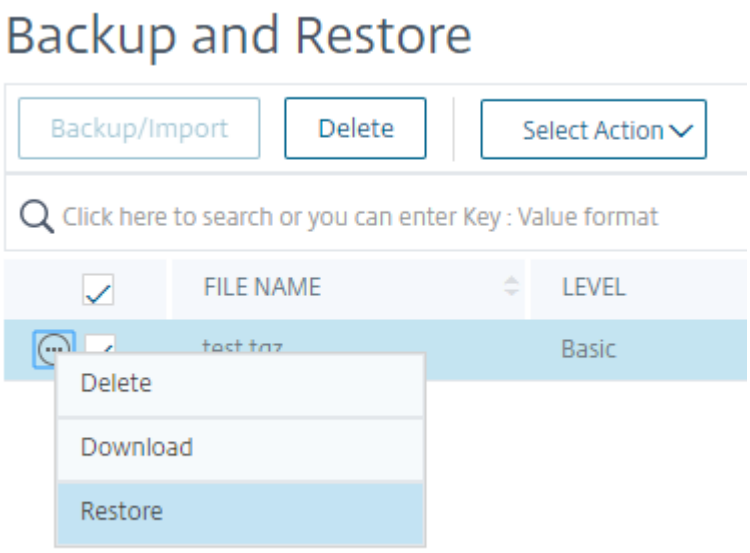
2. Click **Backup/Import** to start the process.
3. In the **Backup/Import** page, select **Create** and set the following parameters.
    a) File Name. Name of the appliance backup file.
    b) Level. Select a backup level as basic or full.
    c) Comment. Provide a brief description for the backup.
4. Click **Backup**.



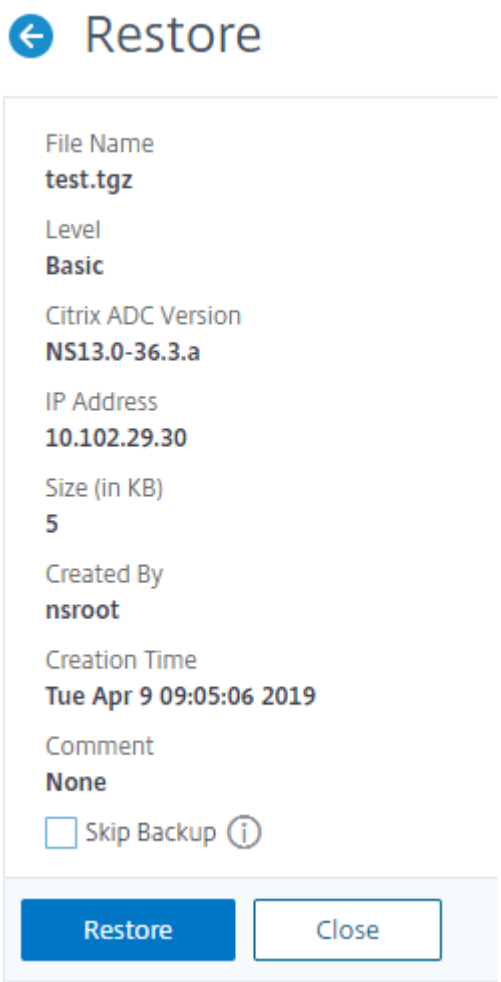
5. If you want to import a backup, you must select **Import**.



6. Once backup is complete, you can select the file and click **Download**.
7. To restore, select the backup file and click **Restore**.



8. In the **Restore** page, verify backup file details and click **Restore**.



9. After you restore, you must reboot the appliance.

For more information on how to backup and restore Citrix ADC instances, see Backup and Restore using Citrix ADM topic.

For more information on how to backup and restore an SDX appliance, see Backup and restore SDX appliance

For information about operations performed on system backup, see System Backup topic.

How to generate technical support bundle for resolving appliance issues

For help with analyzing and resolving any issues with a Citrix ADC appliance, you can generate a technical support bundle on the appliance and send the bundle to Citrix Technical Support. The Citrix ADC technical support bundle is a zipped tar archive of system configuration data and statistics. It collects the following data from the Citrix ADC appliance on which you generate the bundle:

- **Configuration files**. All files in the /flash/nsconfig directory.
- **Newnslog files**. The currently running newnslog and some previous files. To minimize the archive file size, `newnslog` collection is restricted to 500 MB, 6 files, or 7 days, whichever occurs first. If older data is needed, it might require manual collection.
- **Log files**. Files in /var/log/messages, */var/log/ns.log*, and other files under /var/log and /var/nslog.
- **Application core files**. Files created in the /var/core directory within the last week, if any.
- **Output of some CLI show commands**.
- **Output of some CLI stat commands**.
- **Output of BSD shell commands**.

You can use a single command to generate the technical support bundle and securely upload it to the Citrix Technical Support server. To upload, you must specify your Citrix credentials. When you generate the bundle, you can specify the case or service request number that was allotted to you by Citrix Technical Support. If you have already generated a technical support bundle, you can upload the existing archive file to the Citrix Technical Support server by specifying the file name with the full path.

The technical support bundle is saved on the Citrix ADC appliance in an archive at the following location:

`/var/tmp/support/support.tgz`

The path is a symlink to the most recent collector for easy access. The full file name varies, depending on the deployment topology, but generally follows a format similar to:

`collector_<P/S>_<NS IP>_<DateTime>.tgz.`

If your Citrix ADC appliance does not have direct internet connectivity, you can use a proxy server to directly upload the technical support bundle to the Citrix technical support server. The basic format of the proxy string is:

```
proxy_IP:<proxy_port>
```

If the proxy server requires authentication, the format is:

```
username:password@proxsy_IP:<proxy_port>
```

> **Note:**
>
> For Citrix ADC appliances in a high availability pair, you must generate the technical support bundle on each of the two nodes.
>
> For Citrix ADC appliances in a cluster setup, you can generate the technical support bundle on each node individually, or you can generate smaller abbreviated archives for all nodes by using the cluster IP address.
>
> For Citrix ADC admin partitions, you must generate the technical support bundle from the default admin partition. To get the technical support bundle for a specific partition, you must specify the name of the partition for which you want to generate the technical support bundle. If you do not specify the name of the partition, data is collected from all admin partitions.

Generate the Citrix ADC technical support bundle by using the command interface

At the command prompt, type:

```
show techsupport [-scope <scope> <partitionName>] [-upload [-proxy <string
>] [-casenumber <string>] [-file <string>] [-description <string>] [-
userName <string> -password ]]
```

Sr. No	Task	Command
1	Generate and upload the technical support bundle to the Citrix Technical Support server.	show techsupport –upload –userName account1 –password xxxxxxx
2	Generate and upload the technical support bundle to the Citrix Technical Support server through a proxy server	show techsupport –upload –proxy 1.1.1.1:80 –userName account1 –password xxxxxxx

Sr. No	Task	Command
3	Upload an existing technical support bundle to the Citrix Technical Support server.	show techsupport –upload -file,/var/tmp/support/collector_P_10.102.29. –userName account1 –password xxxxxxx
4	Generate small, abbreviated archives for all nodes in a Cluster setup. Run this command by using the cluster IP address	show techsupport –scope CLUSTER
5	Generate a technical support bundle specific to an admin partition. Run this command on the default admin partition.	show techsupport –scope PARTITION partition1

How to collect the technical support bundle from SDX and VPX appliances for insight analysis

A Citrix ADC appliance has a built-in mechanism to collect log files. The log files are in turn sent to Citrix Insight Services for analysis.

> **Note:**
>
> All procedures are applicable for software release 9.2 or later.

Download technical support bundle from Citrix ADC MPX and VPX appliances

To run a collector file by using the Citrix ADC GUI, you must complete the following procedure:

> **Note:**
>
> The procedure is applicable for software release 9.2 or later.

1. Navigate to **System** > **Diagnostic**.
2. In the **Technical Support Tools** section, click **Generate Support File** link.
3. In the **Tech Support** page, set the following parameters:
    a) Scope. To collect data from one or more nodes.
    b) Partition. Name of the partition.

c) Citrix Technical Support load Options. Set all options such as proxy server, service case number, collector archive file name and a brief description of the archive file for uploading the technical support bundle.

d) Citrix Account. Enter your Citrix credentials.

4. Click **Run**.
5. The Technical Support bundle is generated.
6. Click **Yes** to download the Technical support bundle to your local desktop.

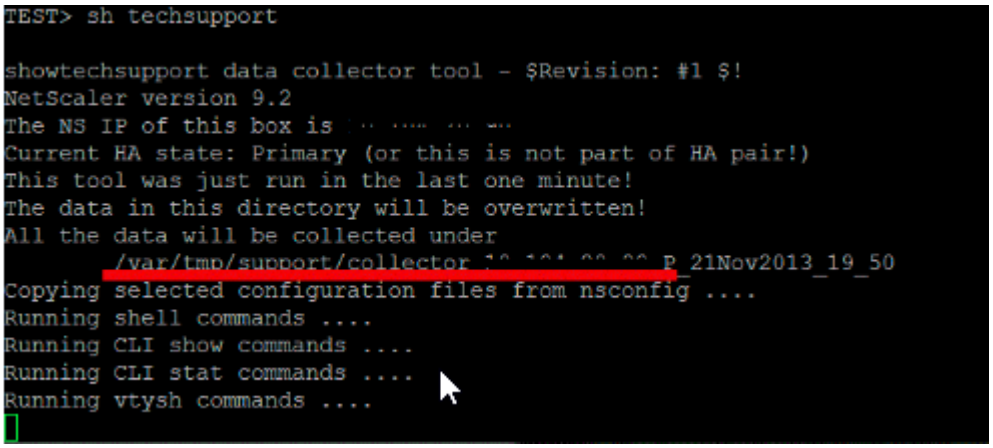
Obtain technical support bundle by using the command interface

1. Download the file from the appliance using a Secure FTP (SFTP) or Secure Copy (SCP) utility, such as WinSCP, and upload it to Citrix Insight Services for analysis.

> **Note:**
>
> In the Citrix ADC software release earlier than 9.0, the collector script must be downloaded separately and ran.

```
> show techsupport -scope CLUSTER
```

1. This collects show technical support information from all nodes in the cluster and compress the files into a single archive.
2. After the appliance generates the collector archive the location of the file is displayed as shown in the following screenshot.



The file is stored in /var/tmp/support and you can verify it by logging into a Citrix ADC appliance and running the following command from a shell prompt.

```
root@NS## cd /var/tmp/support/
root@NS## ls -l
```


Obtain diagnostic bundle from Citrix ADC SDX using the GUI

1. Open the Citrix SDX GUI.
2. Expand the **Diagnostics** node.
3. Select the **Technical Support** node.
4. Click Generate Technical Support File.
5. Select **Appliance** (Including Instances) from the drop-down menu.
6. Click **Add**.
7. Select one or more instances to add in.
8. Click **OK**. Wait for the process to complete.
9. Select the bundle name that was generated and then click **Download**
10. Upload the bundle file to Citrix Insight Services.

More Resources

Watch a Video

Read another topic

Command Reference Doc

System user authentication and authorization

September 14, 2021

To configure Citrix ADC user authentication and authorization, you must first define the users who have access to the Citrix ADC appliance, and then you can organize these users into groups. After configuring users and groups, you need to configure command policies to define types of access, and assign the policies to users and/or groups.

You must log on as an administrator to configure users, groups, and command policies. The default Citrix ADC administrator user name is *nsroot*. After logging on as the default administrator, you should change the password for the nsroot account. Once you have changed the password, no user can access the Citrix ADC appliance until you create an account for that user. If you forget the administrator password after changing it from the default, you can reset it to nsroot.

> **Note:**
>
> - Local users can authenticate to the Citrix ADC even if external authentication servers are configured. You can restrict this by disabling the localAuth parameter of the set system parameter command.
> - For enhanced security, Citrix recommends that you change the nsroot password. Fre-

> quently changing the password is advisable. For information about how to change the nsroot password, see Resetting the default administrator (nsroot) password topic.

User, user groups, and command policies

September 14, 2021

You must first define a user with an account and then organize all users into groups. You can create command policies, or use built-in command policies to regulate user access to commands.

> **Note:**
>
> If you prefer to know more about configuring user and user groups as part of Citrix ADC authentication and authorization setup for traffic management, see Configure users and groups topic.

You can also customize the command-line prompt for a user. Prompts can be defined in a user's configuration, in a user-group configuration, and in the global system configuration settings. The prompt displayed for a user is in the following order of precedence:

1. Display the prompt as defined in the user's configuration.
2. Display the prompt as defined in the group configuration for the user's group.
3. Display the prompt as defined in the system global configuration settings.

You can now specify a timeout value for inactive CLI sessions for a system user. If a user's CLI session is idle for a time that exceeds the timeout value, the Citrix ADC appliance terminates the connection. The timeout can be defined in a user configuration, in a user-group configuration, or in the global system configuration settings. The timeout for inactive CLI sessions for a user is determined in the following order of precedence:

1. User configuration.
2. Group configuration for the user's group.
3. Global system configuration settings.

A Citrix ADC root administrator can configure the maximum concurrent session limit for system users. By restricting the limit, you can reduce the number of open connections and improve server performance. As long as the CLI count is within the configured limit, concurrent users can log on the GUI any number of times. However, if the number of CLI sessions reaches the configured limit, users can no longer log on to the GUI. For example, if the number of concurrent sessions is configured to 20, concurrent users can log on to 19 CLI sessions. But if the user is logged on to the `20<sup>th</sup` > CLI session, any attempt to log on to the GUI, CLI, or NITRO results in an error message ((ERROR: Connection limit to CFE exceeded).

> **Note:**
>
> The default the number of concurrent sessions is configured to 20 and the maximum number of concurrent sessions is configured to 40.

Configure user accounts

To configure user accounts, you simply specify user names and passwords. You can change passwords and remove user accounts at any time.

> **Note:**
>
> All characters in a password are not accepted. However, it works if you type the characters within quotes.
>
> Also, the string must not exceed a maximum length of 127 characters.

To create a user account by using the command line interface

At the command prompt, type the following commands to create a user account and verify the configuration:

- `add system user <username> [-externalAuth ( ENABLED | DISABLED )] [-promptString <string>] [-timeout \<secs>] [-logging ( ENABLED | DISABLED )] [-maxsession <positive_integer>]`
- `show system user <userName>`

External users can configure the “logging” parameter to collect external logs using web logging or audit logging mechanism. If the parameter is enabled, the auditing client authenticates itself with Citrix ADC appliance to collect logs.

Example:

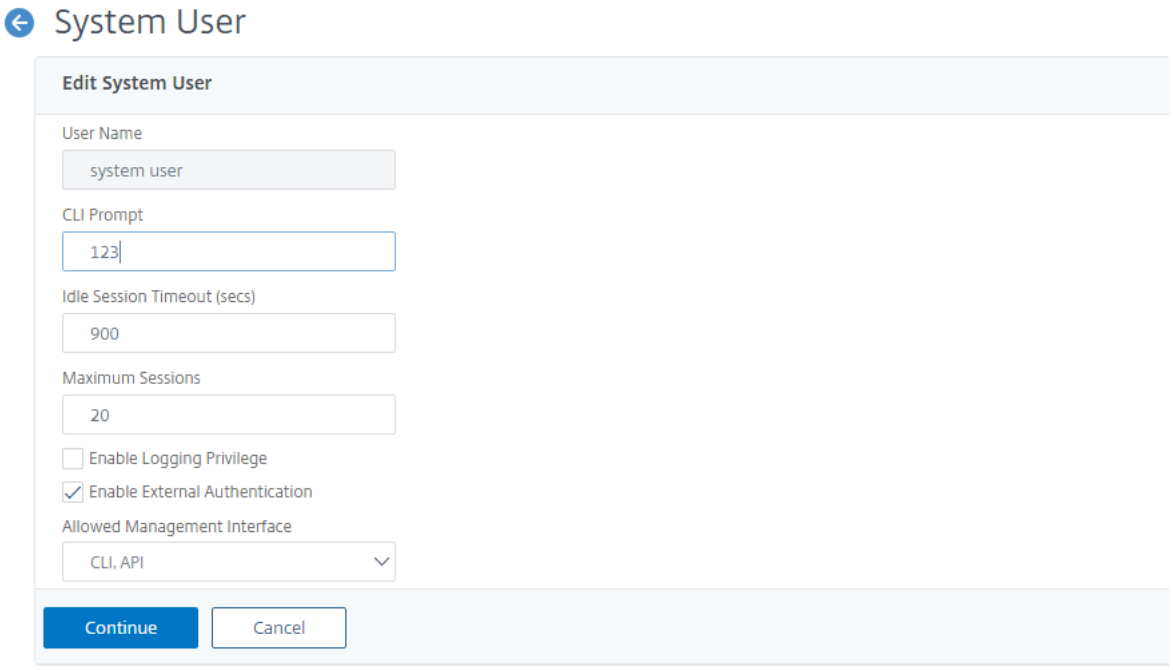
```
> add system user johnd -promptString user-%u-at-%T
```

```
Enter password:
Confirm password:
> show system user johnd
user name: john
     Timeout:900 Timeout Inherited From: Global
     External Authentication: ENABLED
     Logging: DISABLED
     Maximum Client Sessions: 20
<!--NeedCopy-->
```

For parameter description, see Authentication and authorization user command reference topic.

Configure a user account by using the Citrix ADC GUI

1. Navigate to **System** > **User Administration** > **Users**, and create the user.
2. In the details pane, click **Add** to create a system user.
3. In the **Create System Group** page, set the following parameters:
   a) User Name. Name of the user group.
   b) CLI Prompt. The prompt that you prefer to set for the CLI interface access.
   c) Idle Session Timeout (secs). Set the amount of time a user can be inactive before the session times out and closes.
   d) Maximum sessions. Set the maximum of sessions a user can try.
   e) Enable Logging Privilege. Enable logging privilege for the user.
   f) Enable external Authentication. Select the option if you want to user external authentication server for authenticating the user.
   g) Allowed Management Interface. Select the Citrix ADC interfaces for which the user group is granted permission to access.
   h) Command Policies. Bind command policies to the user group.
   i) Partitions. Bind partitions to the user group.
4. Click **Create** and **Close**.



Configure user groups

After configuring a user group, you can easily grant the same access rights to everyone in the group. To configure a group, you create the group and bind users to the group. You can bind each user account

to more than one group. Binding user accounts to multiple groups might allow more flexibility when applying command policies.

To create a user group by using the command line interface

At the command prompt, type the following commands to create a user group and verify the configuration:

- `add system group <groupName> [-promptString <string>] [-timeout <secs>]`
- `show system group <groupName>`

Example:

```
> add system group Managers -promptString Group-Managers-at-%h
```

Bind a user account to a group by using the CLI

At the command prompt, type the following commands to bind a user account to a group and verify the configuration:

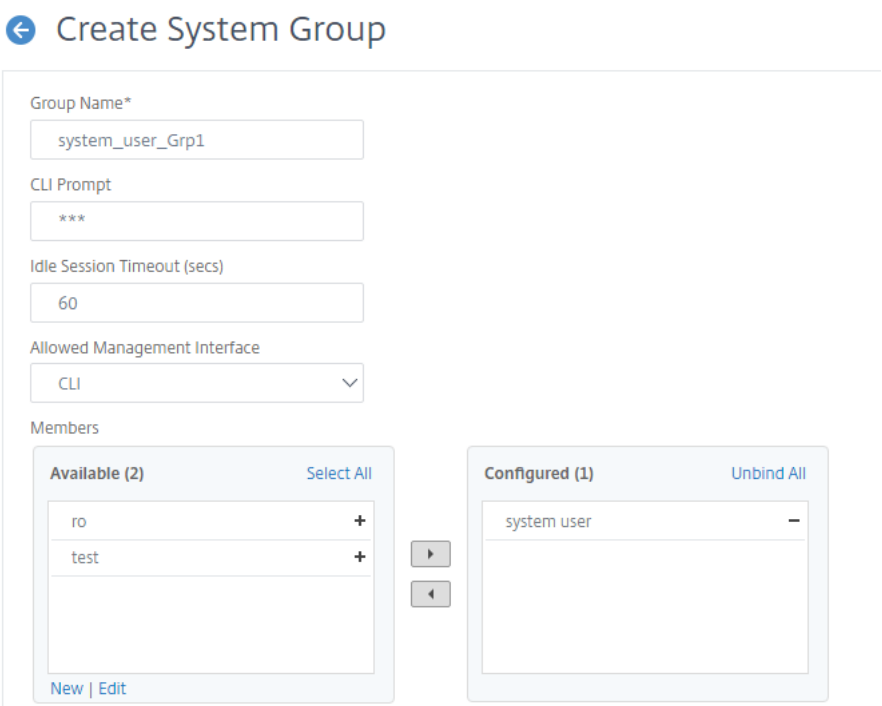
- `bind system group <groupName> -userName <userName>`
- `show system group <groupName>`

Example:

```
> bind system group Managers -userName user1
```

Configure a user group by using the Citrix ADC GUI

1. Navigate to **System** > **User Administration** > **Groups**, and create the user group.
2. In the details pane, click **Add** to create a system user group.
3. In the **Create System Group** page, set the following parameters:
   a) Group Name. Name of the user group.
   b) CLI Prompt. The prompt that you prefer to set for the CLI interface access.
   c) Idle Session Timeout (secs). Set the amount of time a user can be inactive before the session times out and closes.
   d) Allowed Management Interface. Select the Citrix ADC interfaces for which the user group is granted permission to access.
   e) Members. Add user accounts to the group.
   f) Command Policies. Bind command policies to the user group.
   g) Partitions. Bind partitions to the user group.
4. Click **Create** and **Close**.



> **Note:**
>
> To add members to the group, in the Members section, click **Add**. Select users from the Available list and add them to the Configured list.

Configure command policies

Command policies regulate which commands, command groups, virtual servers, and other entities that users and user groups are permitted to use.

The appliance provides a set of built-in command policies, and you can configure custom policies. To apply the policies, you bind them either to users or to groups.

Here are the key points to keep in mind when defining and applying command policies.

- You cannot create global command policies. Command policies must be bound directly to the users and groups on the appliance.
- Users or groups with no associated command policies are subject to the default (DENY-ALL) command policy, and are therefore unable to run any configuration commands until the proper command policies are bound to their accounts.
- All users inherit the policies of the groups to which they belong.
- You must assign a priority to a command policy when you bind it to a user account or group account. This enables the appliance to determine which policy has priority when two or more conflicting policies apply to the same user or group.

- The following commands are available by default to any user and are unaffected by any command you specify:
- help, show CLI attribute, set CLI prompt, clear CLI prompt, show CLI prompt, alias, unalias, history, quit, exit,whoami, config, set CLI mode, unset CLI mode, and show CLI mode.

The following table describes the built-in policies.

Policy name	Allows
read-only	Read-only access to all show commands except show ns runningConfig, show ns ns.conf, and the show commands for the Citrix ADC command group.
operator	Read-only access and access to commands to enable and disable services and servers.
network	Full access, except to the set and unset SSL commands, show ns ns.conf, show ns runningConfig, and show gslb runningConfig commands.
sysadmin	[Included in Citrix ADC 12.0 and later] A sysadmin is lower than a superuser is terms of access allowed on the appliance. A sysadmin user can perform all Citrix ADC operations with the following exceptions: no access to the Citrix ADC shell, cannot perform user configurations, cannot perform partition configurations, and some other configurations as stated in the sysadmin command policy.
superuser	Full access. Same privileges as the nsroot user.

Create custom command policies

Regular expression support is offered for users with the resources to maintain more customized expressions, and for those deployments that require the flexibility that regular expressions offer. For most users, the built-in command policies are sufficient. Users who need more levels of control but are unfamiliar with regular expressions might want to use only simple expressions, such as those in the examples provided in this section, to maintain policy readability.

When you use a regular expression to create a command policy, keep the following in mind.

- When you use regular expressions to define commands that is affected by a command policy, you must enclose the commands in double quotation marks. For example, to create a command policy that includes all commands that begin with show, type the following:
- “^show .*$”
- To create a command policy that includes all commands that begin with rm, type the following:
- “^rm .*$”
- Regular expressions used in command policies are not case sensitive.

The following table lists examples of regular expressions for Command Policies:

Command specification	Matches these commands
“^rm\s+.*$”	All remove actions, because all remove actions begin with the rm string, followed by a space and more parameters such as command groups, command object types, and arguments.
“^show\s+.*$”	All show commands, because all show actions begin with the show string, followed by a space and more parameters such as command groups, command object types, and arguments.
“^shell$”	The shell command alone, but not combined with any additional parameters such as command groups, command object types, and arguments.
“^add\s+vserver\s+.*$”	All create virtual server actions, which consist of add virtual server command followed by a space and more parameters such as command groups, command object types, and arguments.
“^add\s+(lb\s+vserver)\s+.*”	All create lb virtual server actions, which consist of the add lb virtual server command followed by a space and more parameters such as command groups, command object types, and arguments.

For information about built-in command policies, see table Built-in command policy table.

To create a command policy by using the command line interface

At the command prompt, type the following commands to create a command policy and verify the configuration:

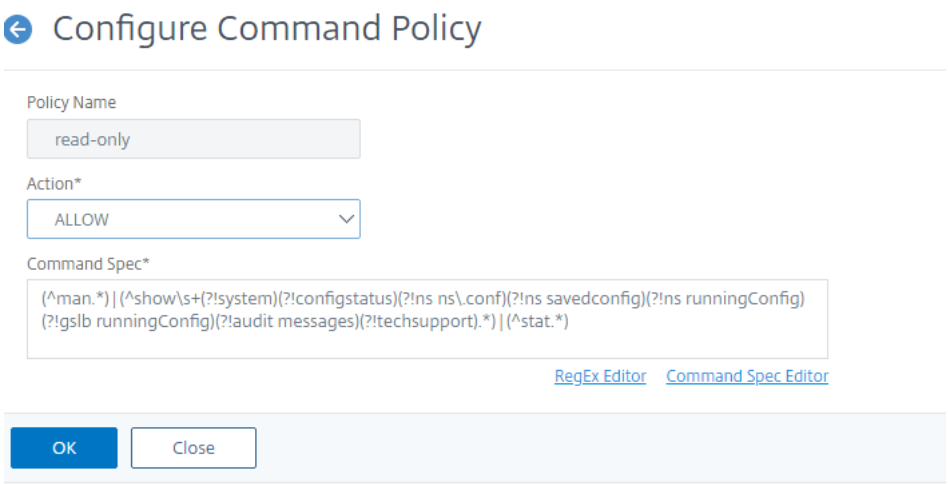
- `add system cmdPolicy <policyname> <action> <cmdspec>`
- `show system cmdPolicy <policyName>`

Example:

```
add system cmdPolicy USER-POLICY ALLOW (\ server\ )|(\ service(Group)*\ )
|(\ vserver\ )|(\ policy\ )|(\ policylabel\ )|(\ limitIdentifier\ )|(^show\
 (?!(system|ns\ (ns.conf|runningConfig)))|(save)|(stat\ .*serv)
```

Configure a command policy by using the Citrix ADC GUI

1. Navigate to **System** > **User Administration** > **Command Policies**.
2. In the details pane, click **Add** to create a new command policy.
3. In the **Configure Command Policy** page, set the following parameters:
   a) Policy name
   b) Action
   c) Command Spec.
4. Click **OK**.



Bind command policies to user accounts and user groups

Once you have defined your command policies, you must bind them to the appropriate user accounts and groups. When you bind a policy, you must assign it a priority so that the appliance can determine which command policy to follow when two or more applicable command policies are in conflict.

Command policies are evaluated in the following order:

- Command policies bound directly to users and the corresponding groups are evaluated according to a priority number. A command policy with a lower priority number is evaluated before one with a higher priority number. Therefore, any privileges the lower-numbered command policy explicitly grants or denies are not overridden by a higher-numbered command policy.
- When two command policies, one bound to a user account and other to a group, have the same priority number, the command policy bound directly to the user account is evaluated first.

To bind command policies to a user by using the command line interface

At the command prompt, type the following commands to bind a command policy to a user and verify the configuration:

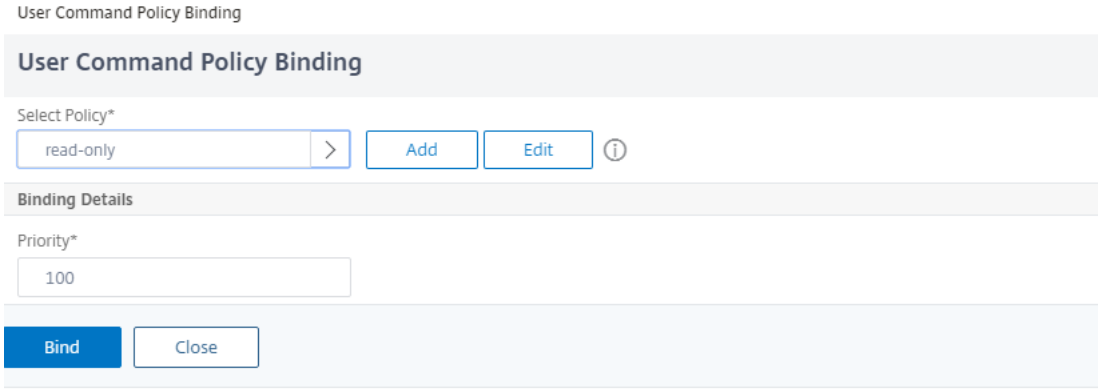
- `bind system user <userName> -policyName <policyName> <priority>`
- `show system user <userName>`

Example:

```
> bind system user user1 -policyName read_all 1
```

Bind command policies to a user account by using the Citrix ADC GUI

Navigate to **System** > **User Administration** > **Users**, select the user and bind command policies.



Optionally, you can modify the default priority to ensure that the policy is evaluated in the proper order.

To bind command policies to a group by using the command line interface

At the command prompt, type the following commands to bind a command policy to a user group and verify the configuration:

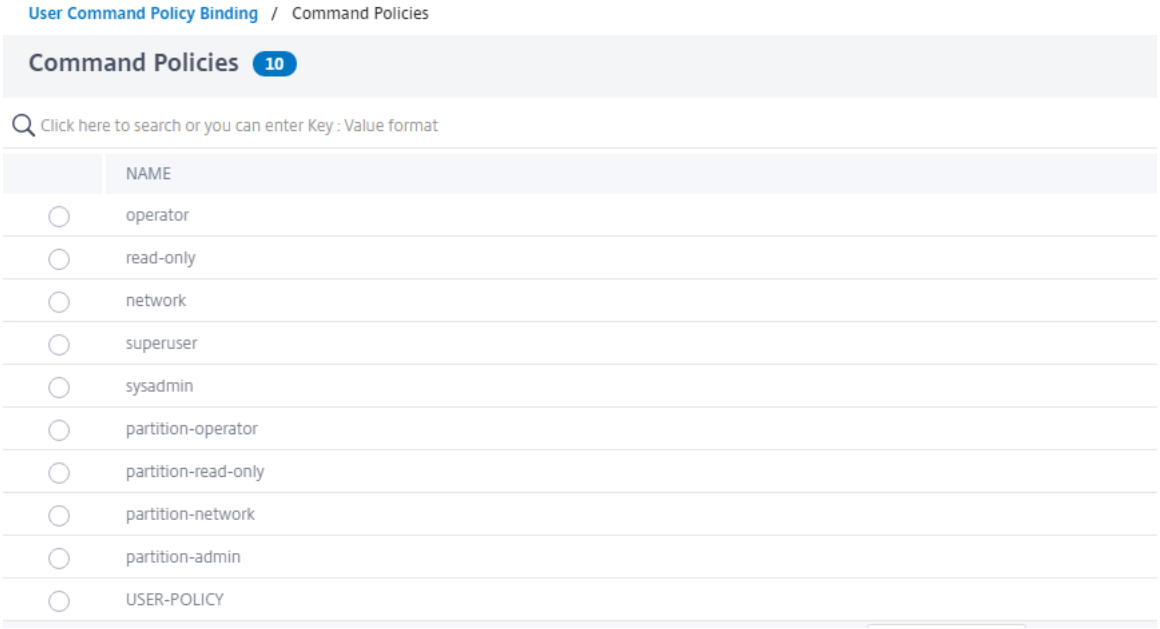
- `bind system group <groupName> -policyName <policyName> <priority>`
- `show system group <groupName>`

Example:

```
> bind system group Managers -policyName read_all 1
```


Bind command policies to a user group by using the Citrix ADC GUI

Navigate to **System** > **User Administration** > **Groups**, select the group and bind command policies.



Optionally, you can modify the default priority to ensure that the policy is evaluated in the proper order.

Example use case: Manage user accounts, user groups, and command policies in a manufacturing organization

The following example shows how to create a complete set of user accounts, groups, and command policies and bind each policy to the appropriate groups and users. The company, Example Manufacturing, Inc., has three users who can access the Citrix ADC appliance:

- **John Doe.** The IT manager. John must be able to see all parts of the Citrix ADC configuration but does not need to modify anything.
- **Maria Ramiez.** The lead IT administrator. Maria must be able to see and modify all parts of the Citrix ADC configuration except for Citrix ADC commands (which local policy dictates must be performed while logged on as nsroot).
- **Michael Baldrock.** The IT administrator in charge of load balancing. Michael must be able to see all parts of the Citrix ADC configuration, but must modify only the load balancing functions.

The following table shows the breakdown of network information, user account names, group names, and command policies for the sample company.

Field	Value	Note
Citrix ADC host name	ns01.example.net	N/A
User accounts	johnd, mariar, and michaelb	John Doe, IT manager, Maria Ramirez, IT administrator and Michael Baldrock, IT administrator.
Groups	Managers and SysOps	All managers and all IT administrators.
Command Policies	read_all, modify_lb, and modify_all	Allow complete read-only access, Allow modify access to load balancing, and Allow complete modify access.

The following description walks you through the process of creating a complete set of user accounts, groups, and command policies on the Citrix ADC appliance named ns01.example.net.

The description includes procedures for binding the appropriate user accounts and groups to one another, and binding appropriate command policies to the user accounts and groups.

This example illustrates how you can use prioritization to grant precise access and privileges to each user in the IT department.

The example assumes that initial installation and configuration have already been performed on the Citrix ADC.

Configure user accounts, groups, and command policies for a sample organization

1. Use the procedure described in Configuring User Accounts section to create user accounts **johnd**, **mariar**, and **michaelb**.
2. Use the procedure described in Configuring User Groups to create user groups **Managers** and **SysOps**, and then bind the users **mariar** and **michaelb** to the **SysOps** group and the user **johnd** to the **Managers** group.
3. Use the procedure described in Creating Custom Command Policies to create the following command policies:
   - **read_all** with action **Allow** and command spec `"(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)"`
   - **modify_lb** with action as **Allow** and the command spec `"^set\s+lb\s+.*$"`
   - modify_all with action as Allow and the command spec `"^\S+\s+(?!system).*"`

4. Use the procedure described in "Binding Command Policies to Users and Groups" to bind the **read_all** command policy to the **SysOps** group, with priority value **1**.
5. Use the procedure described in "Binding Command Policies to Users and Groups" to bind the **modify_lb** command policy to user **michaelb**, with priority value **5**.

The configuration you just created results in the following:

- John Doe, the IT manager, has read-only access to the entire Citrix ADC configuration, but he cannot make modifications.
- Maria Ramirez, the IT lead, has near-complete access to all areas of the Citrix ADC configuration, having to log on only to perform Citrix ADC-level commands.
- Michael Baldrock, the IT administrator responsible for load balancing, has read-only access to the Citrix ADC configuration, and can modify the configuration options for load balancing.

The set of command policies that applies to a specific user is a combination of command policies applied directly to the user's account and command policies applied to one or more groups of which the user is a member.

Each time a user enters a command, the operating system searches the command policies for that user until it finds a policy with an ALLOW or DENY action that matches the command. When it finds a match, the operating system stops its command policy search and allows or denies access to the command.

If the operating system finds no matching command policy, it denies the user access to the command, in accordance with the Citrix ADC appliance's default deny policy.

> **Note:**
>
> When placing a user into multiple groups, take care not to cause unintended user command restrictions or privileges. To avoid these conflicts, when organizing your users in groups, bear in mind the Citrix ADC command policy search procedure and policy ordering rules.

User account and password management

September 14, 2021

Citrix ADC enables you to manage user accounts and password configuration. Following are some of the activities that you can perform for a system user account or `nsroot` administrative user account on the appliance.

- System user account lockout
- Lock system user account for management access
- Unlock a locked system user account for management access

- Disable management access for system user account
- Force password change for `nsroot` administrative users
- Remove sensitive files in a system user account
- Strong password configuration for system users

System user account lockout

To prevent brute force security attacks, you can configure the user lockout configuration. The configuration enables a network administrator to prevent a system user to log on to a Citrix ADC appliance. And also unlock the user account before the lock period expires.

At the command prompt, type:

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <value> -
persistentLoginAttempts (ENABLED | DISABLED)
```

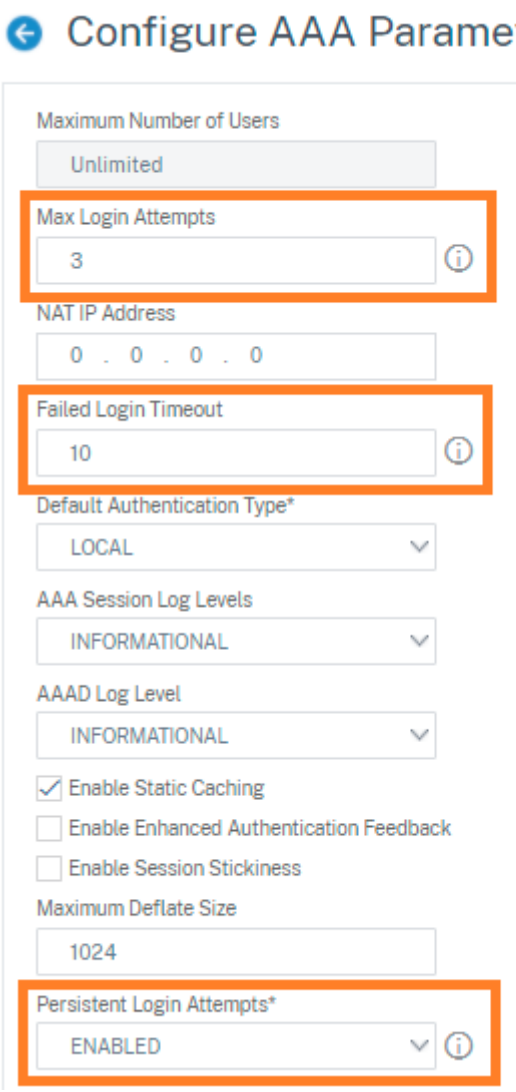
> **Note**
>
> The "persistentLoginAttempts" parameter must be ENABLED to get the details of persistent storage of unsuccessful user login attempts.

Example:

```
set aaa parameter -maxloginAttempts 3 -failedLoginTimeout 10 -persistentLoginAttempts
 ENABLED
```

Configure system user account lockout by using the GUI

1. Navigate to **Configuration > Security > AAA-Application Traffic > Authentication Settings > Change authentication AAA Settings**.
2. In the **Configure AAA Parameter** page, set the following parameters:
   a) Max Login Attempts. The maximum number of logon attempts allowed for the user to try.
   b) Failed Login Timeout. The maximum number of invalid logon attempts by the user.
   c) Persistent Login Attempts. Persistent storage of unsuccessful user login attempts.
3. Click **OK**.



When you set the parameters, the user account gets locked for 10 minutes for three or more invalid login attempts. Also, the user cannot log on even with valid credentials for 10 minutes.

> **Note**
>
> If a locked user tries to log on to the appliance, an error message, `RBA Authentication Failure: maxlogin attempt reached for test.` displays.

Lock system user account for management access

The Citrix ADC appliance enables you to lock a system user for 24 hours and deny access to the user.

Citrix ADC appliance supports the configuration for both system user and external users.

> **Note**
>
> The feature is supported only if you disable the `persistentLoginAttempts` option in the `aaa` parameter.

At the command prompt type:

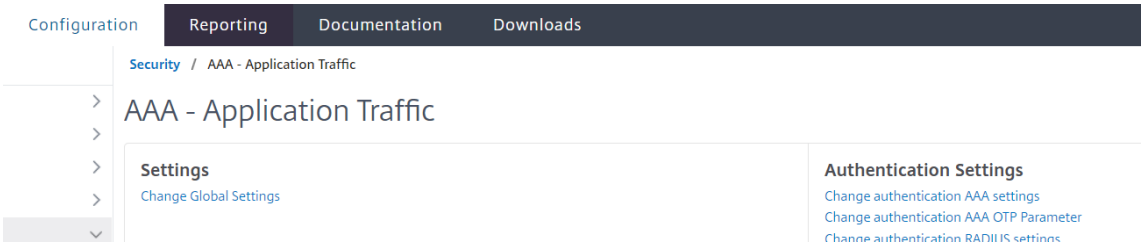
```
set aaa parameter -persistentLoginAttempts DISABLED
```

Now, to lock a user account, at the command prompt, type:

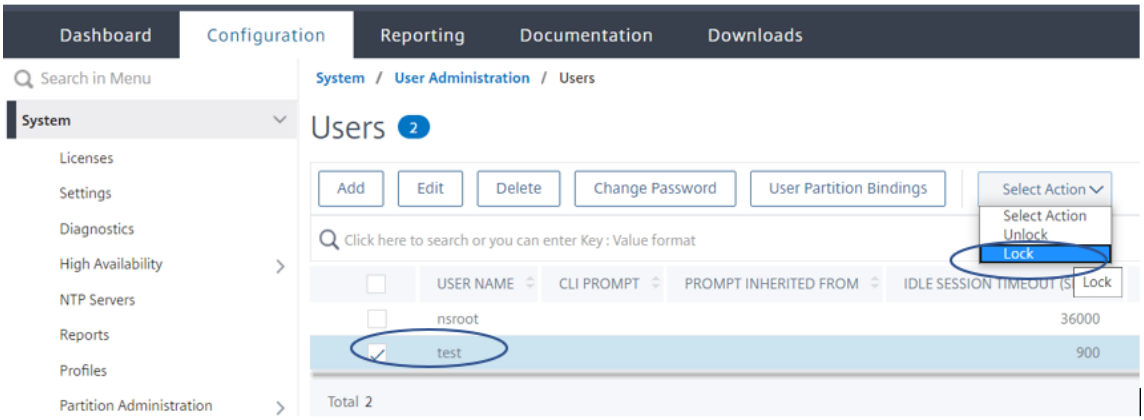
```
lock aaa user test
```

Lock a system user account by using the GUI

1. Navigate to **Configuration > Security > AAA-Application Traffic > Authentication Settings > Change authentication AAA Settings**.



2. In **Configure AAA Parameter**, in the **Persistent Login Attempts** list, select **DISABLED**.
3. Navigate to **System > User Administration > Users**.
4. Select a user.
5. In the Select Action list, select **Lock**.



> **Note**
>
> The Citrix ADC GUI does not have an option to lock external users. To lock an external user, the ADC administrator must use the CLI.

> When a locked system user (locked with lock authentication, authorization, and auditing user command) attempts to log in to Citrix ADC, the appliance displays an error message, “RBA Authentication Failure: User test is locked down for 24 hours.”
>
> When a user is locked to log on to management access, console access is exempted. Locked user is able to log on to console.

Unlock a locked system user account for management access

System users and external users can be locked for 24 hours using the lock authentication, authorization, and auditing user command.

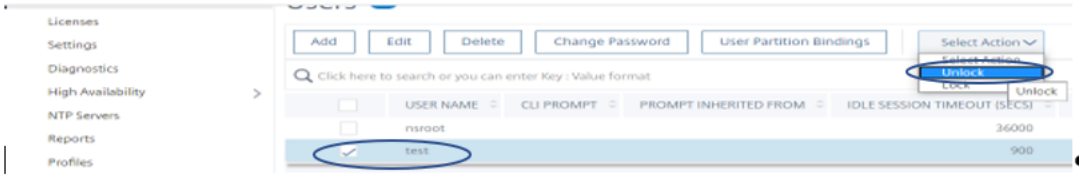
> **Note**
>
> The ADC appliance allows admins to unlock the locked user and the feature does not require any settings in “persistentloginAttempts” command.

At the command prompt, type:

```
unlock aaa user test
```

Configure system user unlock by using the GUI

1. Navigate to **System > User Administration > Users**.
2. Select a user.
3. Click **Unlock**.



The Citrix ADC GUI only lists system users created in the ADC, so there is no option in the GUI to unlock external users. To unlock an external user, the `nsroot` administrator must use the CLI.

Disable management access for system user account

When external authentication is configured on the appliance and as an admin you prefer to deny access to system users to log on to management access, you must disable the localAuth option in the system parameter.

At the command prompt, type the following:

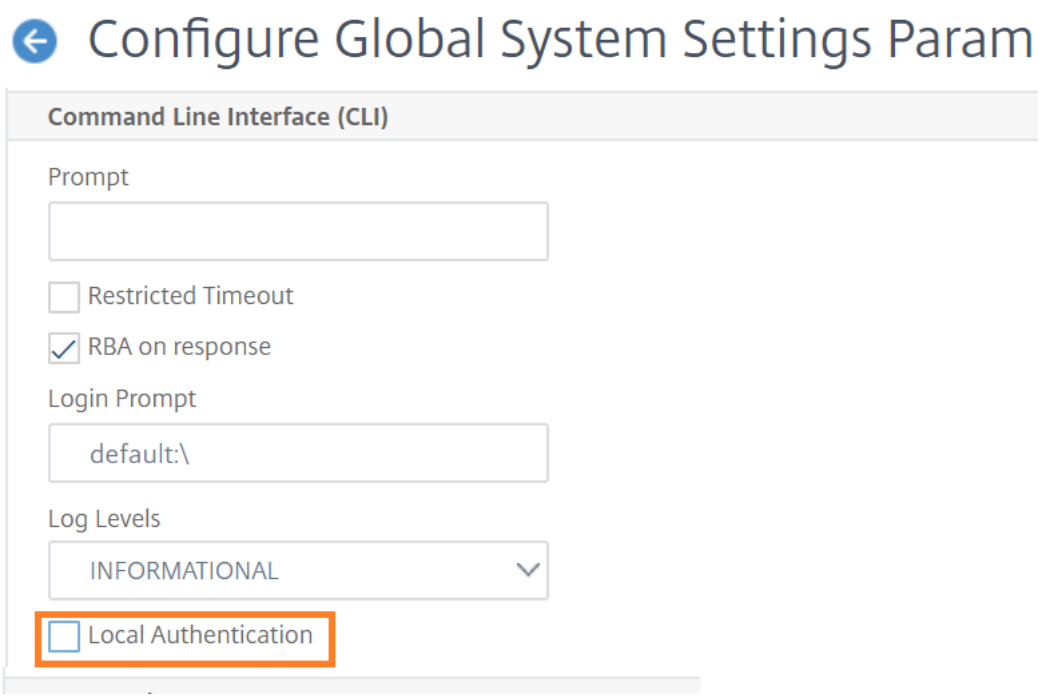
```
set system parameter localAuth <ENABLED|DISABLED>
```


Example:

```
set system parameter localAuth DISABLED
```

Disable management access to system user by using the GUI

1. Navigate to **Configuration > System > Settings > Change Global System Settings**.
2. In **Command Line Interface (CLI)** section, unselect the **Local Authentication** check box.



By disabling the option, local system users cannot log on to ADC management access.

> **Note**
>
> External authentication server must be configured and reachable to disallow local system user authentication in the system parameter. If external server configured in ADC for management access is unreachable, local system users can log on to the appliance. The behavior is set up for recovery purpose.

Force password change for administrative users

For `nsroot` secured authentication, the Citrix ADC appliance prompts the user to change the default password to a new one if the `forcePasswordChange` option is enabled in the system parameter. You can change your `nsroot` password either from CLI or GUI, on your first login with the default credentials.

At the command prompt, type:

```
set system parameter -forcePasswordChange ( ENABLED | DISABLED )
```

SSH session example for NSIP:

```
ssh nsroot@1.1.1.1
Connecting to 1.1.1.1:22...
Connection established.
To escape to local shell, press Ctrl+Alt+].
###############################################################################

WARNING: Access to this system is for authorized users only #
Disconnect IMMEDIATELY if you are not an authorized user! #

###############################################################################

Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
Done
<!--NeedCopy-->
```

Remove sensitive files in a system user account

To manage sensitive data such as authorized keys and public keys for a system user account, you must enable the `removeSensitiveFiles` option. The commands that remove sensitive files when the system parameter is enabled are:

- rm cluster instance
- rm cluster node
- rm high availability node
- clear config full
- join cluster
- add cluster instance

At the command prompt, type:

```
set system parameter removeSensitiveFiles ( ENABLED | DISABLED )
```

Example:

```
set system parameter -removeSensitiveFiles ENABLED
```


Strong password configuration for system users

For secured authentication, the Citrix ADC appliance prompts system users and administrators to set strong passwords to log on to the appliance. The password must be long and must be a combination of:

- One lower case character
- One upper case character
- One numeric character
- One special character

At the command prompt, type:

```
set system parameter -strongpassword <value> -minpasswordlen <value>
```

Where,

`Strongpassword`. After enabling strong password (`enable all` / `enablelocal`) all the passwords or sensitive information must have the following:

- At least 1 lower case character
- At least 1 upper case character
- At least 1 numeric character
- At least 1 special character

Exclude the list in `enablelocal` is - `NS_FIPS`, `NS_CRL`, `NS_RSAKEY`, `NS_PKCS12`, NS_PKCS8, NS_LDAP, `NS_TACACS`, `NS_TACACSACTION`, `NS_RADIUS`, `NS_RADIUSACTION`, `NS_ENCRYPTION_PARAMS` . So no Strong Password checks are performed on these ObjectType commands for the system user.

Possible values: `enableall`, `enablelocal`, disabled
Default value: disabled

`minpasswordlen`. Minimum length of the system user password. When the strong password is enabled by default, the minimum length is 4. User entered value can be greater than or equal to 4. Default minimum value is 1 when the strong password is disabled. Maximum value is 127 in both cases.

Minimum value: 1
Maximum value: 127

Example:

```
set system parameter -strongpassword enablelocal -minpasswordlen 6
```

Default user account

The `nsrecover` user account can be used by the administrator to recover the Citrix ADC appliance. You can log in to the ADC appliance by `nsrecover` if the default system users (`nsroot`) are unable to log in due to any unforeseen issues. The `nsrecover` login is independent of user configurations and

lets you access the shell prompt directly. You are always allowed to log in through the `nsrecover` irrespective of the maximum configuration limit is reached.

How to reset root administrator (nsroot) password

September 14, 2021

The Citrix ADC root administrator (`nsroot`) account provides complete access to all ADC features. So, to preserve security, the administrative account must be used only if necessary.

As an admin, the recommendation is to change your password. If you forget your password, you must first reset to the default one and then change it to a new password.

As a `nsroot` administrator, to reset your password, you must log on to your appliance and change the password. However, if you do not remember the password, you can reboot the appliance in single user mode. Mount the file system in read/write mode, and then remove the **Citrix ADC** entry from the ns.conf file. As a final step, reboot and log on to your appliance with the default one and then set a new password.

Complete the following steps to reset your root administrator password:

1. Connect a computer to the console port of the Citrix ADC and log on.

   > **Note**
   >
   > You cannot log on by using SSH to do this procedure; you must connect directly to the appliance.

2. Reboot the Citrix ADC.
3. Press CTRL+C when the following message appears:

   ```
   Press [Ctrl-C] for command prompt, or any other key to boot immediately
   .
   Booting [kernel] in ## seconds.
   ```

   > **Note**
   >
   > In an Azure serial console, the Citrix ADC appliance does not support single boot until the ADC appliance is booted.

4. Run the following command to start the Citrix ADC in a single user mode:
   `boot -s`

   After the appliance boots, it displays the following message:

   Enter full path name of shell or `RETURN for /bin/sh:`

5. Press ENTER to display the # prompt, and type the following commands to mount the file systems:
   a) Run the following command to check the disk consistency:

      ```
      fsck_ufs /dev/ad0s1a
      ```

      > **Note**
      >
      > Your flash drive has a specific device name depending on your Citrix ADC; so, you have to replace ad0s1a in the preceding command with the appropriate device name.

   b) Access the dev directory and enter 'ls' to check the drive details.
   c) Run the following command to display the mounted partitions:

      ```
      df
      ```

      > **Note**
      >
      > If the flash partition is not listed, you must mount it manually.

   d) Run the following command to mount the flash drive:

      ```
      mount dev/ad0s1a /flash
      ```

6. Run the following command to change to the `nsconfig` directory:

   ```
   cd /flash/nsconfig
   ```

7. Run the following commands to rewrite the ns.conf file and remove the set of system commands defaulting to the admin:
   a) Run the following command to create a configuration file that does not have commands defaulting to the administrator:

      ```
      grep -v "set system user nsroot" ns.conf > new.conf
      ```

   b) Run the following command to make a backup of the existing configuration file:

      ```
      mv ns.conf old.ns.conf
      ```

   c) Run the following command to rename the new.conf file to ns.conf:

      ```
      mv new.conf ns.conf
      ```

8. Run the following command to reboot the Citrix ADC:

   ```
   reboot
   ```

9. Log on using the default administrator credentials.
10. Run the following command to reset the administrator password:

    ```
    set system user nsroot <New_Password>
    ```


> **Note**
>
> To use the "?" character in a password string, precede this character with the \ character.

For example, `yourexamplepasswd\?` is set for the administrator account after you perform the following operation:

```
> set system user nsroot yourexamplepasswd\?
```

> **Note**
>
> For resetting a forgotten (`nsroot`) password in a high availability setup, Citrix recommends you to shut down the peer node. If the peer node is active, the password is overwritten, as the config sync is triggered when the node comes up after reboot.

Also, read Citrix article, CTX224027 to know how secure SSH access to Citrix ADC appliance works.

External user authentication

September 14, 2021

Authentication service in a Citrix ADC appliance can be local or external. In external user authentication, the appliance uses an external server such as LDAP, RADIUS, or TACACS+ to authenticate the user. To authenticate an external user and grant the user access into the appliance, you must apply an authentication policy. The Citrix ADC system authentication uses Advanced authentication policies with advanced policy expressions. The Advanced authentication policies are also used for the system user management in a partitioned Citrix ADC appliance.

> **Note**
>
> If your appliance is still using Classic policies and its expressions, you must stop using it and migrate your Classic policy usage to the Advanced policy infrastructure.

Once you create an authentication policy, you must bind it to the system global entity. You can configure an external authentication server (for example, TACACS) by binding a single authentication policy to the system global entity. Or, you can configure a cascade of authentication servers by binding multiple policies to the system global entity.

> **Note**
>
> When an external user logs into the appliance, the system generates an error message, "User does not exist" in the `ns.log` file. The occurrence is because the system runs the systemuser_systemcmdpolicy_binding command to initialize the GUI for the user.

LDAP authentication (using external LDAP servers)

You can configure the Citrix ADC appliance to authenticate user access with one or more LDAP servers. LDAP authorization requires identical group names in the Active directory, on the LDAP server, and on the appliance. The characters and case must also be the same.

For more information about LDAP authentication policies, see LDAP authentication policies topic.

By default, LDAP authentication is secured by using the SSL/TLS protocol. There are two types of secure LDAP connections. In the first type, the LDAP server accepts the SSL/TLS connection on a port separate from the port used to accept clear LDAP connections. After users establish the SSL/TLS connection, LDAP traffic can be sent over the connection. The second type allows both unsecure and secure LDAP connections and the single port handles it on the server. In this scenario, to create a secure connection, the client first establishes a clear LDAP connection. Then the **LDAP** command StartTLS is sent to the server over the connection. If the LDAP server supports StartTLS, the connection is converted to a secure LDAP connection by using TLS.

The port numbers for LDAP connections are:

- 389 for unsecured LDAP connections
- 636 for secure LDAP connections
- 3268 for Microsoft unsecure LDAP connections
- 3269 for Microsoft secure LDAP connections

LDAP connections that use the StartTLS command use port number 389. If port numbers 389 or 3268 are configured on the appliance, it tries to use StartTLS to make the connection. If any other port number is used, connection attempts use SSL/TLS. If StartTLS or SSL/TLS cannot be used, the connection fails.

When configuring the LDAP server, the case of the alphabetic characters must match that on the server and on the appliance. If the root directory of the LDAP server is specified, all subdirectories are also searched to find the user attribute. In large directories, it can affect performance. For this reason, Citrix recommends that you use a specific organizational unit (OU).

The following table lists examples of the base distinguished name (DN).

LDAP server	Base DN
Microsoft Active Directory	DC=Citrix, DC=local
Novell eDirectory	dc=Citrix, dc=net
IBM Directory Server	cn=users
Lotus Domino	OU=City, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	ou=People, dc=Citrix, dc=com

The following table lists examples of the bind distinguished name (DN).

LDAP server	Bind DN
Microsoft Active Directory	CN=Administrator, CN=Users, DC=Citrix, DC=local
Novell eDirectory	cn=admin, dc=Citrix, dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

LDAP server	Bind DN
Microsoft Active Directory	CN=Administrator, CN=Users, DC=Citrix, DC=local
Novell eDirectory	cn=admin, dc=Citrix, dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

Configure LDAP user authentication by using the CLI

Complete the following steps to configure LDAP authentication for external users

Configure LDAP policy

At the command prompt, do the following:

Step 1: Create an LDAP action.

```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> } } >] [-authTimeout <positive_integer>] [-ldapBase
<string>] [-ldapBindDn <string>] { -ldapBindDnPassword } [-ldapLoginName <
string>] [-groupAttrName <string>] [-subAttributeName <string>]
```

Example:


```
add authentication ldapAction ldap_act -serverIP <IP> -authTimeout 30 -
ldapBase "CN=xxxxx,DC=xxxx,DC=xxx"-ldapBindDn "CN=xxxxx,CN=xxxxx,DC=xxxx,DC
=xxx"-ldapBindDnPassword abcd -ldapLoginName sAMAccountName -groupattrName
memberOf -subAttributeName CN
```

For parameter description, see Authentication and authorization command reference topic.

Step 2: Create a classic LDAP policy.

```
add authentication ldapPolicy <name> <rule> [<reqAction>]
```

Example:

```
add authentication ldappolicy ldap_pol_classic ns_true ldap_act
```

> **Note**
>
> You can configure using a classic or an advanced LDAP policy but Citrix recommends you to use advanced authentication policy because classic policies are deprecated from the Citrix ADC 13.0 release onwards.

Step 3: Create an advanced LDAP policy

```
add authentication Policy <name> <rule> [<reqAction>]
```

Example:

```
add authentication policy ldap_pol_advance -rule true -action ldap_act
```

Step 4: Bind the LDAP policy to system global

At the command line prompt, do the following:

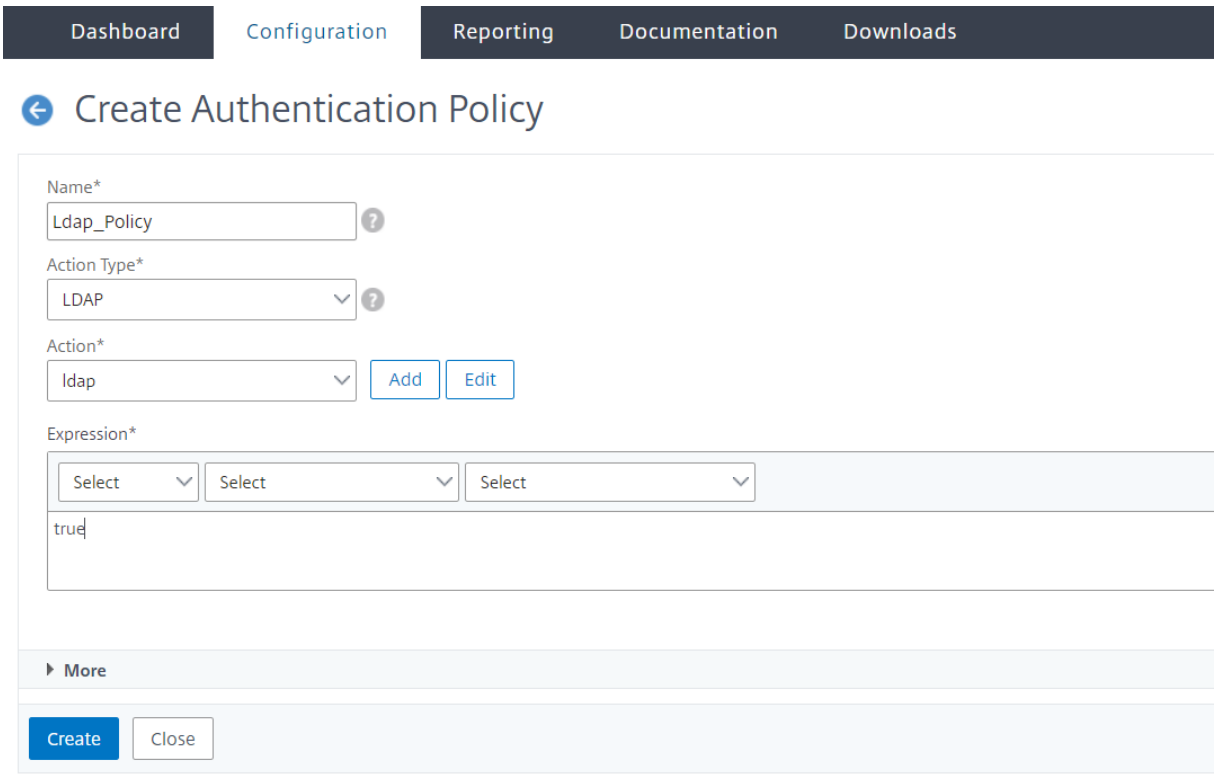
```
bind system global <policyName> [-priority <positive_integer]
```

Example:

```
bind system global ldap_pol_advanced -priority 10
```

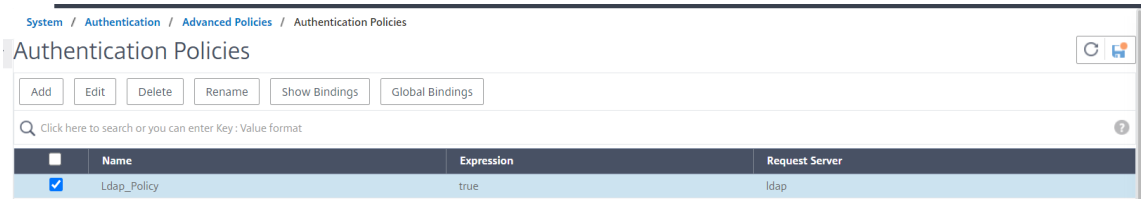
Configure LDAP user authentication by using the Citrix ADC GUI

1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Policy**.
2. Click **Add** to create an authentication policy of type LDAP.
3. Click **Create** and **Close**.

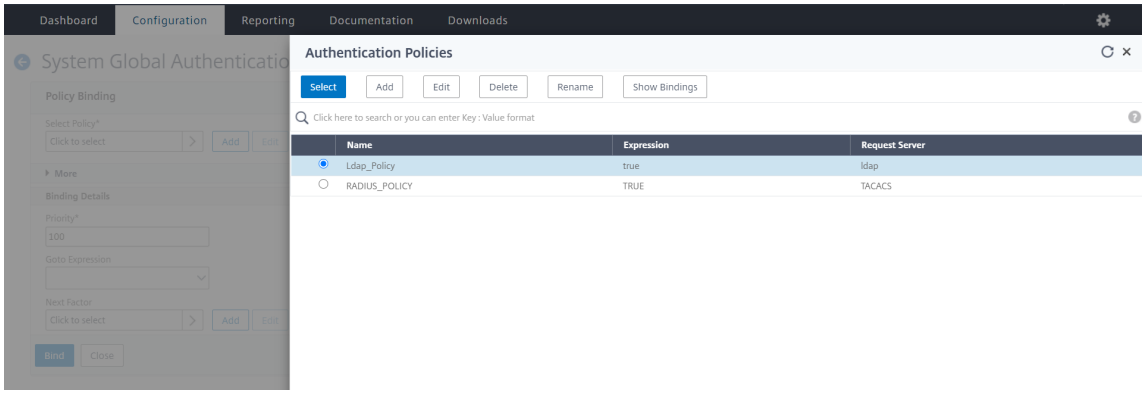


Bind an authentication policy to the system global for LDAP authentication using the Citrix ADC GUI

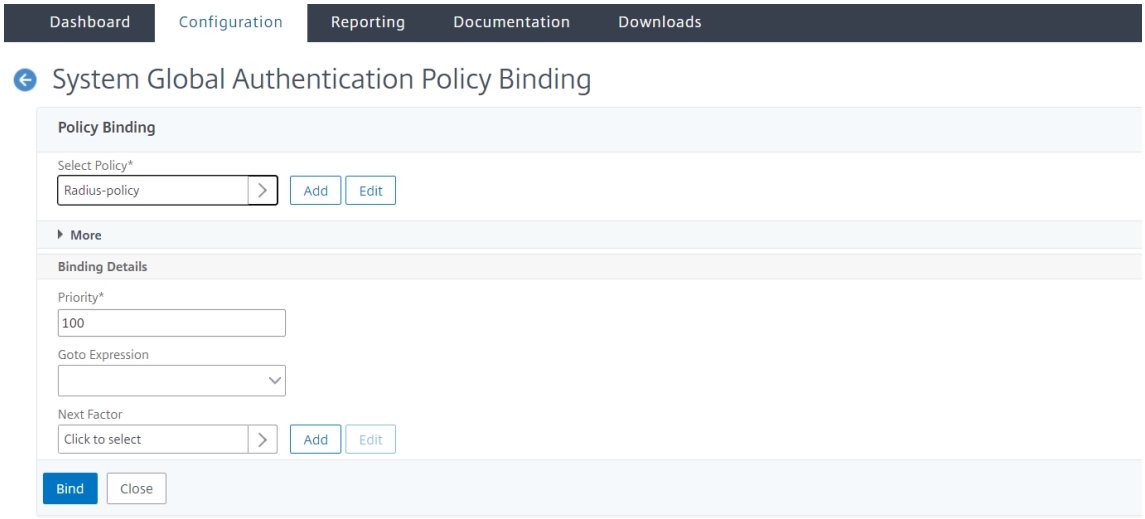
1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Authentication PoliciesPolicy**.
2. In the details pane, click **Global Bindings** to create system global authentication policy binding.
3. Click **Global Bindings**.



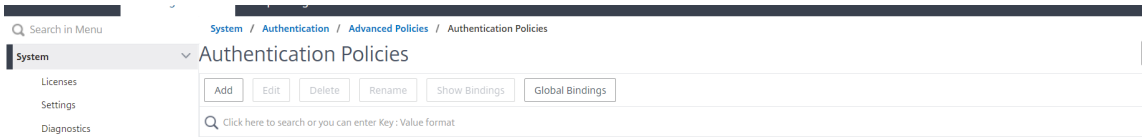
4. Select an authentication profile.



5. Select the LDAP policy.
6. In the **System Global Authentication Policy Binding** page, set the following parameters:
   a) Select Policy.
   b) Binding Details



7. Click **Bind** and **Done**.
8. Click **Global Bindings** to confirm the policy bounded to the system global.



Determining attributes in the LDAP directory

If you need help with determining your LDAP directory attributes, you can easily look them up with the free LDAP browser from Softerra.

You can download the LDAP browser from the Softerra LDAP Administrator website at <http://www.ldapbrowser.com>. After the browser is installed, set the following attributes:

- The host name or IP address of your LDAP server.
- The port of your LDAP server. The default is 389.
- The base DN field can be left blank.
- The information provided by the LDAP browser can help you determine the base DN needed for the Authentication tab.
- The Anonymous Bind check determines whether the LDAP server requires user credentials for the browser to connect to it. If the LDAP server requires credentials, leave the check box cleared.

After completing the settings, the LDAP browser displays the profile name in the left pane and connects to the LDAP server.

For more information, see LDAP topic.

Key-based authentication support for LDAP users

With key-based authentication, you can now fetch the list of public keys that are stored on the user object in the LDAP server through SSH. The Citrix ADC appliance during the role-based authentication (RBA) process must extract public SSH keys from the LDAP server. The retrieved public key, which is compatible with SSH, must allow you to log in through the RBA method.

A new attribute "sshPublicKey" is introduced in the "add authentication ldapAction" and "set authentication ldapAction" commands. By using this attribute, you can obtain the following benefits:

- Can store the retrieved public key, and the LDAP action uses this attribute to retrieve SSH key information from LDAP server.
- Can extract attribute names of up to 24 KB.

> **Note**
>
> The external authentication server, such as LDAP is used only to retrieve SSH key information. It is not used for authentication purpose.

Following is an example of the flow of events through SSH:

- SSH daemon sends an AAA_AUTHENTICATE request with password field empty to authentication, authorization, and auditing daemon port.
- If LDAP is configured to store the SSH public key, authentication, authorization, and auditing responds with "sshPublicKey" attribute along with other attributes.
- SSH daemon verifies these keys with the client keys.
- SSH daemon passes user name in the request payload, and authentication, authorization, and auditing returns the keys specific to this user along with generic keys.

To configure the sshPublicKey attribute, at the command prompt type the following commands:

- With add operation, you can add “sshPublicKey” attribute while configuring ldapAction command.

```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr
|*> | { -serverName <string> } } [-serverPort <port>] … [-Attribute1 <
string>] … [-Attribute16 <string>][-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

- With set operation, you can configure “sshPublicKey” attribute to an already added ldapAction command.

```
set authentication ldapAction <name> [-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

RADIUS authentication (using external RADIUS servers)

You can configure the Citrix ADC appliance to authenticate user access with one or more RADIUS servers. If you are using RSA SecurID, SafeWord, or Gemalto Protiva products, use a RADIUS server.

For more information about RADIUS authentication policies, see RADIUS authentication policies topic.

Your configuration might require using a network access server IP address (NAS IP) or a network access server identifier (NAS ID). When configuring the appliance to use a RADIUS authentication server, use the following guidelines:

- If you enable use of the NAS IP, the appliance sends its configured IP address to the RADIUS server, rather than the source IP address used in establishing the RADIUS connection.
- If you configure the NAS ID, the appliance sends the identifier to the RADIUS server. If you do not configure the NAS ID, the appliance sends its host name to the RADIUS server.
- When the NAS IP address is enabled, the appliance ignores any NAS ID that it used for communicating with the RADIUS server.

Configure RADIUS user authentication by using the CLI

At the command prompt, do the following:

Step 1: Create an RADIUS action

```
add authentication radiusaction <name> -serverip <ip> -radkey <key> -
radVendorID <id> -radattributetype <value>
```

Where,
`radVendorID` RADIUS vendor ID attribute, used for RADIUS group extraction.
`radAttributeType` RADIUS attribute type, used for RADIUS group extraction.

Example:

```
add authentication radiusaction RADserver531 rad_action -serverip 1.1.1.1 -
radkey key123 -radVendorID 66 -radattributetype 6
```

Step 2: Create a classic RADIUS policy.

```
add authentication radiusPolicy <name> <rule> [<reqAction>]
```

Example:

```
add authentication radiuspolicy radius_pol_classic ns_true radius_act
```

> **Note**
>
> You can configure using a classic or an advanced RADIUS policy. Citrix recommends you to use the advanced authentication policy because classic policies are deprecated from the Citrix ADC 13.0 release onwards.

Step 3: Create an advanced RADIUS policy

```
add authentication policy <policyname> -rule true -action <radius action
name>
```

Example:

```
add authentication policy rad_pol_advanced -rule true -action radserver531rad_action
```

Step 4: Bind the RADIUS policy to the system global.

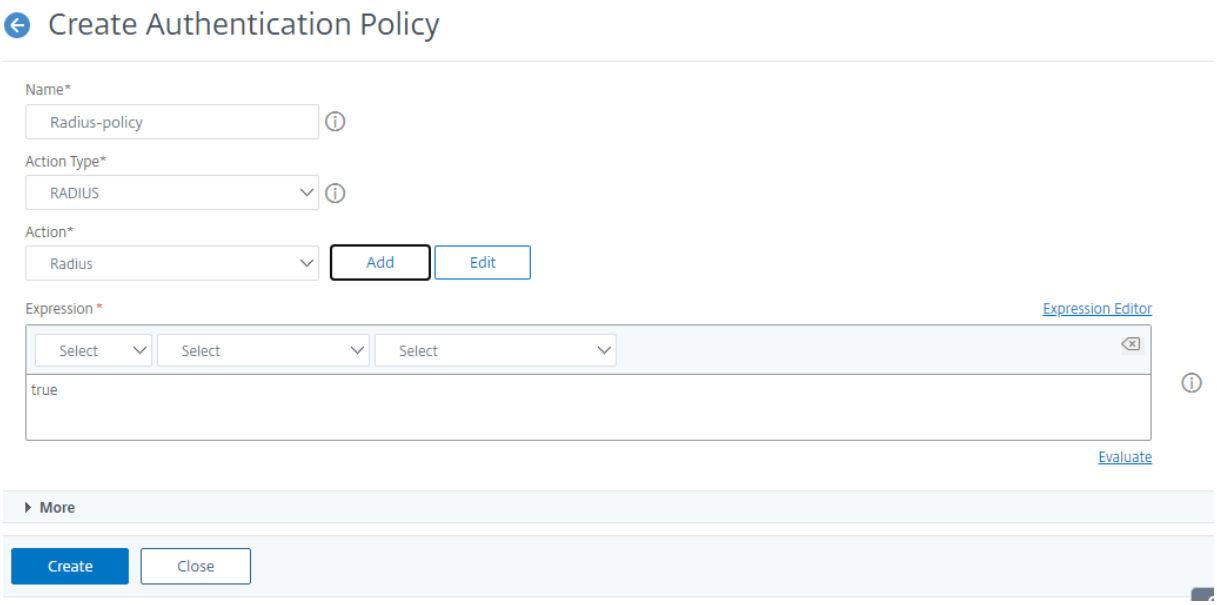
```
bind system global <policyName> -priority <positive_integer
```

Example:

```
bind system global radius_pol_advanced -priority 10
```

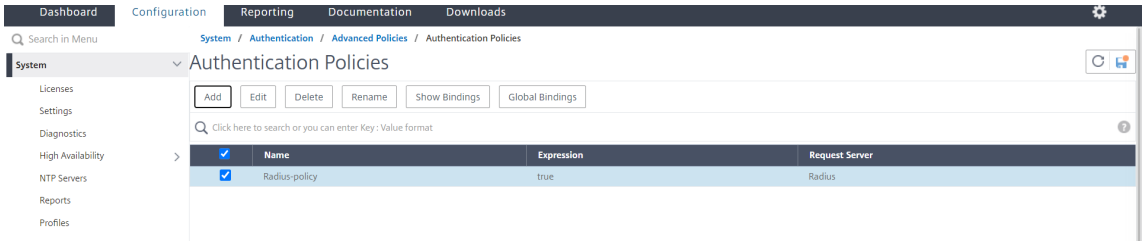
Configure RADIUS user authentication by using the GUI

1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Policy**.
2. Click **Add** to create an authentication policy of type RADIUS.
3. Click **Create** and **Close**.

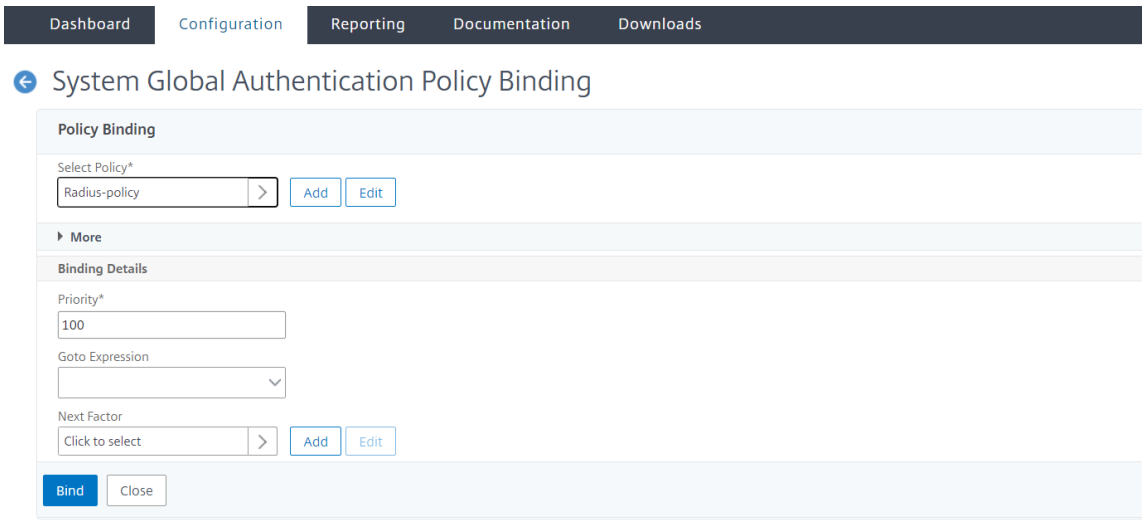


Bind the authentication policy to the system global for RADIUS authentication by using the GUI

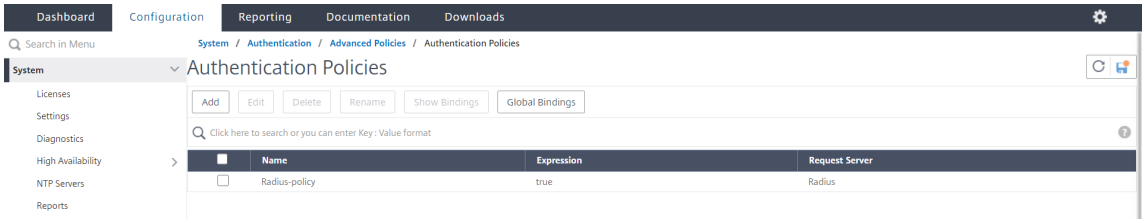
1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Policy**.
2. In the details pane, click **Global Bindings** to create system global authentication policy binding.
3. Click **Global Bindings**.



4. Select RADIUS.
5. In the **System Global Authentication Policy Binding** page, set the following parameters:
   a) Select Policy.
   b) Binding Details.



6. Click **Bind** and **Close**.
7. Click **Global Bindings** to confirm the policy bounded to the system global.



Choose RADIUS user authentication protocols

The Citrix ADC appliance supports implementations of RADIUS that are configured to use any of several protocols for user authentication, including:

- Password Authentication Protocol
- Challenge-Handshake Authentication Protocol (CHAP)
- Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP Version 1 and Version 2)

If your deployment is configured to use RADIUS authentication and your RADIUS server is configured with a Password Authentication Protocol. You can strengthen user authentication by assigning a strong shared secret to the RADIUS server. Strong RADIUS shared secrets consist of random sequences of uppercase and lowercase letters, numbers, and punctuation, and are minimum 22 characters in length. If possible, use a random character generation program to determine RADIUS shared secrets.

To further protect RADIUS traffic, assign a different shared secret to each appliance or virtual server. When you define clients on the RADIUS server, you can also assign a separate shared secret to each client. Also, you must configure separately each policy that uses RADIUS authentication.

Configure IP address extraction

You can configure the appliance to extract the IP address from a RADIUS server. When a user authenticates with the RADIUS server, the server returns a framed IP address that is assigned to the user. The following are attributes for IP address extraction:

- Allows a remote RADIUS server to supply an IP address from the internal network for a user logged on to the appliance.
- Allows configuration for any RADIUS attribute using the type ip-address, including that are vendor encoded.

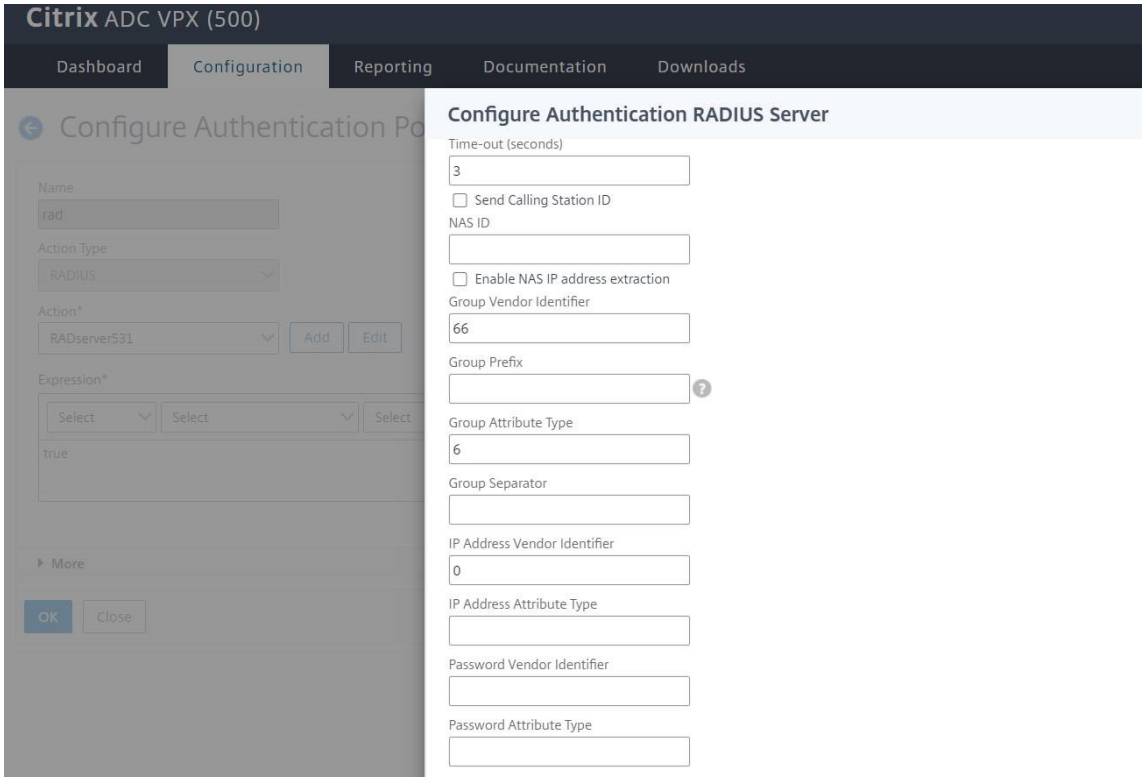
When configuring the RADIUS server for IP address extraction, you configure the vendor identifier and the attribute type.

The vendor identifier enables the RADIUS server to assign an IP address to the client from a pool of IP addresses that are configured on the RADIUS server. The vendor ID and attributes are used to make the association between the RADIUS client and the RADIUS server. The vendor ID is the attribute in the RADIUS response that provides the IP address of the internal network. A value of zero indicates that the attribute is not vendor encoded. The attribute type is the remote IP address attribute in a RADIUS response. The minimum value is one and the maximum value is 255.

A common configuration is to extract the **RADIUS** attribute *framed IP address*. The vendor ID is set to zero or is not specified. The attribute type is set to eight.

Group extraction for RADIUS by using the GUI

1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Radius**, and select a policy.
2. Select or create RADIUS policy.
3. In the **Configure Authentication RADIUS Server** page, set the following parameters.
    a) **Group Vendor Identifier**
    b) **Group Attribute Type**
4. Click **OK** and **Close**.



TACACS+ authentication (using external TACACS+ servers)

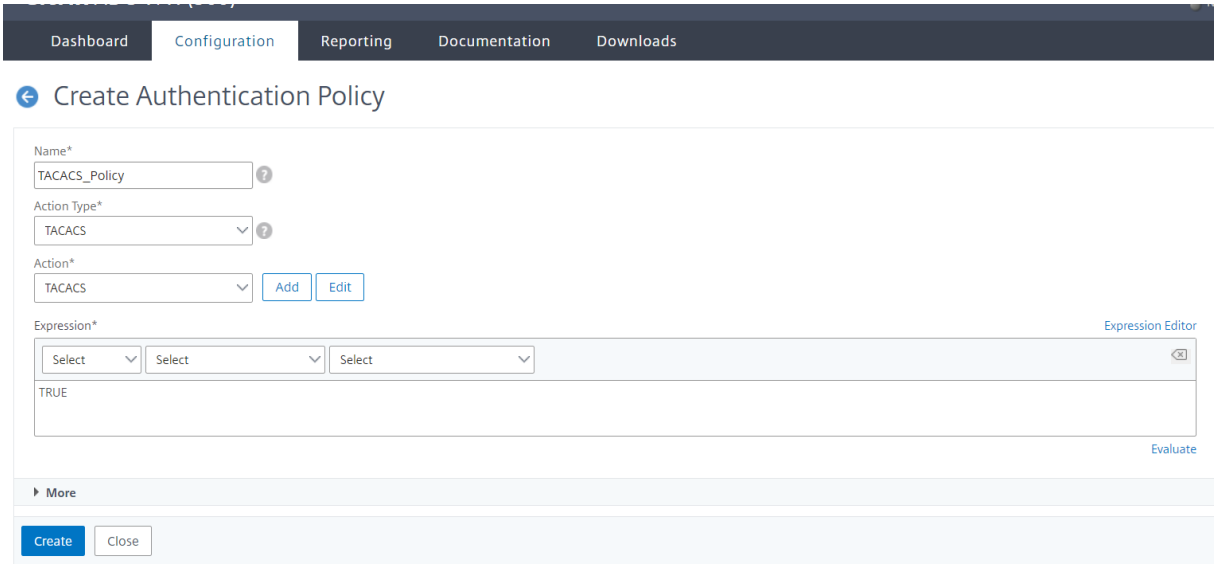
> **Important**
>
> - Citrix recommends you do not modify any TACACS related configurations when you run a "clear ns config" command.
> - TACACS related configuration related to advanced policies is cleared and reapplied when the `RBAconfig` parameter is set to NO in "clear ns config" command for advanced policy.

You can configure a TACACS+ server for authentication. Similar to RADIUS authentication, TACACS+ uses a secret key, an IP address, and the port number. The default port number is 49. To configure the appliance to use a TACACS+ server, provide the server IP address and the TACACS+ secret. You must specify port only when the server port number in use is something other than the default port number of 49.

For more information, see TACACS authentication.

Configure TACACS+ authentication by using the GUI

1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Policy**.
2. Click **Add** to create an authentication policy of type TACACS.
3. Click **Create** and **Close**.



After the TACACS+ server settings are configured on the appliance, bind the policy to the system global entity.

Bind authentication policies to the system global entity by using the CLI

When the authentication policies are configured, bind the policies to the system global entity.

At the command line prompt, do the following:

```
bind system global <policyName> [-priority <positive_integer>]
```

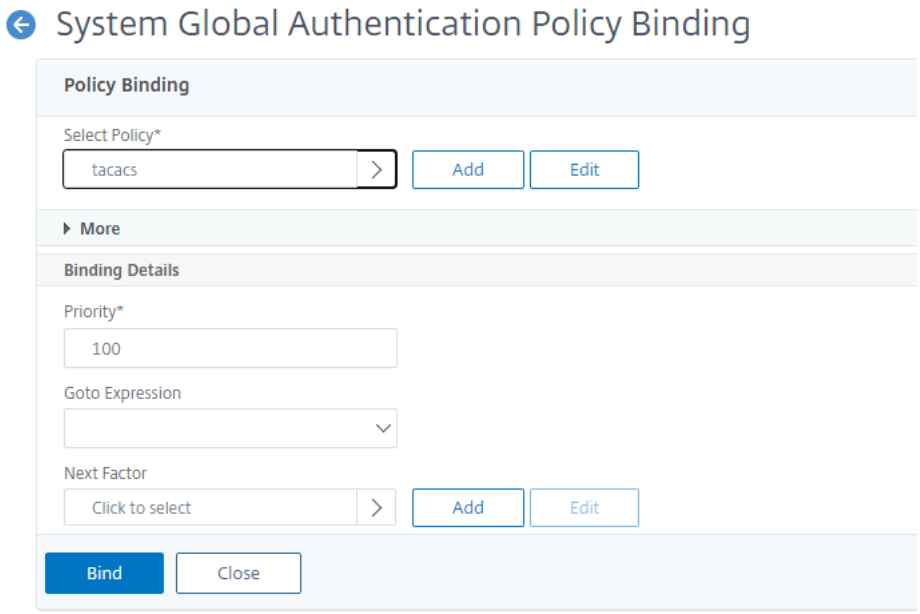
Example:

```
bind system global pol_classic -priority 10
```

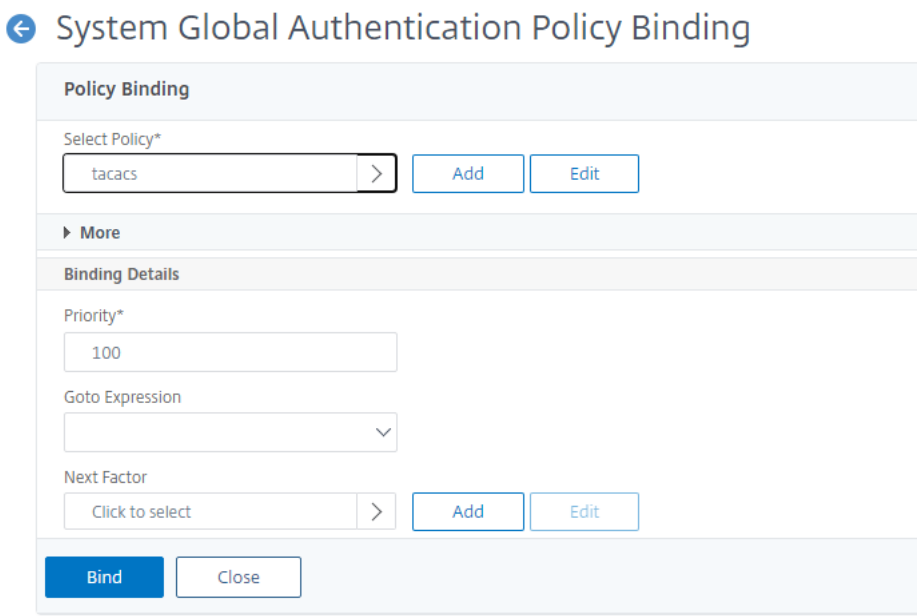
Also, read the Citrix article, CTX113820 to know about external authentication using TACACS.

Bind authentication policies to the system global entity by using the GUI

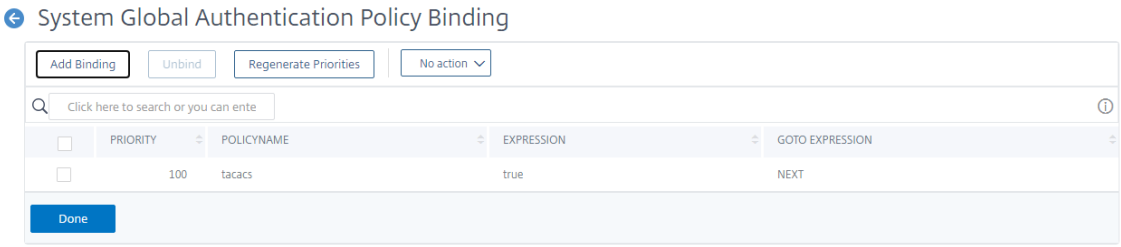
1. Navigate to **System** > **Authentication** > **Advanced Policies** > **Authentication Policies** > **Policy**.
2. In the details pane, click **Global Bindings** to create system global authentication policy binding.
3. Click **Global Bindings**.



4. Select the TACACS policy.
5. In the System Global Authentication Policy Binding page, set the following parameters:
   a) Select Policy.
   b) Binding Details



6. Click **Bind** and **Close**.
7. Click **Global Bindings** to confirm the policy bounded to the system global.



For more information about TACACS group extraction, read Citrix article CTX220024.

Display number of unsuccessful logon attempts for external users

The Citrix ADC appliance displays the number of invalid login attempts to the external user when you attempt at least one unsuccessful login before successfully logging on to the Citrix ADC management console.

> **Note**
>
> Currently, Citrix supports only keyboard interactive authentication for external users with the "persistentLoginAttempts" parameter enabled in the system parameter.

At the command prompt, type:

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <value> -
persistentLoginAttempts (ENABLED | DISABLED )]
```

Example:

```
set aaa parameter -maxloginAttempts 5 -failedLoginTimeout 4 -persistentLoginAttempts
 ENABLED
```

```
1  Following msg will be seen to external user when he tries 1 invalid
       login attempt before successfully login to the ADC management access
       .
2
3  Connection established.
4  To escape to local shell, press 'Ctrl+Alt+]'.
5  ##############################################################################

6  #

       #
7  #         WARNING: Access to this system is for authorized users only
              #
8  #          Disconnect IMMEDIATELY if you are not an authorized user!
               #
```



```
#
   #
################################################################################


WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Mon Aug 24 17:09:00 2020 from 10.10.10.10

The number of unsuccessful login attempts since the last successful
   login : 1
Done
>
The number of unsuccessful login attempts since the last successful
   login : 1
Done
>
<!--NeedCopy-->
```

SSH key-based authentication for local system users

September 14, 2021

To have a secured user access for the Citrix ADC appliance you can have the public key authentication of the SSH server. The SSH key-based authentication is preferred over traditional user name or password based authentication for the following reasons:

- Provides better cryptographic strength than user passwords.
- Eliminates the need of remembering complicated passwords and prevents shoulder-surfing attacks which are possible if passwords are used.
- Provides a password-less login for making automation scenarios more secured.

Citrix ADC supports SSH key-based authentication by applying the public and private key concept. The SSH key-based authentication in Citrix ADC can be enabled either for a specific user or for all local users.

> **Note**
>
> The feature is supported only for Citrix ADC local users and not supported for external users.

SSH key-based authentication for local system users

In a Citrix ADC appliance, an administrator can set up SSH key-based authentication for a secured system access. When a user logs into the Citrix ADC using a private key, the system authenticates the user using the public key configured on the appliance.

Configure SSH key-based authentication for the Citrix ADC local system users by using CLI

Following configuration helps you to configure key-based authentication for Citrix ADC local system users.

1. Log on to a Citrix ADC appliance using administrator credentials.
2. By default your `sshd_config` file accesses this path: **AuthorizedKeysFile /nsconfig/ssh/authorized_keys**.
3. Append the public key to the authorized_keys file: **/nsconfig/ssh/authorized_keys**. The file path for `sshd_config` is `/etc/sshd_config`.
4. Copy the `sshd_config` file into `/nsconfig` to ensure that the changes persist even after restarting the appliance.
5. You can use the following command to restart your `sshd` process.

```
    kill -HUP `cat /var/run/sshd.pid`
<!--NeedCopy-->
```

> **Note**
>
> If the authorized_keys file is not available, you must first create one and then append the public key. **Make sure the file has the following permission for the authorized_keys**.
>
> `root@Citrix ADC## chmod 0644 authorized_keys`

```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
The Regents of the University of California. All rights reserved.
root@ns# cd /nsconfig/ssh
root@ns# vi authorized_keys
### Add public keys in authorized_keys file
<!--NeedCopy-->
```

User-specific SSH key-based authentication for local system users

In a Citrix ADC appliance, an administrator can now set up a user specific SSH key-based authentication for a secured system access. The administrator must first configure the `Authorizedkeysfile`

option in the sshd_config file and then add the public key in the authorized_keys file for a system user.

> **Note**
>
> If the authorized_keys file is not available for a user, the administrator must first create one and then add the public key to it.

Configure user-specific SSH key-based authentication by using the CLI

Following procedure helps you to configure user-specific SSH key-based authentication for Citrix ADC local system users.

1. Log on to a Citrix ADC appliance using administrator credentials.
2. At the shell prompt, access the sshd_config file and add the following configuration line:

   AuthorizedKeysFile ~/.ssh/authorized_keys

   > **Note**
   >
   > The ~ is the home directory and differs for different users. It expands to the different home directory.

3. Change the directory to the system user folder and add the public keys in the authorized_keys file.

   /var/pubkey/<username>/.ssh/authorized_keys

Once you have completed the earlier steps, restart the sshd process on your appliance by the following command:

```
    kill -HUP `cat /var/run/sshd.pid`

<!--NeedCopy-->
```

> **Note**
>
> If the authorized_keys file is not available, you must first create one and then add the public key.

```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
The Regents of the University of California. All rights reserved.
root@ns# cd /var/pubkey/<username>/
root@ns# ls
.ssh
root@ns# cd .ssh
```



```
root@ns# vi authorized_keys
### Add public keys in authorized_keys file

<!--NeedCopy-->
```

Also, read Citrix article, CTX109011 to know how secure SSH access to Citrix ADC appliance works.

Two factor authentication for system users and external users

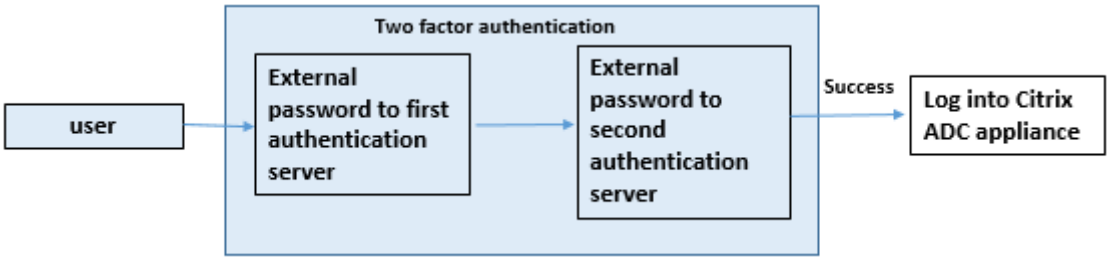
September 14, 2021

Two factor authentication is a security mechanism where a Citrix ADC appliance authenticates a system user at two authenticator levels. The appliance grants access to the user only after successful validation of passwords by both levels of authentication. If a user is authenticated locally, the user profile must be created in the Citrix ADC database. If the user is authenticated externally then, the user name and password must match the user identity registered in the external authentication server.

> **Note**
>
> Two factor authentication feature works only from Citrix ADC 12.1 build 51.16 onwards.

How two factor authentication works

Consider a user trying to log on to a Citrix ADC appliance. The requested application server sends the user name and password to the first external authentication server (RADIUS, TACACS, LDAP, or AD). Once the user name and password are validated, the user is prompted for a second level of authentication. The user can now provide the second password. Only if both passwords are correct, the user is allowed to access the Citrix ADC appliance. The following diagram is an illustration of how two-factor authentication works for a Citrix ADC appliance.



Following are the different use cases for configuring two factor authentication for external and system users.

You can configure two-factor authentication on a Citrix ADC appliance in different ways. The following are the different configuration scenarios for two factor authentication on a Citrix ADC appliance.

1. Two factor authentication (2FA) across Citrix ADC, GUI, CLI, API and SSH.
2. External authentication enabled and local authentication disabled for system users.
3. External authentication enabled with policy based local authentication for system users.
4. External authentication disabled for system users with local authentication enabled.
5. External authentication enabled and local authentication enabled for system users.
6. External authentication enabled for selected LDAP users

Use case 1: Two factor authentication (2FA) across Citrix ADC, GUI, CLI, API and SSH interfaces

Two-factor authentication is enabled and available across all Citrix ADC management access for GUI, API, and SSH.

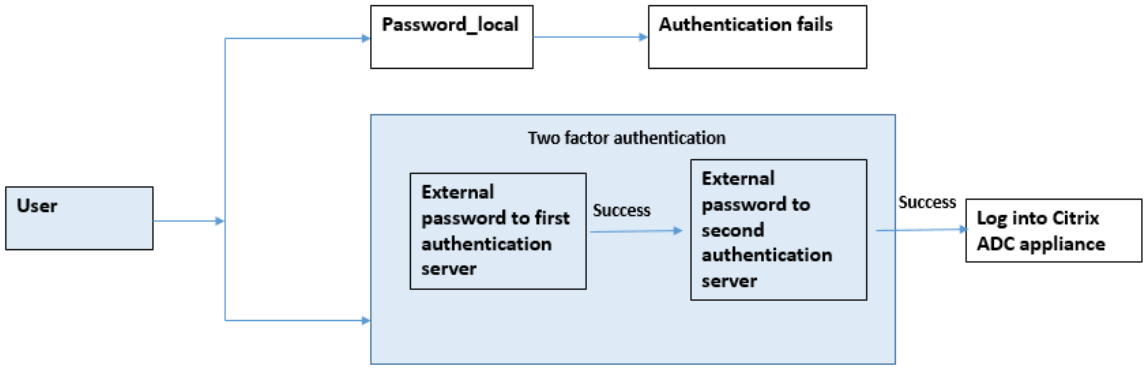
Use case 2: Two factor authentication supported on external authentication servers such as LDAP, RADIUS, Active Directory and TACACS

You can configure two-factor authentication on the following external authentication servers for first-level and second-level user authentication.

- RADIUS
- LDAP
- Active Directory
- TACACS

Use case 3: External authentication enabled and local authentication disabled for system users

You begin the authentication process by enabling the external authentication option and disabling local authentication for system users.



Complete the following steps by using the command line interface:

1. Add authentication action for LDAP policy
2. Add authentication policy for LDAP policy
3. Add authentication action for RADIUS policy
4. Add authentication policy for RADIUS policy
5. Add authentication login schema
6. Add and bind authentication policy label to RADIUS server
7. Bind system global authentication for LDAP policy
8. Disable local authentication in system parameter

Add authentication action for LDAP server (first level authentication)

At the command prompt, type:

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname <
loginname> -groupattrname <grp attribute name> -subAttributename <string>-
ssoNameAttribute <string>
```

Example:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

Add authentication policy for LDAP server (first level authentication)

At the command prompt, type:

```
add authentication policy <ldap policy name> -rule true -action <ldap
action name>
```

Example:

```
add authentication policy pol1 -rule true -action ldapact1
```

Add authentication action for RADIUS server (second level authentication)

At the command prompt, type:

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

Example:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```


Add authentication policy for RADIUS server (second level authentication)

At the command prompt, type:

```
add authentication policy <radius policy name> -rule true -action <rad
action name>
```

Example:

```
add authentication policy radpol11 -rule true -action radact1
```

Add authentication login schema

You can use the “SingleAuth.xml” login schema for system users to provide the second password for the Citrix ADC appliance. At the command prompt, type:

```
add authentication loginSchema <login schema name> -authenticationSchema
LoginSchema/SingleAuth.xml
```

Example:

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

Add and bind authentication policy label to RADIUS server

At the command prompt, type:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

```
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

Example:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
```

```
bind authentication policylabel label1 -policyName radpol11 -priority 1
```

Bind authentication system global for LDAP policy

At the command prompt, type:

```
bind system global ldappolicy -priority <priority> -nextFactor <policy
label name>
```

Example:


```
bind system global pol11 -priority 1 -nextFactor label1
```

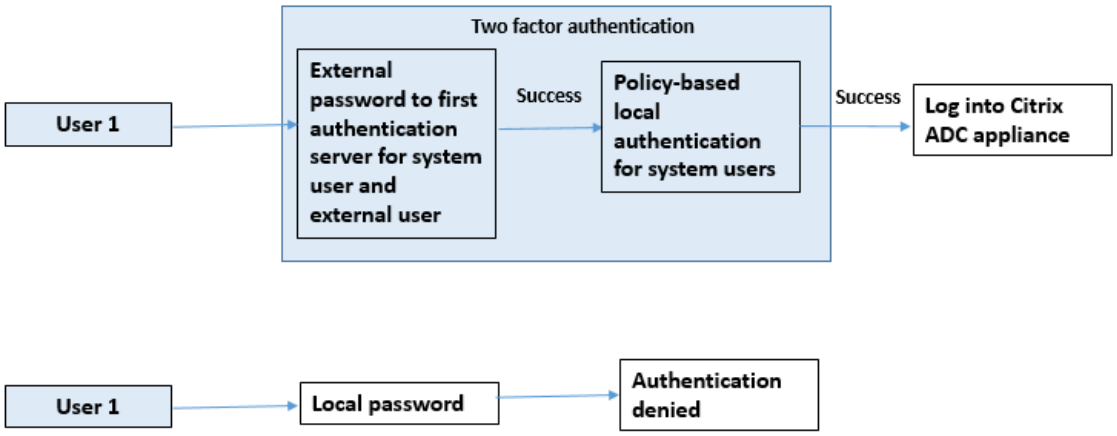
Disable local authentication in system parameter

At the command prompt, type:

```
set system parameter -localauth disabled
```

Use case 4: External authentication enabled for system user with local authentication policy attached

In this scenario, the user is allowed to log on to the appliance using two-factor authentication with local authentication policy evaluation at the second level of user identification.



Complete the following steps by using the command line interface.

1. Add authentication action for LDAP server
2. Add authentication policy for LDAP policy
3. Add local authentication policy
4. Add authentication policy label
5. Bind LDAP policy as system global
6. Disable local authentication in system parameter

Add authentication action for LDAP server (first level authentication)

At the command prompt, type:

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname <
loginname> -groupattrname <grp attribute name> -subAttributename <string>-
ssoNameAttribute <string>
```


Example:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name -ssoNameAttribute name
```

Add authentication policy for LDAP server (first level authentication)

At the command prompt, type:

```
add authentication policy <ldap policy name> -rule true -action <ldap
action name>
```

Example:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

Add local authentication policy for system users (second level authentication)

At the command prompt, type:

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type
```

Example:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

Add and bind authentication policy label

At the command prompt, type:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

Example:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel label1 -policyName radpol11 -priority 1 -
gotoPriorityExpression NEXT
```

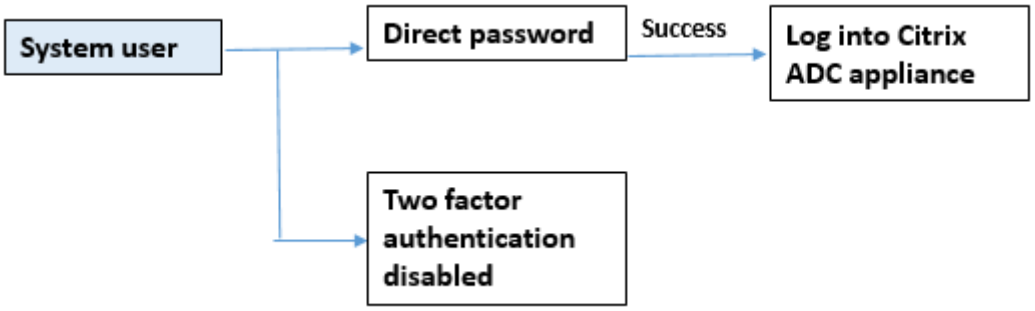

Disable local authentication in system parameter

At the command prompt, type:

```
set system parameter -localauth disabled
```

Use case 5: External authentication disabled and local authentication enabled for system user

If the user has “externalAuth” disabled, it indicates the user does not exist on the authentication server. User is not authenticated with the external authentication server even if a user with the same user name exists on the external authenticated server. User is authenticated locally.



To enable system user password and disable external authentication

At the command prompt, type the following:

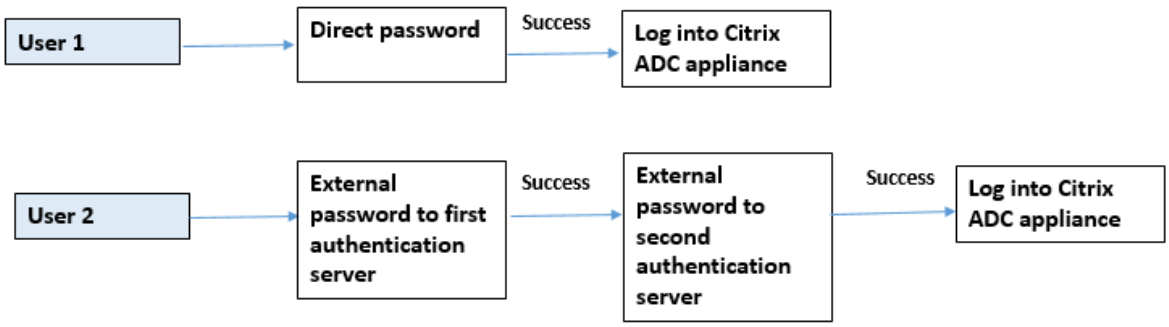
```
add system user <name> <password> -externalAuth DISABLED
```

Example:

```
add system user user1 password1 -externalAuth DISABLED
```

Use case 6: External authentication enabled and local authentication enabled for system users

To configure the appliance to authenticate system users by using a local password. If this authentication fails, the user is then authenticated by using an external authentication password on the external authentication servers at two levels.



Configure the following steps by using the CLI.

1. Add authentication action for LDAP server
2. Add authentication policy for LDAP policy
3. Add authentication action for RADIUS policy
4. Add authentication policy for RADIUS policy
5. Add authentication login schema
6. Add authentication policy label
7. Bind authentication policy label for login schema
8. Bind authentication system global for RADIUS policy
9. Bind authentication system global for LDAP policy

Add authentication action for LDAP server

At the command prompt, type:

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
ssoNameAttribute <>
```

Example:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

Add authentication policy for LDAP policy

At the command prompt, type:

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```


Example:

```
add authentication policy pol1 -rule true -action ldapact1
```

Add authentication action for RADIUS server

At the command prompt, type:

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

Example:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

Add advanced authentication policy for RADIUS server

At the command prompt, type:

```
add authentication policy <policy name> -rule true -action <rad action name
>
```

Example:

```
add authentication policy radpol11 -rule true -action radact1
```

Add authentication login schema

You can use the SingleAuth.xml login schema to display the login page and authenticate the system user at the second level authentication.

At the command prompt, type:

```
add authentication loginSchema <name> -authenticationSchema <string>
```

Example:

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

Add and bind authentication policy label to RADIUS authentication policy for user login

At the command prompt, type:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```


Example:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

Example:

```
bind authentication policylabel label1 -policyName rad pol11 -priority 1
```

Bind authentication policy global

At the command prompt, type:

```
bind system global [<policyName> [-priority <positive_integer>] [-nextFactor
 <string>] [-gotoPriorityExpression <expression>]]
```

Example:

```
bind system global radpol11 -priority 1 -nextFactor label11
```

Use case 7: External authentication enabled for selected external users only

To configure selective external users with two-factor authentication as per the search filter configured in the LDAP action while other system users are authenticated using single factor authentication.

Configure the following steps by using the CLI.

1. Add authentication action for LDAP server
2. Add authentication policy for LDAP policy
3. Add authentication action for RADIUS policy
4. Add authentication policy for RADIUS policy
5. Add authentication login schema
6. Add authentication policy label
7. Bind authentication policy label for login schema
8. Bind authentication system global for RADIUS policy

Add authentication action for LDAP server

At the command prompt, type:

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
ssoNameAttribute <>
```


Example:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

Add authentication policy for LDAP policy

At the command prompt, type:

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

Example:

```
add authentication policy pol1 -rule true -action ldapact1
```

Add authentication action for RADIUS server

At the command prompt, type:

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

Example:

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

Add advanced authentication policy for RADIUS server

At the command prompt, type:

```
add authentication policy <policy name> -rule true -action <rad action name
>
```

Example:

```
add authentication policy radpol11 -rule true -action radact1
```

Add authentication login schema

You can use the SingleAuth.xml login schema to provide the login page for the appliance to authenticate a system user at a second level of authentication.

At the command prompt, type:

```
add authentication loginSchema <name> -authenticationSchema <string>
```


Example:

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

Add and bind authentication policy label to RADIUS authentication policy for user login

At the command prompt, type:

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

Example:

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

Example:

```
bind authentication policylabel label1 -policyName radpol11 -priority
```

Bind authentication policy global

At the command prompt, type:

```
bind system global [<policyName> [-priority <positive_integer>] [-nextFactor
 <string>] [-gotoPriorityExpression <expression>]]
```

Example:

```
bind system global radpol11 -priority 1 -nextFactor label11
```

To configure without two-factor authentication for group users using the search filter:

1. Add authentication action for LDAP server
2. Add authentication policy for LDAP server
3. Bind authentication system global for LDAP server

Add authentication action for LDAP server

At the command prompt, type:

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
searchFilter<>
```


Example:

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name - searchFilter "memberOf=CN=grp4,CN=Users,DC=
aaatm-test,DC=com"
```

Add authentication policy for LDAP server

At the command prompt, type:

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

Example:

```
add authentication policy pol1 -rule true -action ldapact1
```

Bind authentication system global for LDAP policy

At the command prompt, type:

```
bind system global ldappolicy -priority <priority> -nextFactor <policy
label name>
```

Example:

```
bind system global pol11 -priority 1 -nextFactor label11
```

Display customized prompt message for two factor authentication

When you configure two factor password field with SingleAuth.xml file at /flash/nsconfig/loginschema/LoginSchema

Following is the snippet of a SingleAuth.xml file where ‘SecondPassword:’ is the second password field name which is prompted to the user to enter a second password.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <AuthenticateResponse xmlns="http://citrix.com/authentication/response
     /1">
3 <Status>success</Status>
4 <Result>more-info</Result>
5 <StateContext/>
6 <AuthenticationRequirements>
7 <PostBack>/nf/auth/doAuthentication.do</PostBack>
8 <CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
9 <CancelButtonText>Cancel</CancelButtonText>
```



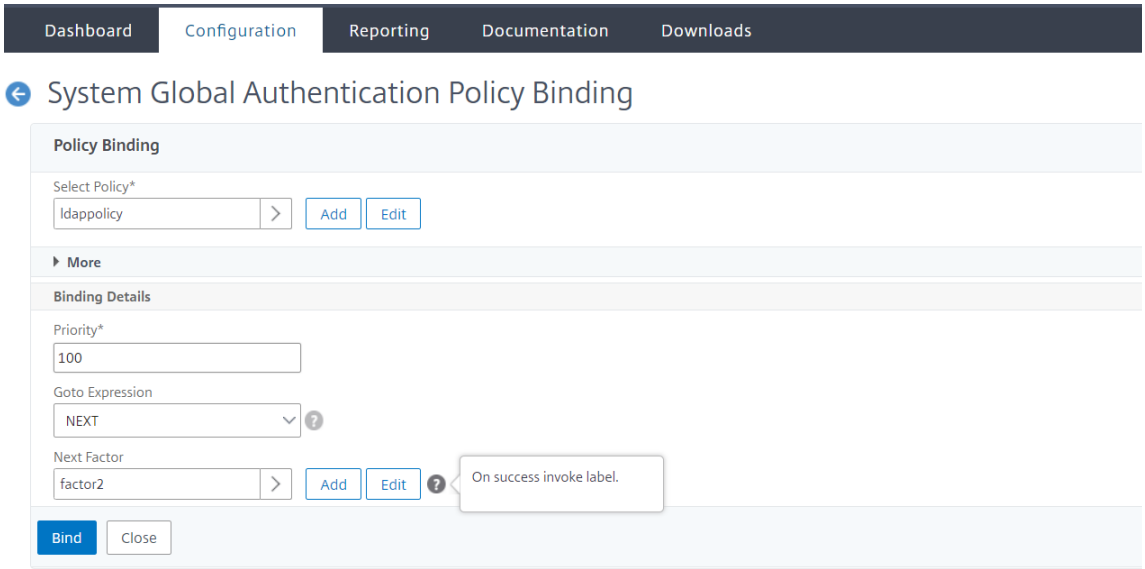
```
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</SaveID><Type>username</Type></Credential><Label><Text>singleauth_user_name</Text><Type>nsg-login-label</Type></Label><Input><AssistiveText>singleauth_please_supply_either_domain\username_or_user@fully.qualified.domain</AssistiveText><Text><Secret>false</Secret><ReadOnly>false</ReadOnly><InitialValue/><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password</SaveID><Type>password</Type></Credential><Label><Text>SecondPassword:</Text><Type>nsg-login-label</Type></Label><Input><Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue/><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>singleauth_first_factor</Text><Type>nsg_confirmation</Type></Label><Input/></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials</Type></Credential><Label><Text>singleauth_remember_my_password</Text><Type>nsg-login-label</Type></Label><Input><CheckBox><InitialValue>false</InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential><Label><Type>none</Type></Label><Input><Button>singleauth_log_on</Button></Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

Configuring two-factor authentication by using the Citrix ADC GUI

1. Log on to Citrix ADC appliance.
2. Go to **System** > **Authentication** > **Advanced Policies** > **Policy**.
3. Click Add to create the first level authentication policy.
4. In **Create Authentication Policy** page, set the following parameters.
   a) Name. Name of the policy
   b) Action Type. Select action type as LDAP, Active Directory, RADIUS, TACACS, and so on
   c) Action. The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the plus and create an action of the proper type.
   d) Expression. Provide an advanced policy expression.
5. Click **Create** and then **Close**.

a) Expression. Provide an advanced policy expression.

6. Click **Create**.
7. Click **Add** to create the second level authentication policy.
8. In the **Create Authentication Policy** page, set the follow parameters
   a) Name. Name of the policy
   b) Action Type. Select action type as LDAP, Active Directory, RADIUS, TACACS, and so on
   c) Action. The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the + icon to create an action of the proper type.
   d) Expression. Provide an advanced policy expression
9. Click **Create** and then **Close**.
   a) Expression. Provide an advanced policy expression.
10. Click **Create**.
11. In the **Authentication Policies** page, click **Global Binding**.
12. In the **Create Global Authentication Policy Binding** page, select the first level authentication policy, and click **Add Binding**.
13. In the **Policy Binding** page, select the authentication policy and set the following policy binding parameter.
    a) Next Factor. Select the second level authentication policy label.
14. Click **Bind** and **Close**.



15. Click **Done**.

16. Log on to the Citrix ADC appliance for the second level authentication. The user can now provide the second password. Only if both passwords are correct, the user is allowed to access the Citrix ADC appliance.

> **Note**
>
> The TACACS configured for a second factor authentication does not support authorization and accounting even if you enable it on the "tacacsAction" command. The second factor is used for the authentication purpose only.

Also, see Two factor authentication in Citrix ADC nFactor authentication topic.

Restricted system user authentication to Citrix ADC management interfaces

September 14, 2021

You can restrict system user access to specific Citrix ADC management interfaces such as CLI or API. The `allowedManagementInterface` parameter defines the list of permitted management interfaces. For example, if the management interface for a user or a group is set to API, all users in the group can access Citrix ADC through API and not through CLI. However, the Citrix ADC GUI is part of the API interface and users with API permission can also access the GUI interface.

> **Note:**
>
> By default, users and groups have access to all interfaces (CLI, API, and the GUI).

You can configure the parameter either at the user level or at the user group level. When you configure at the group level, the configuration is applied across all user accounts in the group. If a user is bound to multiple groups, the appliance allows access to an aggregated set of management interfaces. You can specify settings for a user in a group by configuring the parameter at user level. In this case – user level setting is configured for a group.
In certain scenarios, when the customer is using an external authentication server for managing user accounts, the server details are configured on the appliance. In this case, the administrator can create a user group in the Citrix ADC appliance and add all users (grouped in the external server) to the group. For example, all users managed in the external server are added to the API_users group and the admin can configure the group locally on the appliance.

> **Note:**
>
> The Citrix ADC appliance allows only `nsroot` administrator (superuser) to configure the parameter and does not allow any system user to change the parameter setting.

Configure user access to Citrix ADC management interfaces by using the CLI

To allow user access to a specific management interface, you must set the allowed management interface parameter. At the command prompt, type:

```
set system group <groupName> [-allowedManagementInterface ( CLI | API )]
```

Example:

```
set system group network_usergroup -allowedManagementInterface CLI
```

For parameter description, see Authentication and authorization command reference topic.

To know about Citrix GUI and CLI interfaces, see Access Citrix ADC topic.

TCP Configurations

October 4, 2021

TCP configurations for a Citrix ADC appliance can be specified in an entity called a TCP profile, which is a collection of TCP settings. The TCP profile can then be associated with services or virtual servers that want to use these TCP configurations.

A default TCP profile can be configured to set the TCP configurations that will be applied by default, globally to all services and virtual servers.

> **Note:**
>
> When a TCP parameter has different values for service, virtual server, and globally, the value of the most-specific entity (the service) is given the highest precedence. The Citrix ADC appliance also provides other approaches for configuring TCP. Read on for more information.

Supported TCP configuration

The Citrix ADC appliance supports the following TCP capabilities:

Defending TCP against spoofing attacks

The **Citrix ADC implementation of** window attenuation is RFC 4953 compliant.

Explicit Congestion Notification (ECN)

The appliance sends notification of the network congestion status to the sender of the data and takes corrective measures for data congestion or data corruption. The Citrix ADC implementation of ECN is RFC 3168 compliant.

Round trip time measurement (RTTM) using the timestamp option

For the TimeStamp option to work, at least one side of the connection (client or server) must support it. The Citrix ADC implementation of the `TimeStamp` option is RFC 1323 compliant.

Detection of spurious retransmissions

This can be done using TCP duplicate selective acknowledgment (D-SACK) and forward RTO-Recovery (F-RTO). If there are spurious retransmissions, the congestion control configurations are reverted to their original state. The Citrix ADC implementation of D-SACK is RFC 2883 compliant, and F-RTO is RFC 5682 compliant.

Congestion control

This functionality use New-Reno, BIC, CUBIC, Nile, and TCP Westwood algorithms.

Window scaling

This increases the **TCP receive** window size beyond its maximum value of 65,535 bytes.

Points to consider before you configure window scaling

- You do not set a high value for the scale factor, because this might have adverse effects on the appliance and the network.
- You do not configure window scaling unless you clearly know why you want to change the window size.
- Both hosts in the TCP connection send a window scale option during connection establishment. If only one side of a connection sets this option, window scaling is not used for the connection.
- Each connection for the same session is an independent window scaling session. For example, when a client's request and the server's response flow through the appliance, it is possible to have window scaling between the client and the appliance without window scaling between the appliance and the server.

TCP maximum congestion window

The window size is a user configurable one. The default value is 8190 bytes.

Selective acknowledgment (SACK)

This uses the data receiver (either a Citrix ADC appliance or a client) notifies the sender about all the segments that have been received successfully.

Forward acknowledgment (FACK)

This functionality avoids TCP congestion by explicitly measuring the total number of data bytes outstanding in the network, and helping the sender (either a Citrix ADC or a client) control the amount of data injected into the network during retransmission timeouts.

TCP connection multiplexing

This functionality enables reuse of existing TCP connections. The Citrix ADC appliance stores established TCP connections to the reuse pool. Whenever a client request is received, the appliance checks for an available connection in the reuse pool and serves the new client if the connection is available. If it is unavailable, the appliance creates a connection for the client request and stores the connection to the reuse pool. The Citrix ADC supports connection multiplexing for HTTP, SSL, and DataStream connection types.

Dynamic receive buffering

This allows the receive buffer to be adjusted dynamically based on memory and network conditions.

Multipath TCP Connection

Multipath TCP (MPTCP) connections between the client and the Citrix ADC appliance. MPTCP connections are not supported between the Citrix ADC appliace and the back-end server. The Citrix ADC implementation of MPTCP is RFC 6824 and RFC 8684 compliant supporting both MPTCP version 0 and 1.

You can view MPTCP statistics such as active MPTCP connections and active subflow connections by using the command line interface.

At the command prompt, type one of the following commands to display a summary or detailed summary of MPTCP statistics, or to clear the statistics display:

1. `Stat MPTCP`
2. `Stat mptcp -detail`
3. `Clearstats basic`

> **Note:**
>
> To establish an MPTCP connection, both the client and the Citrix ADC appliance must support the same MPTCP version. If you use the Citrix ADC appliance as an MPTCP gateway for your servers, the servers do not have to support MPTCP. When the client starts a new MPTCP connection, the appliance identifies the client's MPTPC version from the MP_CAPABALE option in the SYN packet. If the client's version is higher than the one supported on the appliance, the appli-

> ance indicates its highest version in the MP_CAPABALE option of the SYN-ACK packet. The client then falls back to a lower version and sends the version number in the MP_CAPABALE option of the ACK packet. If that version is supportable, the appliance continues the MPTCP connection. Otherwise, the appliance falls back to a regular TCP. The Citrix ADC appliance does not initiate subflows (MP_JOIN's). The appliance expects the client to initiate subflows.

Support for additional address advertisement (ADD_ADDR) in MPTCP

In an MPTCP deployment, if you have a virtual server bound with an IP set that has additional virtual server IP addresses, then the additional address advertisement (ADD_ADDR) functionality advertises the IP address of the virtual servers bound to the IP set. Clients can initiate more `MP-JOIN` sub flows to the advertised IP addresses.

Points to remember about MPTCP ADD_ADDR functionality

- You can send a maximum of 10 IP addresses as part of the `ADD_ADDR` option. If there are more than 10 IP addresses with the `mptcpAdvertise` parameter enabled, after advertising the 10 IP address, the appliance ignore the rest of the IP addresses.
- If the MP-CAPABLE subflow is made to one of the IP addresses in the IP set instead of the primary virtual server IP address, then the virtual server IP address is advertised if the `mptcpAdvertise` parameter is enabled for the virtual server IP address

Configure more address advertisement (ADD_ADDR) feature to advertise more VIP address by using the CLI

You can configure the `MPTCP ADD_ADDR` functionality for both IPv4 and IPv6 address types. In general, multiple IPv4 and IPv6 IPs can be attached to a single IP set and the parameter can be enabled on any subset of IP addresses. In the ADD_ADDR feature, only the IP addresses that have the "mptcpAdvertise" option enabled is advertised and the remaining IP addresses from the IP set is ignored. Complete the following steps to configure the `ADD_ADDR` feature:

1. Add an IP set.
2. Add an IP address of type virtual server IP (VIP) with MPTCP advertise enabled.
3. Bind the IP address with the IP set.
4. Configure IP set with the load balancing virtual server.

Add an IP set

At the command prompt, type:

```
add ipset <name> [-td <positive_integer>]
<!--NeedCopy-->
```


Example:

```
add ipset ipset_1
<!--NeedCopy-->
```

Add an IP address of type virtual server IP (VIP) with MPTCP advertise enabled

At the command type:

```
add ns ip <IPAddress>@ <netmask> [-mptcpAdvertise ( YES | NO )] -type <
   type>
<!--NeedCopy-->
```

Example:

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
```

Bind IP addresses to the IP set

At the command prompt, type:

```
bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

Example:

```
bind ipset ipset_1 10.10.10.10
```

Configure IP set to load balancing virtual server

At the command prompt, type:

```
set lb vserver <name> [-ipset  <string>]
<!--NeedCopy-->
```

Example:

```
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

Sample Configuration:

```
Add ipset ipset_1
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
bind ipset ipset_1 10.10.10.10
```



```
4 set lb vserver lb1 -ipset ipset_1
5 <!--NeedCopy-->
```

Configure advertising external IP address using ADD_ADDR functionality

If the advertised IP address is owned by the external entity and the Citrix ADC appliance needs to advertise the IP address, the "MPTCPAdvertise" parameter must be enabled with state and ARP parameters disabled.

Complete the following steps to configure ADD_ADDR for advertising the external IP address.

1. Add an IP address of type virtual server IP (VIP) with MPTCP advertise enabled.
2. Bind the IP address with the IP set.
3. Bind IP set with the load balancing virtual server

Add external IP address of type virtual server IP (VIP) with MPTCP advertise enabled

At the command prompt, type:

```
1 add ns ip <IPAddress>@ <External-IP-mask -type VIP> [-mptcpAdvertise (
     YES | NO )] -type <type> -state DISABLED -arp DISABLED
2 <!--NeedCopy-->
```

Example:

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP -state
DISABLED -arp DISABLED
```

Bind IP addresses to the IP set

At the command prompt, type:

```
1 bind ipset <name> <IPAddress>
2 <!--NeedCopy-->
```

Example:

```
bind ipset ipset_1 10.10.10.10
```

Configure IP set to load balancing virtual server

At the command prompt, type:

```
1 set lb vserver <name> [-ipset  <string>]
2 <!--NeedCopy-->
```


Example:

```
set lb vserver lb1 -ipset ipset_1
```

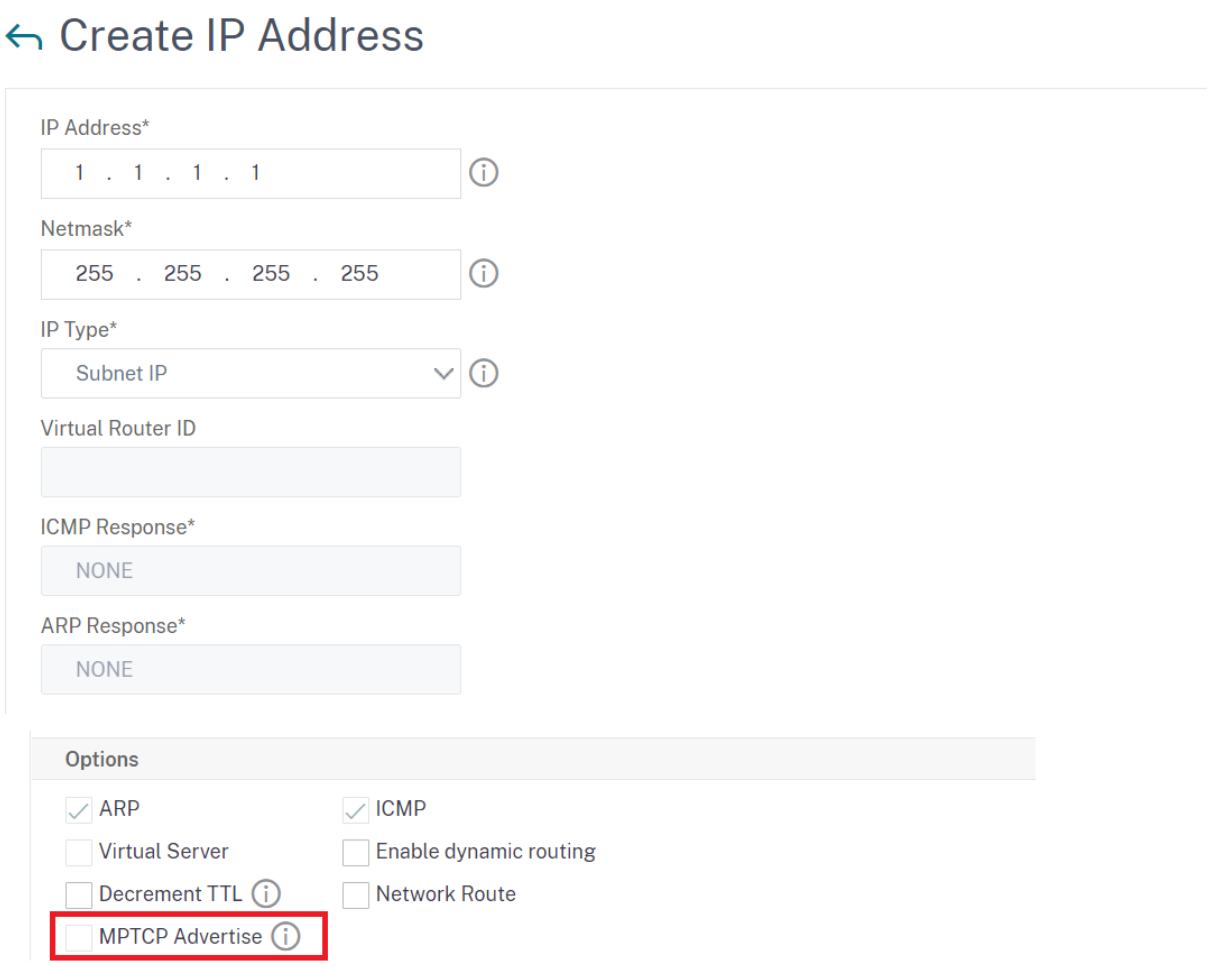
Sample Configuration:

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
   state DISABLED -arp DISABLED
bind ipset ipset_1 10.10.10.10
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

Advertise IP address to MPTCP enabled clients by using the Citrix ADC GUI

Complete the following step to advertise the IP address to the MPTCP enabled clients:

1. Navigate to **System > Network > IPs**.
2. In the details pane, click **Add**.
3. In the **Create IP Address** page, select the **MPTCP Advertise** check box to set the parameter. By default, it is disabled.



Extracting the TCP/IP path overlay option and inserting the client-IP HTTP header

Extracting TCP/IP path overlay and inserting client-IP HTTP header. Data transport through overlay networks often uses connection termination or Network Address Translation (NAT), in which the IP address of the source client is lost. To avoid this, the Citrix ADC appliance extracts the TCP/IP path overlay option and inserts the source client's IP address into the HTTP header. With the IP address in the header, the web server can identify the source client that made the connection. The extracted data is valid for a lifetime of the TCP connection and therefore, this prevents the next hop host from having to interpret the option again. This option is applicable only for web services that have the client-IP insertion option enabled.

TCP segmentation offload

Offloads TCP segmentation to the NIC. If you set the option as "AUTOMATIC", TCP segmentation is offloaded to the NIC, if NIC is supported.

Synchronizing cookie for TCP handshake with clients

This is used for resisting SYN flood attacks. You can enable or disable the `SYNCOOKIE` mechanism for TCP handshake with clients. Disabling `SYNCOOKIE` prevents `SYN` attack protection on the Citrix ADC appliance.

Learning MSS to enable MSS learning for all the virtual servers configured on the appliance

Supportable TCP Parameters

The following table provides a list of TCP parameters and its default value configured on a Citrix ADC appliance.

Parameter	Default Value	Description
Window Management		
TCP Delayed-ACK Timer	100 millisec	Timeout for TCP delayed ACK, in milliseconds.
TCP minimum Retransmission Timeout(RTO) in milli sec	1000 milli sec	Minimum retransmission timeout, in milliseconds, specified in 10-millisecond increments (value must yield a whole number if divided by 10)
Connection idle time before starting keep-alive probes	900 seconds	Silently drop TCP established connections on idle timeouts established connections on idle timeout
TCP Timestamp Option	DISABLED	The timestamp option allows for accurate RTT measurement. Enable or Disable TCP Timestamp option.
Multipath TCP session timeout	0 seconds	MPTCP session timeout in seconds. If this value is not set, idle. MPTCP sessions are flushed after the virtual server's client idle timeout.
Silently Drop HalfClosed connections on idle timeout	0 seconds	Silently drop TCP half closed connections on idle timeout.

Parameter	Default Value	Description
Silently Drop Established connections on idle timeout	DISABLED	Silently drop TCP established connections on idle timeout
Memory Management		
TCP Buffer Size	131072 bytes	TCP buffer size is the receive buffer size on the Citrix ADC. This buffer size is advertised to clients and servers from Citrix ADC and it controls their ability to send data to Citrix ADC. The default buffer size is 8K and usually it is safe to increment this when talking to internal server farms. The buffer size is also impact by the actual application layer in Citrix ADC like for SSL endpoint cases it is set to 40 K and for Compression it is set to 96 K. **Note:** The buffer size argument must be set for dynamic adjustments to take place.
TCP Send Buffer Size	8190 bytes	TCP Send Buffer Size
TCP Dynamic Receive Buffering	DISABLED	Enable or disable dynamic receive buffering. When enabled, it allows the receive buffer to be adjusted dynamically based on memory and network conditions. **Note:** The buffer size argument must be set for dynamic adjustments to take place
TCP Max congestion window(CWND)	524288 bytes	TCP Maximum Congestion Window

Parameter	Default Value	Description
Window Scaling status	ENALBED	Enable or disable window scaling.
Window Scaling factor	8	Factor used to calculate the new window size. This argument is needed only when window scaling is enabled.
Connection Setup		
Keep-alive probes	DISABLED	Send periodic TCP keep-alive (KA) probes to check if peer is still up.
Connection idle time before starting keep-alive probes	900 seconds	Duration, in seconds, for the connection to be idle, before sending a keep-alive (KA) probe.
Keep-alive probe interval	75 seconds	Time interval, in seconds, before the next keep-alive (KA) probe, if the peer does not respond.
Maximum keep-alive probes to be missed before dropping connection.	3	Number of keep-alive (KA) probes to be sent when not acknowledged, before assuming the peer to be down.
RST window attenuation (spoof protection).	DISABLED	Enable or disable RST window attenuation to protect against spoofing. When enabled, the reply is with corrective ACK when a sequence number is invalid.
Accept RST with last acknowledged sequence number.	ENABLED	
Data transfer		

Parameter	Default Value	Description
Immediate ACK on PUSH packet	ENABLED	Send immediate positive acknowledgment (ACK) on receipt of TCP packets with PUSH flag.
Maximum packets per MSS	0	Maximum number of octets to allow in a TCP data segment
Nagle's Algorithm	DISABLED	Nagle's Algorithm fights with the problem of small packets in TCP transmission. Applications like Telnet and other real time engines which require every key stroke to be passed to the other side often create small packets. With Nagle's algorithm Citrix ADC can buffer such small packets and sends them together to increase on the connection efficiency. This algorithm needs to work along with other TCP optimization techniques in the Citrix ADC.
Maximum TCP segments allowed in a burst	10 MSS	Maximum number of TCP segments allowed in a burst
Maximum out-of-order packets to queue	300	Maximum size of out-of-order packets queue. A value of 0 means no limit
Congestion Control		
TCP Flavor	CUBIC	
Initial congestion window(cwnd) setting	4 MSS	Initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server

Parameter	Default Value	Description
TCP Explicit Congestion Notification(ECN)	DISABLED	Explicit Congestion Notification (ECN) provides end to end notification of network congestion without dropping packets.
TCP Max congestion window(CWND)	524288 bytes	TCP maintains a congestion window (CWND), limiting the total number of unacknowledged packets that may be in transit end-to-end. In TCP, the congestion window is one of the factors that determines the number of bytes that can be outstanding at any time. The congestion window is a means of stopping a link between the sender and the receiver from becoming overloaded with too much traffic. It is calculated by estimating how much congestion there is on the link.
TCP Hybrid Start (HyStart)	8 bytes	
TCP minimum Retransmission Timeout(RTO) in milli sec	1000	Minimum retransmission timeout, in milliseconds, specified in 10-millisecond increments (value must yield a whole number if divided by 10).
TCP dupack threshold	DISABLED	
Burst Rate Control	3	TCP Burst Rate Control DISABLED/FIXED/DYNAMIC. FIXED requires a TCP rate to be set

Parameter	Default Value	Description
TCP Rate	DISABLED	TCP connection payload send rate in Kb/s
TCP Rate Maximum Queue	0	Maximum connection queue size in bytes, when BurstRateControl is used.
MPTCP		
Multipath TCP	DISABLED	Multipath TCP (MPTCP) is a set of extensions to regular TCP to provide a Multipath TCP service, which enables a transport connection to operate across multiple paths simultaneously.
Multipath TCP drop data on pre-established subflow	DISABLED	Enable or disable silently dropping the data on Pre-Established subflow. When enabled, DSS data packets are dropped silently instead of dropping the connection when data is received on pre established subflow.
Multipath TCP fastopen	DISABLED	Enable or disable Multipath TCP fastopen. When enabled, DSS data packets are accepted before receiving the third ack of SYN handshake.
Multipath TCP session timeout	0 seconds	MPTCP session timeout in seconds. If this value is not set, idle MPTCP sessions are flushed after the virtual server's client idle timeout.
Security		

Parameter	Default Value	Description
SYN spoof protection	DISABLED	Enable or disable drop of invalid SYN packets to protect against spoofing. When disabled, established connections are reset when a SYN packet is received.
TCP Syncookie	DISABLED	This is used for resisting SYN flood attacks. Enable or disable the SYNCOOKIE mechanism for TCP handshake with clients. Disabling SYNCOOKIE prevents SYN attack protection on the Citrix ADC appliance.
Loss Detection and Recovery		
Duplicate Selective Acknowledgment (DSACK)	ENABLED	A Citrix ADC appliance uses Duplicate Selective Acknowledgment (DSACK) to determine if a retransmission was sent in error.

Parameter	Default Value	Description
Forward RTO recovery (FRTO)	ENABLED	Detects spurious TCP retransmission timeouts. After retransmitting the first unacknowledged segment triggered by a timeout, the algorithm of the TCP sender monitors the incoming acknowledgments to determine whether the timeout was spurious. It then decides whether to send new segments or retransmit unacknowledged segments. The algorithm effectively helps to avoid another unnecessary retransmissions and thereby improves TCP performance in the case of a spurious timeout.
TCP Forward Acknowledgment (FACK)	ENABLED	Enable or disable FACK (Forward ACK).
Selective Acknowledgement(SACK) status	ENABLED	TCP SACK addresses the problem of multiple packet losses which reduces the overall throughput capacity. With selective acknowledgment the receiver can inform the sender about all the segments which are received successfully, enabling the sender to only retransmit the segments which were lost. This technique helps Citrix ADC improve overall throughput and reduce the connection latency.

Parameter	Default Value	Description
Maximum packets per retransmission	1	Allows Citrix ADC to control how many packets to be retransmitted in one attempt. When Citrix ADC receives a partial ACK and it has to do retransmission then this setting is considered. This does not impact the RTO based retransmissions.
TCP Delayed-ACK Timer	100 millisec	Timeout for TCP delayed ACK, in milliseconds
TCO Optimization		
TCP Optimization mode	TRANSPARENT	TCP Optimization modes TRANSPARENT/ENDPOINT
Apply adaptive TCP optimizations	DISABLED	Apply Adaptive TCP optimizations
TCP Segmentation Offload	AUTOMATIC	Offload TCP segmentation to the NIC. If set to AUTOMATIC, TCP segmentation is offloaded to the NIC, if the NIC supports it.
ACK Aggregation	DISABLED	Enable or disable ACK Aggregation
TCP Time-wait(or Time_wait)	40 secs	Time to elapse before releasing a closed TCP connection
Delink client and server on RST	DISABLED	Delink client and server connection, when there is outstanding data to be sent to the other side.

Setting Global TCP Parameters

The Citrix ADC appliance allows you to specify values for TCP parameters that are applicable to all Citrix ADC services and virtual servers. This can be done using:

- Default TCP profile
- Global TCP command
- TCP buffering feature

> **Note**:
>
> The `recvBuffSize` parameter of the set ns tcpParam command is deprecated from release 9.2 onwards. In later releases, set the buffer size by using the `bufferSize` parameter of the set ns tcpProfile command. If you upgrade to a release where the `recvBuffSize` parameter is deprecated, the `bufferSize` parameter is set to its default value.

Default TCP profile

A TCP profile, named as `nstcp_default_profile`, is used to specify TCP configurations that is used if no TCP configurations are provided at the service or virtual server level.

> **Notes:**
>
> - Not all TCP parameters can be configured through the default TCP profile. Some settings have to be performed by using the global TCP command (see section below).
> - The default profile does not have to be explicitly bound to a service or virtual server.

To configure the default TCP profile

- Using the command line interface, at the command prompt enter:

```
1 set ns tcpProfile nstcp_default_profile...
2 <!--NeedCopy-->
```

- On the GUI, navigate to **System** > **Profiles**, click **TCP Profiles** and update nstcp_default_profile.

Global TCP command

Another approach you can use to configure global TCP parameters is the global TCP command. In addition to some unique parameters, this command duplicates some parameters that can be set by using a TCP profile. Any update made to these duplicate parameters is reflected in the corresponding parameter in the default TCP profile.

For example, if the SACK parameter is updated using this approach, the value is reflected in the SACK parameter of the default TCP profile (nstcp_default_profile).

> **Note:**
>
> Citrix recommends that you use this approach only for TCP parameters that are not available in the default TCP profile.

To configure the global TCP command

- Using the command line interface, at the command prompt enter:

```
set ns tcpParam …
<!--NeedCopy-->
```

- On the GUI, navigate to **System > Settings**. Click **Change TCP parameters** and, update the required TCP parameters.

TCP buffering feature

Citrix ADC provides a feature called TCP buffering that you can use to specify the TCP buffer size. The feature can be enabled globally or at service level.

> **Note:**
>
> The buffer size can also be configured in the default TCP profile. If the buffer size has different values in the TCP buffering feature and the default TCP profile, the greater value is applied.

To configure the TCP buffering feature globally

- At the command prompt enter:

  enable ns mode TCPB

  ```
  set ns tcpbufParam -size <positiveInteger> -memLimit <positiveInteger>
  ```

- On the GUI, navigate to **System** > **Settings**, click **Configure Modes** and, select **TCP Buffering**.

  And, navigate to **System** > **Settings**, click **Change TCP parameters**, specify values for **Buffer size** and **Memory usage limit**.

Setting Service or Virtual Server Specific TCP Parameters

Using TCP profiles, you can specify TCP parameters for services and virtual servers. You must define a TCP profile (or use a built-in TCP profile) and associate the profile with the appropriate service and virtual server.

> **Note:**
>
> You can also modify the TCP parameters of default profiles as per your requirements.

You can specify the TCP buffer size at service level using the parameters specified by the TCP buffering feature.

To specify service or virtual server level TCP configurations by using the command line interface

At the command prompt, perform the following:

1. Configure the TCP profile.

```
set ns tcpProfile <profile-name>...
<!--NeedCopy-->
```

2. Bind the TCP profile to the service or virtual server.

```
    set service <name> ....
    <!--NeedCopy-->
```

Example:

```
> set service service1 -tcpProfileName profile1
```

To bind the TCP profile to the virtual server:

```
set lb vserver <name> ....
<!--NeedCopy-->
```

Example:

```
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

To specify service or virtual server level TCP configurations by using the GUI

At the GUI, perform the following:

1. Configure the TCP profile.

   Navigate to **System** > **Profiles** > **TCP Profiles**, and create the TCP profile.

2. Bind the TCP profile to the service or virtual server.

   Navigate to **Traffic Management** > **Load Balancing** > **Services/Virtual Servers**, and create the TCP profile, which should be bound to the service or virtual server.

Built-in TCP Profiles

For convenience of configuration, the Citrix ADC provides some built-in TCP profiles. Review the built-in profiles listed for the following and select a profile and use it as it is or modify it to meet your requirements. You can bind these profiles to your required services or virtual servers.

Built-in profile	Description
nstcp_default_profile	Represents the default global TCP settings on the appliance.
nstcp_default_tcp_lan	Useful for back-end server connections, where these servers reside on the same LAN as the appliance.
nstcp_default_WAN	useful for WAN deployments.
nstcp_default_tcp_lan_thin_stream	Similar to the nstcp_default_tcp_lan profile. However, the settings are tuned for small size packet flows.
nstcp_default_tcp_interactive_stream	Similar to the nstcp_default_tcp_lan profile. However, it has a reduced delayed ACK timer and ACK on **PUSH packet** settings.
nstcp_default_tcp_lfp	Useful for long fat pipe networks (WAN) on the client side. Long fat pipe networks have long delay, high bandwidth lines with minimal packet drops.
nstcp_default_tcp_lfp_thin_stream	Similar to the nstcp_default_tcp_lfp profile. However, the settings are tuned for small size packet flows.
nstcp_default_tcp_lnp	Useful for long narrow pipe networks (WAN) on the client side. Long narrow pipe networks have considerable packet loss occasionally.
nstcp_default_tcp_lnp_thin_stream	Similar to the nstcp_default_tcp_lnp profile. However, the settings are tuned for small size packet flows.
nstcp_internal_apps	Useful for internal applications on the appliance (for example, GSLB site syncing). This contains tuned window scaling and SACK options for the desired applications. This profile should not be bound to applications other than internal applications.
nstcp_default_Mobile_profile	Useful for mobile devices.
nstcp_default_XA_XD_profile	Useful for a XenApp or XenDesktop deployment.

Sample TCP Configurations

Sample command line interface examples for configuring the following:

Defending TCP against spoofing attacks

Enable the Citrix ADC to defend TCP against spoof attacks. By default the “rstWindowAttenuation” parameter is disabled. This parameter is enabled to protect the appliance against spoofing. If you enable, it replies with corrective acknowledgment (ACK) for an invalid sequence number. Possible values are Enabled, Disabled.

Where, the RST window attenuate parameter protects the appliance against spoofing. When enabled, reply with corrective ACK when a sequence number is invalid.

```
1      > set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -
          spoofSynDrop ENABLED
2      Done
3      > set lb vserver lbvserver1 -tcpProfileName profile1
4      Done
5  <!--NeedCopy-->
```

Explicit Congestion Notification (ECN)

Enable ECN on the required TCP profile

```
1      > set ns tcpProfile profile1 -ECN ENABLED
2      Done
3      > set lb vserver lbvserver1 -tcpProfileName profile1
4      Done
5  <!--NeedCopy-->
```

Selective Acknowledgment (SACK)

Enable SACK on the required TCP profile.

```
1      > set ns tcpProfile profile1 -SACK ENABLED
2      Done
3      > set lb vserver lbvserver1 -tcpProfileName profile1
4      Done
5  <!--NeedCopy-->
```


Forward Acknowledgment (FACK)

Enable FACK on the required TCP profile.

```
> set ns tcpProfile profile1 -FACK ENABLED
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

Window Scaling (WS)

Enable window scaling and set the window scaling factor on the required TCP profile.

```
set ns tcpProfile profile1  - WS ENABLED  - WSVal 9
Done
set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```

Maximum Segment Size (MSS)

Update the MSS related configurations.

```
> set ns tcpProfile profile1  - mss 1460 - maxPktPerMss 512
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```

Citrix ADC to learn the MSS of a virtual server

Enable the Citrix ADC to learn the VSS and update other related configurations.

```
> set ns tcpParam -learnVsvrMSS ENABLED  - mssLearnInterval 180 -
    mssLearnDelay 3600
Done
<!--NeedCopy-->
```

TCP keep-alive

Enable TCP keep-alive and update other related configurations.

```
> set ns tcpProfile profile1 -KA ENABLED -KaprobeUpdateLastactivity ENABLED
-KAconnIdleTime 900 -KAmaxProbes 3 -KaprobeInterval 75
```



```
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

Buffer size - using TCP profile

Specify the buffer size.

```
> set ns tcpProfile profile1 -bufferSize 8190
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

Buffer size - using TCP buffering feature

Enable the TCP buffering feature (globally or for a service) and then specify the buffer size and the memory limit.

```
> enable ns feature TCPB
Done
> set ns tcpbufParam -size 64 -memLimit 64
Done
```

MPTCP

Enable MPTCP and then set the optional MPTCP configurations.

```
> set ns tcpProfile profile1 -mptcp ENABLED
Done
> set ns tcpProfile profile1 -mptcpDropDataOnPreEstSF ENABLED -mptcpFastOpen
 ENABLED -mptcpSessionTimeout 7200
Done
> set ns tcpparam -mptcpConCloseOnPassiveSF ENABLED -mptcpChecksum ENABLED
-mptcpSFtimeout 0 -mptcpSFReplaceTimeout 10
-mptcpMaxSF 4 -mptcpMaxPendingSF 4 -mptcpPendingJoinThreshold 0 -mptcpRTOsToSwitchSF
 2 -mptcpUseBackupOnDSS ENABLED
Done
```

Congestion control

Set the required TCP congestion control algorithm.


```
set ns tcpProfile profile1 -flavor Westwood
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

Dynamic receive buffering

Enable dynamic receive buffering on the required TCP profile.

```
> set ns tcpProfile profile1 -dynamicReceiveBuffering ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

Support for TCP Fast Open (TFO) in Multipath TCP (MPTCP)

A Citrix ADC appliance now supports the TCP Fast Open (TFO) mechanism for establishing Multipath TCP (MPTCP) connections and speed up data transfers. The mechanism allows subflow data to be carried during the initial MPTCP connection handshake in SYN and SYN-ACK packets and also enables data to be consumed by the receiving node during the MPTCP connection establishment.

For more information, see TCP Fast Open topic.

Support for Variable TFO Cookie Size for MPTCP

A Citrix ADC appliance now enables you to configure a variable length TCP Fast Open (TFO) cookie of a minimum size of 4 bytes and a maximum size of 16 bytes in a TCP profile. By doing this, the appliance can respond with the configured TFO cookie size in the SYN-ACK packet to the client.

To configure the TCP Fast Open (TFO) cookie in a TCP profile by using the command line interface

At the command prompt, type:

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize <positive_integer
>
```

Example

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize 8
```

To configure the TCP Fast Open (TFO) cookie in a TCP profile by using the GUI

1. Navigate to **Configuration** > **System** > Profiles.
2. In the details pane, go to **TCP Profiles** tab and select a TCP profile.
3. In the **Configure TCP Profile** page, set the **TCP Fast Open** cookie size.
4. Click **OK** and **Done**.

SYN-Cookie timeout interval

The TCPSyncookie parameter is enabled by default in TCP profiles to provide robust (RFC 4987) based protection against SYN Attacks. If you need to accommodate custom TCP clients that are not compatible with this protection but still want to ensure a fallback in case of attack, the synAttackDetection handles this for you by automatically activating the SYNCookie behavior internally for time determined by the autosyncookietimeout parameter..

To configure the maximum SYN ACK retransmission threshold by using the command line interface:

At the command prompt, type:

```
    set ns tcpparam [-maxSynAckRetx <positive_integer>]

    Set ns tcpparam [-maxSynAckRetx 150]
<!--NeedCopy-->
```

To configure auto SYN cookie timeout interval by using the command line interface

At the command prompt, type:

```
set ns tcpparam [-autosyncookietimeout <positive_integer>]
```

```
Set ns tcpparam [-autosyncookietimeout 90]
```

Delink client and server connection

When enabled, the parameter delinks client and server connection when there is outstanding data to be sent to the other side. By default, the parameter is disabled.

```
set ns tcpparam -delinkClientServerOnRST ENABLED
Done

<!--NeedCopy-->
```

HTTP configurations

September 14, 2021

> **Important:**
>
> Starting from Citrix ADC release 13.0 build 71.x, a Citrix ADC appliance can handle large header

> size HTTP requests to accommodate the L7 application requests. The header size can be configurable up to 128 KB.

HTTP configurations for a Citrix ADC appliance can be specified in an entity called an HTTP profile, which is a collection of HTTP settings. The HTTP profile can then be associated with services or virtual servers that want to use these HTTP configurations.

A default HTTP profile can be configured to set the HTTP configurations that is applied by default, globally to all services and virtual servers.

> **Note:**
>
> When an HTTP parameter has different values for service, virtual server, and globally, the value of the most-specific entity (the service) is given the highest precedence.

The Citrix ADC appliance also provides other approaches for configuring HTTP. Read on for more information.

The Citrix ADC supports a WebSocket protocol which allows browsers and other clients to create a bidirectional, full duplex TCP connection to the servers. The Citrix ADC implementation of WebSocket is RFC 6455 compliant.

> **Note:**
>
> A Citrix ADC appliance now supports the User Source IP (USIP) address configuration for both HTTP/1.1 and HTTP/2 protocols.

Setting global HTTP parameters

The Citrix ADC appliance allows you to specify values for HTTP parameters that are applicable to all Citrix ADC services and virtual servers. This can be done using:

- Default HTTP profile
- Global HTTP command

Default HTTP profile

An HTTP profile, named as nshttp_default_profile, is used to specify HTTP configurations that is used if no HTTP configurations are provided at the service or virtual server level.

> **Notes:**
>
> - Not all HTTP parameters can be configured through the default HTTP profile. Some settings are performed by using the global HTTP command (see the following section).
> - The default profile does not have to be explicitly bound to a service or virtual server.

To configure the default HTTP profile

- Using the command line interface, at the command prompt enter:

  ```
  set ns httpProfile nshttp_default_profile …
  ```

- On the GUI, navigate to **System** > **Profiles**, click **HTTP Profiles** and update nshttp_default_profile.

Global HTTP command

Another approach you can use to configure global HTTP parameters is the global HTTP command. In addition to some unique parameters, this command duplicates some parameters that can be set by using an HTTP profile. Any update made to these duplicate parameters is reflected in the corresponding parameter in the default HTTP profile.

For example, if the maxReusePool parameter is updated using this approach, the value is reflected in the maxReusePool parameter of the default HTTP profile (nshttp_default_profile).

> **Note:**
>
> Citrix recommends that you use this approach only for HTTP parameters that are not available in the default HTTP profile.

To configure the global HTTP command

- Using the command line interface, at the command prompt enter:

  ```
  set ns httpParam …
  ```

- On the GUI, navigate to **System > Settings**, click **Change HTTP parameters** and update the required HTTP parameters.

To configure an ignore Coding scheme for connect request

To enable HTTP/2 and set HTTP/2 parameters to ignore the Coding scheme in the connect request, at the command prompt, type:

```
set ns httpParam [-ignoreConnectCodingScheme ( ENABLED | DISABLED )]
```

Example:

```
set ns httpParam -ignoreConnectCodingScheme ENABLED
```

To bind the HTTP profile to a virtual server by using the Citrix ADC command line

Configure HTTP profile to drop TRACE or TRACK invalid requests

You can enable the markTraceReqInval parameter to mark TRACK and TRACK requests as invalid. When you enable this option along with the dropInvalidReqs option on the virtual IP address, you can reset a client sending TRACE or TRACK requests to a Citrix ADC appliance.

To configure the HTTP profile using the CLI

At the command prompt, type:

```
set ns httpProfile <profile name> [-markTraceReqInval ENABLED | DISABLED ]
```

Example:

```
set ns httpProfile profile1 -markTraceReqInval ENABLED
```

Configure HTTP profile for a service group

At the command prompt, type:

```
1 add serviceGroup <serviceGroupName>@ <serviceType> [-cacheType <
     cacheType>] [-td <positive_integer>] [-maxClient <positive_integer>]
      [-maxReq <positive_integer>] [-cacheable ( YES | NO )] [-cip (
     ENABLED | DISABLED ) [<cipHeader>]] [-usip ( YES | NO )] [-
     pathMonitor ( YES | NO )] [-pathMonitorIndv ( YES | NO )] [-
     useproxyport ( YES | NO )] [-healthMonitor ( YES | NO )] [-sp ( ON |
      OFF )] [-rtspSessionidRemap ( ON | OFF )] [-cltTimeout <secs>] [-
     svrTimeout <secs>] [-CKA ( YES | NO )] [-TCPB ( YES | NO )] [-CMP (
     YES | NO )] [-maxBandwidth
2 <positive_integer>] [-monThreshold <positive_integer>] [-state ENABLED
     DISABLED )][-downStateFlush ( ENABLED | DISABLED )] [-tcpProfileName
      <string>] [-httpProfileName <string>] [-comment <string>] [-
     appflowLog ( ENABLED | DISABLED )] [-netProfile <string>] [-
     autoScale <autoScale> -memberPort <port> [-autoDisablegraceful ( YES
      | NO )] [-autoDisabledelay <secs>] ] [-monConnectionClose ( RESET |
      FIN )]
3
4 <!--NeedCopy-->
```

Example:

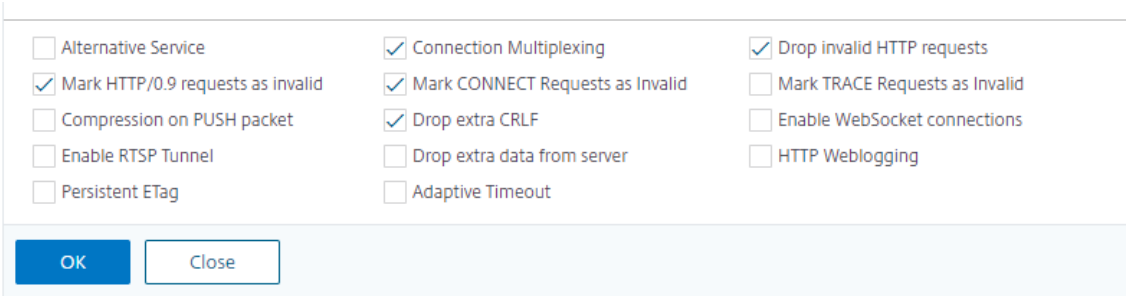
```
add serviceGroup rl-lips-30016 HTTP -maxClient 0 -maxReq 0 -cip ENABLED X-
Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360 -
CKA NO -TCPB NO -CMP NO -tcpProfileName live-tcp-profile-sg -httpProfileName
 profile1
```

Configure the HTTP profile using the Citrix ADC GUI

To mark TRACE or TRACK invalid requests, complete the following procedure.

1. Sign into Citrix ADC appliance and navigate to **Configuration** > **System** > **Profiles**.
2. In the **HTTP Profiles** tab page, click **Add**.
3. In the **Create HTTP Profile** page, select **Mark TRACE Requests as Invalid** option.

4. Click **Create**.



Setting service or virtual server specific HTTP parameters

Using HTTP profiles, you can specify HTTP parameters for services and virtual servers. You have to define an HTTP profile (or use a built-in HTTP profile) and associate the profile with the appropriate service and virtual server.

> **Note:**
>
> You can also modify the HTTP parameters of default profiles as per your requirements.

To specify service or virtual server level HTTP configurations by using the command line interface

At the command prompt, perform the following:

1. Configure the HTTP profile.

   ```
   set ns httpProfile <profile-name>...
   ```

2. Bind the HTTP profile to the service or virtual server.

   To bind the HTTP profile to the service:

```
set service <name> .....
```

Example:

```
1 > set service service1 -httpProfileName profile1
2 <!--NeedCopy-->
```

To bind the HTTP profile to the virtual server:

```
set lb vserver <name> .....
```

Example:

```
1 > set lb vserver lbvserver1 -httpProfileName profile1
2 <!--NeedCopy-->
```


To specify service or virtual server level HTTP configurations by using the GUI

At the GUI, perform the following:

1. Configure the HTTP profile.

   Navigate to **System** > **Profiles** > **HTTP Profiles**, and create the HTTP profile.

2. Bind the HTTP profile to the service or virtual server.

   Navigate to **Traffic Management** > **Load Balancing** > **Services/Virtual Servers**, and create the HTTP profile, which must be bound to the service/virtual server.

Built-in HTTP profiles

For convenience of configuration, the Citrix ADC provides some built-in HTTP profiles. Review the profiles listed and use it as it is or modify it to meet your requirements. You can bind these profiles to the required services or virtual servers.

Built-in profile	Description
nshttp_default_profile	Represents the default global HTTP settings on the appliance.
nshttp_default_strict_validation	Settings for deployments that require strict validation of HTTP requests and responses.

Sample HTTP configurations

Sample command line interface examples to configure the following:

- HTTP band statistics
- WebSocket connections

HTTP band statistics

Specify the band size for HTTP requests and responses.

```
1 > set protocol httpBand reqBandSize 300 respBandSize 2048
2 Done
3 > show protocol httpband -type REQUEST
4 <!--NeedCopy-->
```


WebSocket connections

Enable WebSocket on the required HTTP profile.

```
> set ns httpProfile http_profile1 -webSocket ENABLED
Done
> set lb vserver lbvserver1 -httpProfileName profile1
Done
<!--NeedCopy-->
```

HTTP/2 configuration

September 14, 2021

Note: The HTTP/2 functionality is supported on the Citrix ADC MPX, VPX, and SDX models. In a Citrix ADC VPX appliance, the HTTP/2 functionality is supported from 11.0 release onwards.

The problem with web application performance is directly related to the trend toward increasing the page size and the number of objects on the webpages. HTTP/1.1 was developed to support smaller webpages, slower Internet connections, and more limited server hardware than are common today. It is not suitable for new technologies such as JavaScript and cascading style sheets (CSS) or new media types such as Flash videos and graphics-rich images. This is because it can request only one resource per connection to the server. The limitation significantly increases the number of round trips, causing longer page-rendering and reduced network performance.

The HTTP/2 protocol addresses these limitations by allowing communication to occur with less data transmitted over the network, and providing the ability to send multiple requests and responses across a single connection. At its core, HTTP/2 addresses the key limitations of HTTP/1.1 by using the underlying network connections more efficiently. It changes the way requests and responses travel over the network.

HTTP/2 is a binary protocol. It is more efficient to parse, more compact on the wire, and most importantly, it is less error-prone, compared to textual protocols like HTTP/1.1. The HTTP/2 protocol uses a binary framing layer that defines the frame type and how HTTP messages are encapsulated and transferred between the client and server. The HTTP/2 functionality supports the use of the CONNECT method to establish a tunnel connection through a single HTTP/2 stream to a remote host.

The HTTP/2 protocol includes much performance-enhancing changes that significantly improve performance, particularly for clients connecting over a mobile network.

The following table lists the major improvements in HTTP/2 over HTTP/1.1:

HTTP/2 features	Description
Header Compression	HTTP headers have much repetitive information and therefore consume unnecessary bandwidth during data transmission. HTTP/2 reduces bandwidth requirements by compressing the header and minimizing the requirement to transport HTTP headers with every request and response.
Connection Multiplexing	Latency can have a huge impact on page load times and the end user experience. Connection multiplexing overcomes this problem by sending multiple requests and responses across a single connection.
Server Push	Server push enables the server to proactively push content to the client browser, avoiding round trip delay. This feature caches the responses it thinks the client needs, reduces the number round trips, and improves the page rendering time. Important: The Citrix ADC appliance does not support the server push functionality.
No Head-of-line Blocking	Under HTTP 1.1, browsers can download one resource at a time per connection. When a browser has to download a large resource, it blocks all other resources from downloading until the first download is complete. HTTP/2 overcomes this problem with a multiplexing approach. It allows the client browser to download other web components in parallel over the same connection and display them as they become available.

HTTP/2 features	Description
Request Prioritization	Not all resources have equal priority when the browser renders a webpage. To accelerate the load time, all modern browsers prioritize requests by type of asset, their location on the page, and even by learned priority from previous visits. With HTTP/1.1, the browser has limited ability to use the priority data, because this protocol does not support multiplexing, and there is no way to communicate request prioritization by the server. The result is unnecessary network latency. HTTP/2 overcomes this problem by allowing the browser to dispatch all requests. The browser can communicate its stream prioritization preference via stream dependencies and weights, enabling the servers to optimize response delivery. Important: The Citrix ADC appliance does not support the request prioritization functionality.

How HTTP/2 works

A Citrix ADC appliance supports HTTP/2 on the client side as well on the server side. On the client side, the Citrix ADC appliance acts as a server that hosts an HTTP/HTTPS virtual server for HTTP/2. On the back-end side, the Citrix ADC acts as a client to the servers that are bound to the virtual server.

Therefore, the Citrix ADC appliance maintains separate connections on the client side as well on the server side. The Citrix ADC appliance has separate HTTP/2 configurations for the client side and the server side.

HTTP/2 for HTTPS (SSL) load balancing configuration

For an HTTPS load balancing configuration, the Citrix ADC appliance uses the TLS ALPN extension (RFC 7301) to determine whether the client/server supports HTTP/2. If it does, the appliance chooses HTTP/2 as the application-layer protocol to transmit data (as described in RFC 7540 - Section 3.3) on the client/server side.
The appliance uses the following order of preference when choosing the application-layer protocol through the TLS ALPN extension:

- HTTP/2 (if enabled in the HTTP profile)
- SPDY (if enabled in the HTTP profile)
- HTTP/1.1

HTTP/2 for HTTP load balancing configuration

For an HTTP load balancing configuration, the Citrix ADC appliance uses one of the following methods to start communicating with the client/server using HTTP/2.

> **Note:**
>
> In the following method descriptions, client and server are generals terms for an HTTP/2 connection. For example, for a load balancing setup of a Citrix ADC appliance using HTTP/2, the Citrix ADC appliance acts as a server on the client side and acts as a client to the server side.

- **HTTP/2 Upgrade**. A client sends an HTTP/1.1 request to a server. The request includes an upgrade header, which asks the server for upgrading the connection to HTTP/2. If the server supports HTTP/2, the server accepts the upgrade request and notifies it in its response. The client and the server start communicating using HTTP/2 after the client receives the upgrade confirmation response.
- **Direct HTTP/2**. A client directly starts communicating to a server in HTTP/2 instead of using the HTTP/2 upgrade method. If the server does not support HTTP/2 or is not configured to directly accept HTTP/2 requests, it drops the HTTP/2 packets from the client. This method is helpful if the admin of the client device already knows that the server supports HTTP/2.
- **Direct HTTP/2 using Alternative Service (ALT-SVC).** A server advertises that it supports HTTP/2 to a client by including an Alternative Service (ALT-SVC) field in its HTTP/1.1 response. If the client is configured to understand the ALT-SVC field, the client and the server start directly communicating using HTTP/2 after the client receives the response.

The Citrix ADC appliance provides configurable options in an HTTP profile for the HTTP/2 methods. These HTTP/2 options can be applied to the client side as well to the server side of an HTTPS or HTTP load balancing setup. For more information for HTTP/2 methods and options, refer to the HTTP/2 options PDF.

Before you Begin

Before you begin configuring HTTP/2 on a Citrix ADC appliance, note the following points:

- The Citrix ADC appliance supports HTTP/2 on the client side as well on the server side.
- The Citrix ADC appliance does not support the HTTP/2 server push functionality.
- The Citrix ADC appliance does not support the HTTP/2 request prioritization functionality.

- The Citrix ADC appliance does not support HTTP/2 SSL renegotiation for HTTPS load balancing setups.
- The Citrix ADC appliance does not support HTTP/2 NTLM authentication.
- HTTP/2 does not work if User Source IP (USIP) mode is enabled and the proxy mode is disabled on the Citrix ADC appliance.

Configuring HTTP/2

Configuring HTTP/2 for a load balancing setup (HTTPS or HTTP) consists of the following tasks:

- **Enable HTTP/2 and set optional HTTP/2 parameters in an HTTP Profile**. Enable HTTP/2 in an HTTP profile. When you only enable HTTP/2 in an HTTP profile, the Citrix ADC appliance uses only the upgrade method (for HTTP) or TLS ALPN method (for HTTPS) for communicating in HTTP/2.

  For the Citrix ADC appliance to use the direct HTTP/2 method, **Direct HTTP/2** option must be enabled in the HTTP profile. For the Citrix ADC appliance to use the direct HTTP/2 using the alternative service method, the **Alternative Service (altsvc)** option must be enabled in the HTTP profile.

- **Bind the HTTP profile to a virtual server or a service**. Bind the HTTP profile to a virtual server to configure HTTP/2 for the client side of the load balancing setup. Bind the HTTP profile to a service to configure HTTP2 for the server side of the load balancing setup.

> **Note:**
>
> Citrix recommends binding separate HTTP profiles for the client side and the server side.

- **Enable the global parameter for HTTP/2 server side support**. Enable the **HTTP/2 Service Side** (**HTTP2Serverside**) global HTTP parameter for enabling the HTTP/2 support on the server side of all the load balancing setups that has HTTP/2 configured.

  HTTP/2 does not work on the server side of any load balancing setups if **HTTP/2 Service Side** is disabled even if the **HTTP/2** is enabled on the HTTP profile bound to the related load balancing services.

Citrix ADC Command Line procedures:

To enable HTTP/2 and set HTTP/2 parameters by using the Citrix ADC command line

- To enable HTTP/2 and set HTTP/2 parameters while adding an HTTP profile, at the command prompt, type:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )] [-altsvc ( ENABLED | DISABLED )]
show ns httpProfile <name>
```


- To enable HTTP/2 and set HTTP/2 parameters while modifying an HTTP profile, at the command prompt, type:

```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED)] [-altsvc (ENABLED | DISABLED )]
show ns httpProfile <name>
```

To bind the HTTP profile to a virtual server by using the Citrix ADC command line

At the command prompt, type:

```
set lb vserver <name> - httpProfileName <string>
show lb vserver <name>
```

To bind the HTTP profile to a load balancing service by using the Citrix ADC command line

At the command prompt, type:

```
set service <name> -httpProfileName <string>
show service <name>
```

To enable HTTP/2 support globally on the server side by using the Citrix ADC command line

At the command prompt, type:

```
set ns httpParam -HTTP2Serverside( ENABLED | DISABLED )
show ns httpParam
```

To enable HTTP/2 and set HTTP/2 parameters by using the Citrix ADC GUI

1. Navigate to **System** > **Profiles**, and click **HTTP Profiles** tab.
2. Enable **HTTP/2** while adding an HTTP profile or modifying an existing HTTP profile.

To bind the HTTP profile to a virtual server by using the Citrix ADC GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In **Advanced Settings**, click **+ HTTP Profile** to bind the created HTTP profile to the virtual server.

To bind the HTTP profile to a load balancing service by using the Citrix ADC GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Service**, and open the service.
2. In **Advanced Settings**, click **+ HTTP Profile** to bind the created HTTP profile to the service.

To enable HTTP/2 support globally on the server side by using the GUI

Navigate to **System** > **Settings**, click **Change HTTP parameters** and enable **HTTP/2 Server Side**.

Sample configurations

In the following sample configuration, HTTP/2 and direct HTTP/2 is enabled on HTTP profile HTTP-PROFILE-HTTP2-CLIENT-SIDE. The profile is bound to virtual server LB-VS-1.


```
set ns httpProfile HTTP-PROFILE-HTTP2-CLIENT-SIDE -http2 enabled -
   http2Direct enabled
Done

set lb vserver LB-VS-1 -httpProfileName HTTP-PROFILE-HTTP2-CLIENT-SIDE

Done
<!--NeedCopy-->
```

In the following sample configuration, HTTP/2 and alternative service (ALT-SVC) is enabled on HTTP profile HTTP-PROFILE-HTTP2-SERVER-SIDE. The profile is bound to service LB-SERVICE-1.

```
set ns httpparam -HTTP2Serverside ENABLED
Done

set ns httpProfile HTTP-PROFILE-HTTP2-SERVER-SIDE -http2 ENABLED -
   altsvc ENABLED
Done

set service LB-SERVICE-1 -httpProfileName HTTP-PROFILE-HTTP2-SERVER-
   SIDE
Done
<!--NeedCopy-->
```

Configure HTTP/2 initial connection window size

As per RFC 7540, the flow-control window for HTTP2 stream and connection must be set to 64 K (65535) octets, and any change made to this value must be communicated to the peer. The ADC appliance communicates the change in flow-control window size as follows:

- Using the `SETTINGS` frame for the stream.
- Using the `WINDOW_UPDATE` frame for the connection.

In an HTTP profile, you must configure the `http2InitialWindowSize` parameter to set the initial window size at the stream level. Because of an internal system error, the ADC appliance initializes the flow-control window for the connection also. When there is a change in the configured flow-control window for the stream, the ADC appliance communicates to the peer using the SETTINGS frame. But the ADC appliance fails to communicate the change in flow-control window for the connection using the `WINDOW_UPDATE` frame. This leads to a connection freeze.

To overcome the issue, the `http2InitialConnWindowSize` parameter (in bytes) is now added to control the flow-control window for connection. By using separate configurable parameters, you can

now enable the appliance to send updates for changed window size at both stream and connection levels.

Configure the HTTP/2 initial connection window size parameter by using the CLI

At the command prompt, type:

```
set http profile p1 -http2InitialConnWindowSize 8290
Initial window size for stream level flow control, in bytes.
Default value: 65535
Minimum value: 8192
Maximum value: 20971520
<!--NeedCopy-->
```

HTTP/2 DoS mitigation

September 14, 2021

The Http/2 Denial-of-Service (DoS) attacks no longer have any impact on a Citrix ADC appliance. If the appliance receives frames more than the maximum limit, the appliance silently closes the connection.

To mitigate attacks, HTTP profile enables you to change the default configuration of frames received in a HTTP/2 connection.

The HTTP/2 DoS mitigation table shows the list of HTTP/2 DoS attacks and its mitigation.

Configure the maximum limit for HTTP/2 frames to mitigate DoS attacks by using the command line interface

At the command prompt, type the following:

```
set ns httpprofile <profile_name> - http2MaxEmptyFramesPerMin <positive_integer
> -http2MaxPingFramesPerMin <positive_integer> -http2MaxSettingsFramesPerMin
 <positive_integer> -http2MaxResetFramesPerMin <positive_integer>
```

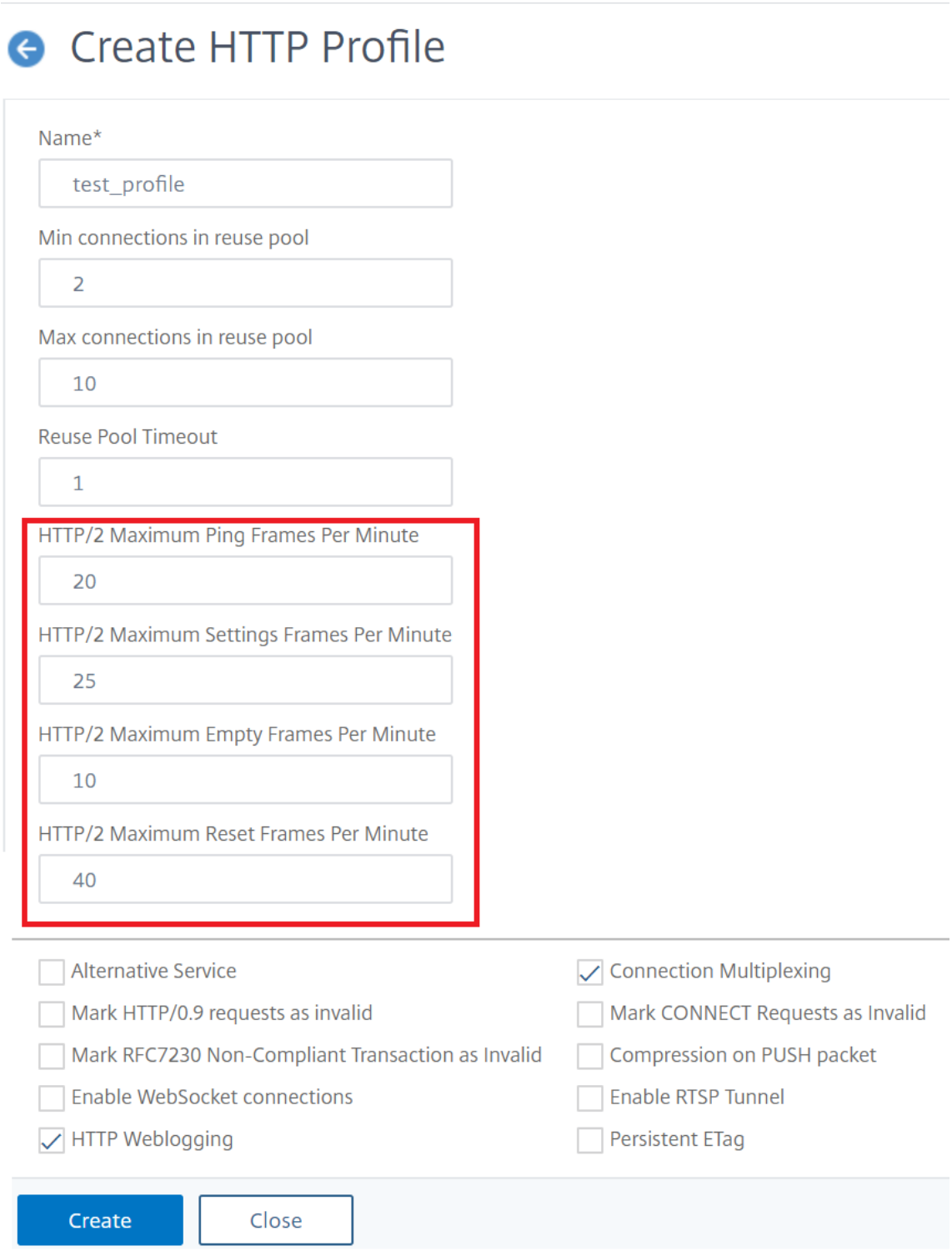
Example:

```
set ns httpprofile profile1 -http2MaxEmptyFramesPerMin 20 -http2MaxPingFramesPerMin
 20 -http2MaxSettingsFramesPerMin 20 -http2MaxResetFramesPerMin 20
```


Configure the maximum limit for frames received in a HTTP/2 connection by using the Citrix ADC GUI

Follow the steps given below to configure the maximum limit for frames received in a HTTP/2 connection:

1. On the navigation pane, expand **System** and then click **Profiles**.
2. On the **Profile** page, select the **HTTP Profiles** tab.
3. In the **HTTP Profiles** tab page, click **Add**.
4. In the **Configure HTTP Profile** page, set the following parameter.
    a) http2MaxPingFramesPerMin. Set the maximum PING frames received per connection in a minute. If the number of PING frames exceed configuration limit, the appliance silently drops packets on the connection.
    b) http2MaxSettingsFramesPerMin. Set the maximum SETTINGS frames received per connection in a minute. If the number of SETTINGS frames exceed configuration limit, ADC silently drops packets on the connection.
    c) http2MaxResetFramesPerMin. Set the maximum RESET frames sent per connection in a minute. If the number of RESET frames exceed configuration limit, ADC silently drops packets on the connection.
    d) http2MaxEmptyFramesPerMin. Set the maximum empty frames sent per connection in a minute. If the number of empty frames exceed configuration limit, ADC silently drops packets on the connection.
5. Click **OK** and **Close**.



HTTP3 over QUIC protocol

September 14, 2021

HTTP/2 over TCP is the preferred standard for sending multiple streams of HTTP requests over a single connection. But, in TCP transport mechanism there are certain limitations and latency issues in accessing websites and web applications. When you multiplex several requests over the same connection they are subjected to reliability of the same connection. If packet for one request is lost, all other multiplexed requests are delayed until the lost packet is detected and retransmitted. This causes head-of-line-blocking delays and latency issues.

For connection and transport delays, HTTP/3 uses QUIC instead of TCP protocol. The QUIC is an emerging protocol that uses UDP instead of TCP as its base transport. In HTTP-over-QUIC, you can multiplex several independent requests without depending on a single TCP connection. QUIC implements a reliable connection upon which, you can stream multiple HTTP requests. QUIC also incorporates TLS as an integrated component and not as an additional layer as in HTTP/1.1 or HTTP/2.

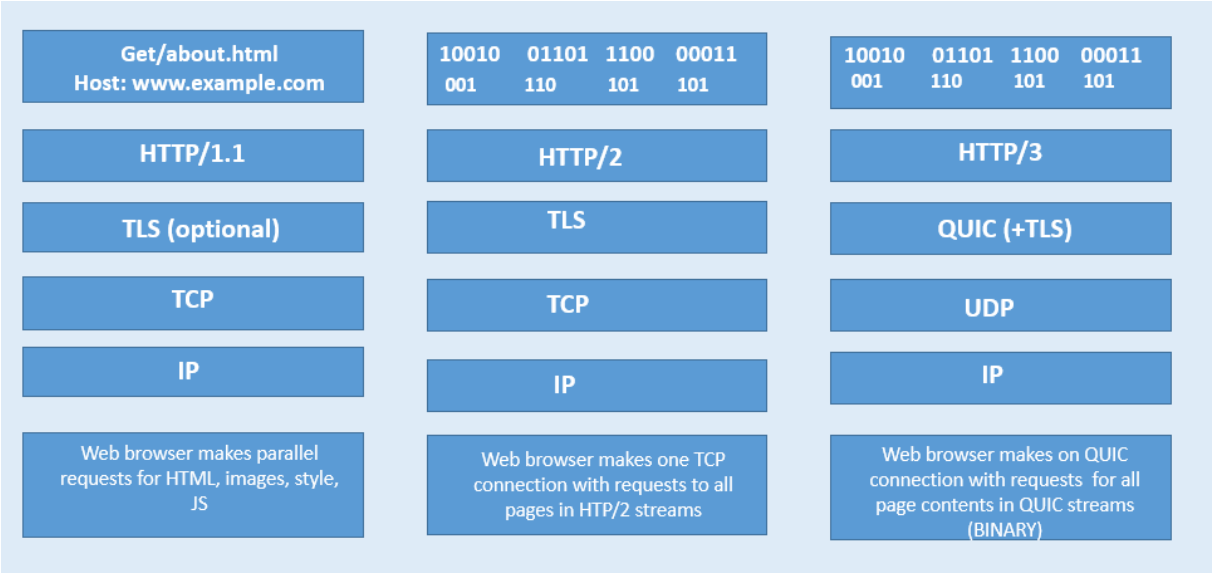
Advantage of using HTTP/3 protocol

Some of the important benefits of using QUIC protocol for HTTP/3 data transport is given below:

- Stream multiplexing
- Stream and connection-level flow control
- Low-latency connection establishment
- Connection migration and resilience to NAT rebinding
- Authenticated and encrypted header and payload

Transport stack in HTTP protocols

The below illustration shows transport stack in HTTP/1.1, HTTP/2 and HTTP/3 protocols.



How QUIC and HTTP/3 connection management works in Citrix ADC

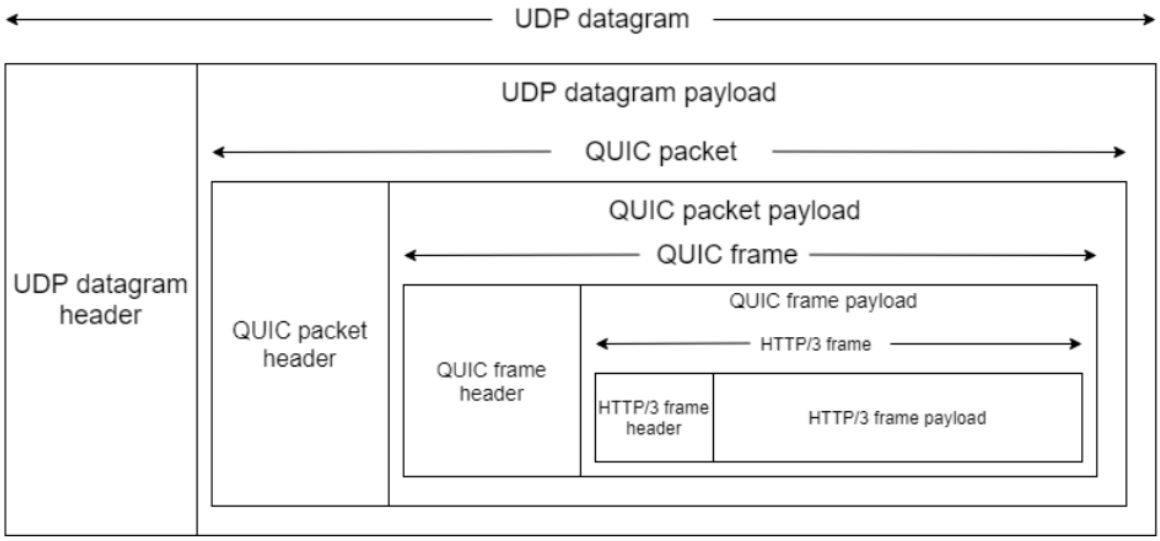
The following illustration shows how QUIC and HTTP/3 connection management in a Citrix ADC appliance and how the components interact with each other.

Step 1: Client-side HTTP/3 request over QUIC protocol to Citrix ADC appliance.
Step 2: Request forwarded by Citrix ADC AS HTTP/1.1 or HTTP/2 depending on back-end server support.
Step 3: Response through HTTP/2 or HTTP/1.1 from back-end server to Citrix ADC.
Step 4: ADC forwards the response as HTTP/3 response to client.

How HTTP/3 protocol works

In HTTP/3, when a client knows that an HTTP/3 server exists at a certain endpoint, it opens a QUIC connection. The QUIC protocol provides multiplexing and flow control. Within each stream, the basic unit of HTTP/3 communication is a frame. Each frame type serves a different purpose. For example, HEADERS and DATA frames form the basis of HTTP requests and responses.

Multiplexing of requests is performed using the QUIC stream abstraction. Each request-response pair consumes a single QUIC stream. Streams are independent of each other, so one stream that is blocked or suffers packet loss does not prevent progress on other streams. Server push is an interaction mode introduced in HTTP/2 which permits a server to push a request-response exchange to a client in anticipation of the client making the indicated request. This trades off network usage against a potential latency gain. Several HTTP/3 frames are used to manage server push, such as PUSH_PROMISE, MAX_PUSH_ID, and CANCEL_PUSH. As in HTTP/2, request and response fields are compressed for transmission. Because HPACK relies on in-order transmission of compressed field sections (a guarantee not provided by QUIC), HTTP/3 replaces HPACK with QPACK. QPACK uses separate unidirectional streams to modify and track field table state, while encoded field sections refer to the state of the table without modifying it.



HTTP/3 configuration and Stat summary

September 14, 2021

To configure a HTTP/3 protocol for sending multiple streams of HTTP/3 data using QUIC, you must complete the following steps:

1. Enable SSL and load balancing features.
2. Add load balancing and content switching (optional) virtual servers of type HTTP_QUIC.
3. Associate QUIC protocol parameters with the HTTP_QUIC virtual server.
4. Enable HTTP/3 on the HTTP_QUIC virtual server.
5. Bind SSL certificate-key pair with HTTP_QUIC virtual server.
6. Associate SSL/TLS protocol parameters with the HTTP_QUIC virtual server.

Enable SSL and load balancing

Before you begin, make sure that the SSL and Load Balancing features is enabled on the appliance. At the command prompt type:

```
enable ns feature ssl lb
<!--NeedCopy-->
```

Add load balancing and content switching (optional) virtual servers of type HTTP_QUIC for HTTP/3 service

You add a load balancing virtual server to accept HTTP/3 traffic over QUIC.
Note: The load balancing virtual server of type HTTP_QUIC has built-in QUIC, SSL, and HTTP3 profiles. If you prefer to create user-define profiles, you can add new profiles and bind it with the load balancing virtual server.

```
add lb vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-
   port>
add cs vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-
   port>
<!--NeedCopy-->
```

Example:

```
add lb vserver lb-http3 HTTP_QUIC 1.1.1.1 443
add cs vserver cs-http3 HTTP_QUIC 10.10.10.10 443
```

Associate QUIC protocol parameters with HTTP_QUIC virtual server

You can create a QUIC profile and specify QUIC parameters for the QUIC service and associate it to the load balancing virtual server. You must either create a user-defined profile or use the in-built QUIC profile and bind the profile to the load balancing virtual server.

Step 1: configure a user-defined QUIC profile
At the command prompt, type:

```
set quic profile <profile_name> -transport_param <value>
<!--NeedCopy-->
```

Example:

```
set quic profile quic_http3 -ackDelayExponent 10 -activeConnectionIDlimit 4
```

The different QUIC transport parameters are as follows:

-ackDelayExponent. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, indicating an exponent that the remote QUIC endpoint should use, to decode the ACK Delay field in QUIC ACK frames sent by the Citrix ADC.

-activeConnectionIDlimit. An integer value advertised by the Citrix ADC to the remote QUIC endpoint. It specifies the maximum number of QUIC connection IDs from the remote QUIC endpoint, that the Citrix ADC is willing to store.

-activeConnectionMigration. Specify whether the Citrix ADC must allow the remote QUIC endpoint to perform active QUIC connection migration.

-congestionCtrlAlgorithm. Specify the congestion control algorithm to be used for QUIC connections.

-initialMaxData. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the initial value, in bytes, for the maximum amount of data that can be sent on a QUIC connection.

-initialMaxStreamDataBidiLocal. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the initial flow control limit, in bytes, for bi-directional QUIC streams initiated by the Citrix ADC.

-initialMaxStreamDataBidiRemote. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the initial flow control limit, in bytes, for bi-directional QUIC streams initiated by the remote QUIC endpoint.

-initialMaxStreamDataUni. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the initial flow control limit, in bytes, for uni-directional streams initiated by the remote QUIC endpoint.

-initialMaxStreamsBidi. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the initial maximum number of bi-directional streams the remote QUIC endpoint must initiate.

-initialMaxStreamsUni. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the initial maximum number of uni-directional streams the remote QUIC endpoint must initiate.

-maxAckDelay. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the maximum amount of time, in milliseconds, by which the Citrix ADC delays sending acknowledgments.

-maxIdleTimeout. An integer value advertised by the Citrix ADC to the remote QUIC endpoint, specifying the maximum idle timeout, in seconds, for a QUIC connection. A QUIC connection that remains idle, for longer than the minimum of the idle timeout values advertised by the Citrix ADC and the remote QUIC endpoint, and three times the current Probe Timeout (PTO), will be silently discarded by the Citrix ADC.

-maxUDPPayloadSize. An integer value advertised by the Citrix ADC to the remote QUIC endpoint,

specifying the size of the largest UDP datagram payload, in bytes, that the Citrix ADC is willing to receive on a QUIC connection.

-newTokenValidityPeriod. An integer value, specifying the validity period, in seconds, of address validation tokens issued through QUIC NEW_TOKEN frames sent by the Citrix ADC.
-retryTokenValidityPeriod. An integer value, specifying the validity period, in seconds, of address validation tokens issued through QUIC Retry packets sent by the Citrix ADC.
-statelessAddressValidation. Specify whether the Citrix ADC must perform stateless address validation for QUIC clients, by sending tokens in QUIC Retry packets during QUIC connection establishment, and by sending tokens in QUIC NEW_TOKEN frames after QUIC connection establishment.

Step 2: Associate the user-defined QUIC profile to a load balancing virtual server of type http_quic

At the command prompt, type:

```
set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
    serviceName>@] [-persistenceType <persistenceType>] [-
    quicProfileName <string>]
<!--NeedCopy-->
```

Example:

```
set lb vserver lb-http3 -quicProfileName quic_http3
```

Enable and bind HTTP/3 on a HTTP_QUIC virtual server

To enable HTTP/3 on an HTTP_QUIC virtual server, a set of configuration parameters is added to the HTTP profile configuration. To facilitate ease of configuration, when you add an HTTP_QUIC virtual server, a new default/built-in HTTP profile is available on the appliance. The profile has the HTTP/3 protocol support parameters set to ENABLED, and also bounded to the HTTP_QUIC virtual servers (applicable if you choose not to associate the HTTP_QUIC virtual server with a user-added HTTP profile). The value of the HTTP/3 parameters in the HTTP profile decides whether to select the HTTP/3 protocol and advertise when processing the TLS ALPN (Application Layer Protocol Negotiation) extension, during the QUIC protocol handshake.

You can create a HTTP/3 profile and specify HTTP parameters for the HTTP/3 service and load balancing virtual server. You must either create a user-defined profile or use the in-built HTTP/3 profile and bind the profile to the load balancing virtual server.

Step 1: configure a user-defined HTTP/3 profile
At the command prompt, type:

```
Add ns httpProfile <profile_name> -http3 ENABLED
<!--NeedCopy-->
```


Example:

```
add ns httpProfile http3_quic -http3 ENABLED
```

Step 2: Bind the user-defined HTTP/3 profile to a load balancing virtual server of type http_quic
At the command prompt, type:

```
1 set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
     serviceName>@] [-persistenceType <persistenceType>] [-
     httpProfileName <string>]
2 <!--NeedCopy-->
```

Example:

```
set lb vserver lb-http3 -httpProfileName http3_quic
```

Bind SSL certificate-key pair with HTTP_QUIC virtual server

For processing encrypted traffic, you must add an SSL certificate-key pair and bind it to the HTTP_QUIC virtual server.

At the command prompt, type:

```
1 bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
2
3 <!--NeedCopy-->
```

Example:

```
bind ssl vserver lb-http3 -certkeyName rsa_certkeypair
```

For more information, see Bind SSL certificate topic.

Bind SSL/TLS protocol parameters with a HTTP_QUIC virtual server

Virtual servers of type HTTP_QUIC has in-built TLS 1.3 server functionality because the QUIC protocol uses TLS 1.3 as a mandatory security component. To facilitate the configuration when adding a HTTP_QUIC virtual server, a new default or built-in SSL profile of type - QUIC-FrontEnd is added. The SSL profile has TLS 1.3 version enabled with TLS 1.3 cipher suites (and elliptic curves) configured. The SSL profile must then be bound to the newly added HTTP_QUIC virtual servers.
You can create an SSL profile and specify SSL encryption parameters for the TLP 1.1 service and load balancing virtual server. You must either create a user-defined profile or use the in-built SSL profile and bind the profile to the load balancing virtual server.

Step 1: configure a user-defined SSL profile
At the command prompt, type:


```
add ssl profile <name> -sslprofileType QUIC-FrontEnd
<!--NeedCopy-->
```

Example:

```
add ssl profile ssl_profile1 -sslprofileType QUIC-FrontEnd -tls13 ENABLED -
tls12 DISABLED -tls11 DISABLED -tls1 DISABLED
```

Step 2: Bind the user-defined SSL profile to a load balancing virtual server of type HTTP_QUIC
At the command prompt, type:

```
set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
    serviceName>@] [-persistenceType <persistenceType>] [-
    httpProfileName <string>]
<!--NeedCopy-->
```

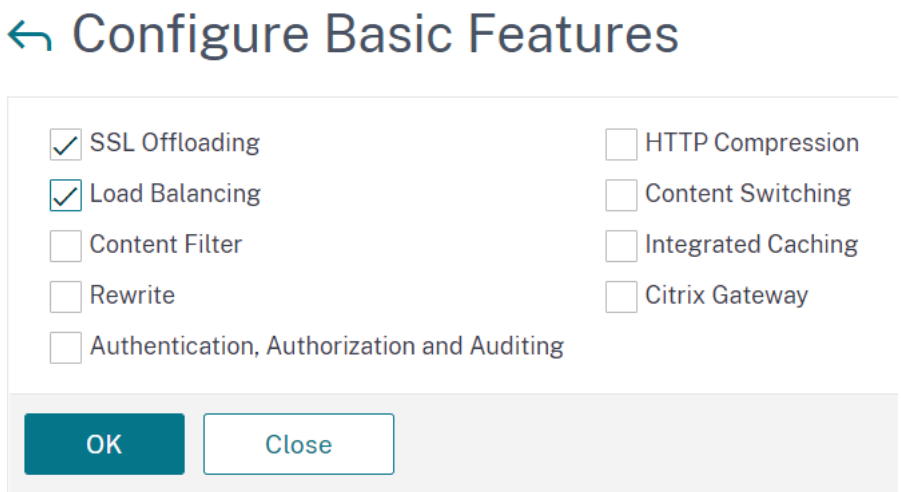
Example:

```
set ssl vserver lb-http3 -sslprofile ssl_profile1
```

Enable SSL and load balancing features by using the GUI

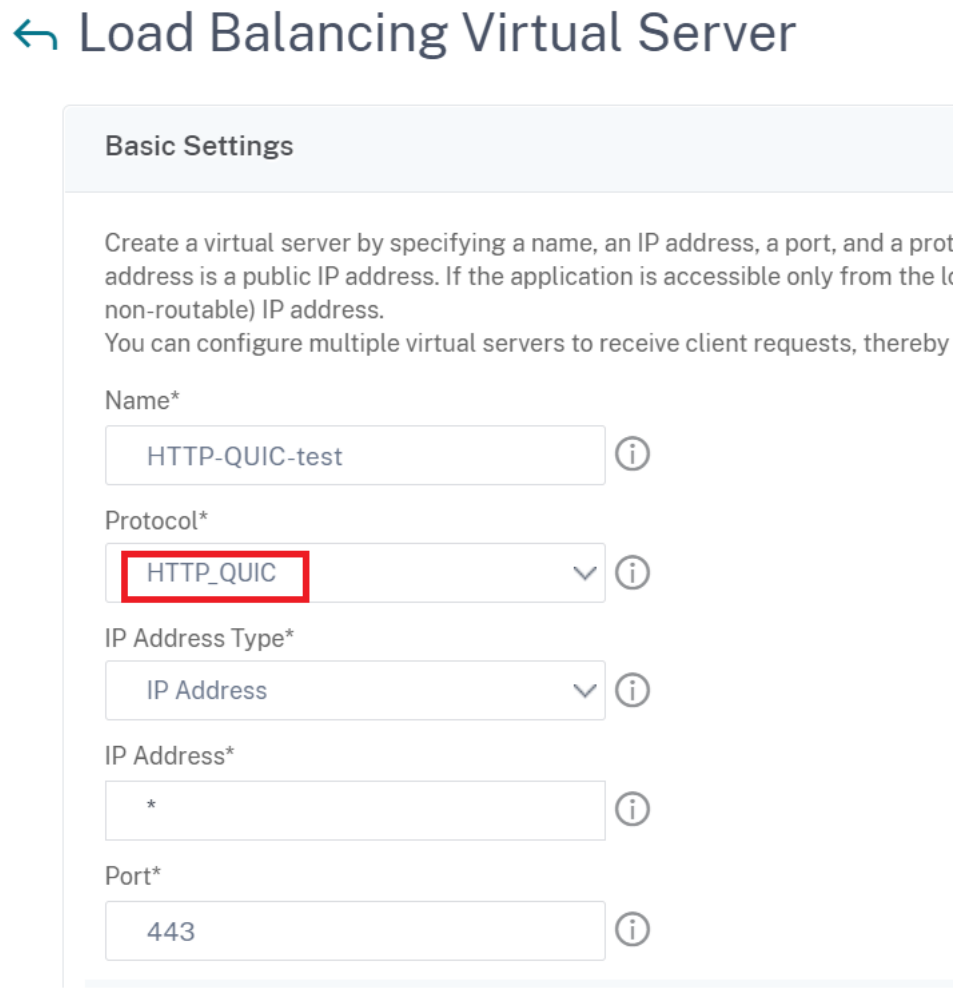
Complete the following steps to enable SSL and load balancing features:

1. On the navigation pane, expand **System** and then click **Settings**.
2. On the **Configure Basic Features** page, select the **SSL** and **Load Balancing**.
3. Click **OK**, and then click **Close**.



Add load balancing and content switching (optional) virtual servers of type HTTP_QUIC by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Click **Add** to create a load balancing virtual server of type HTTP_QUIC.
3. In **Load Balancing Virtual Server** page, click **Profiles**.
4. In the **Profiles** section, select the profile type as QUIC. Note: QUIC, HTTP/3 and SSL profiles are built-in ones.
5. Click **OK** and then **Done**.



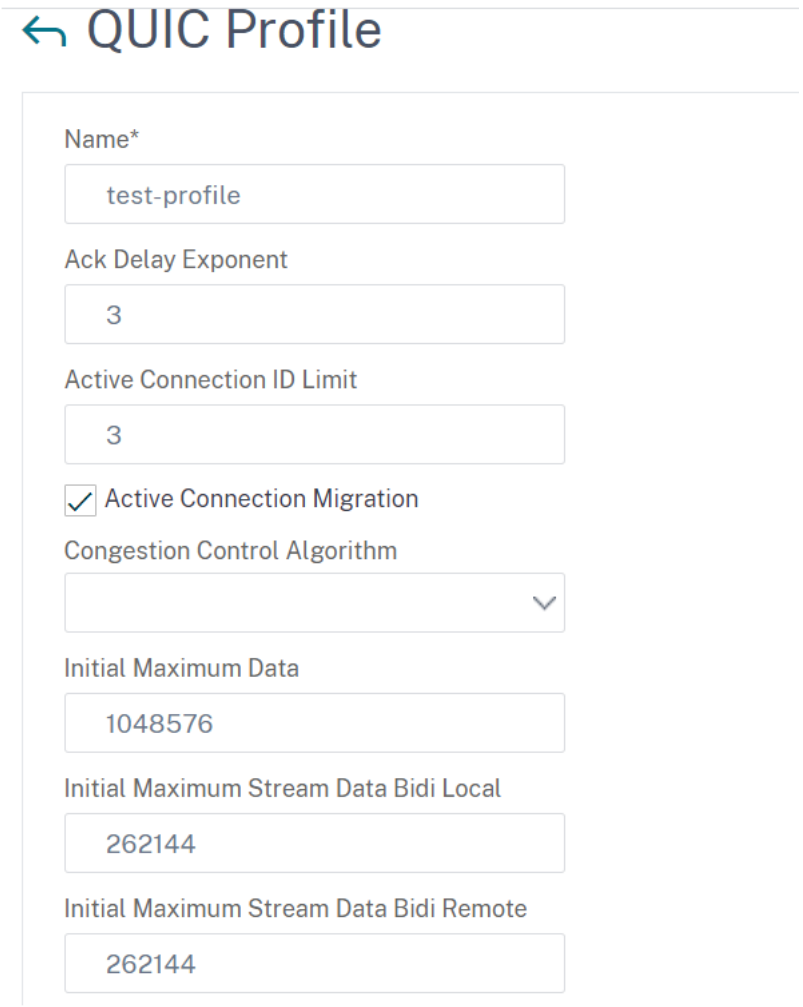
Associate QUIC protocol parameters with the HTTP_QUIC virtual server by using the GUI

Step 1: Add QUIC profile

1. Navigate to **System > Profiles > QUIC Profile**.
2. Click **Add**.

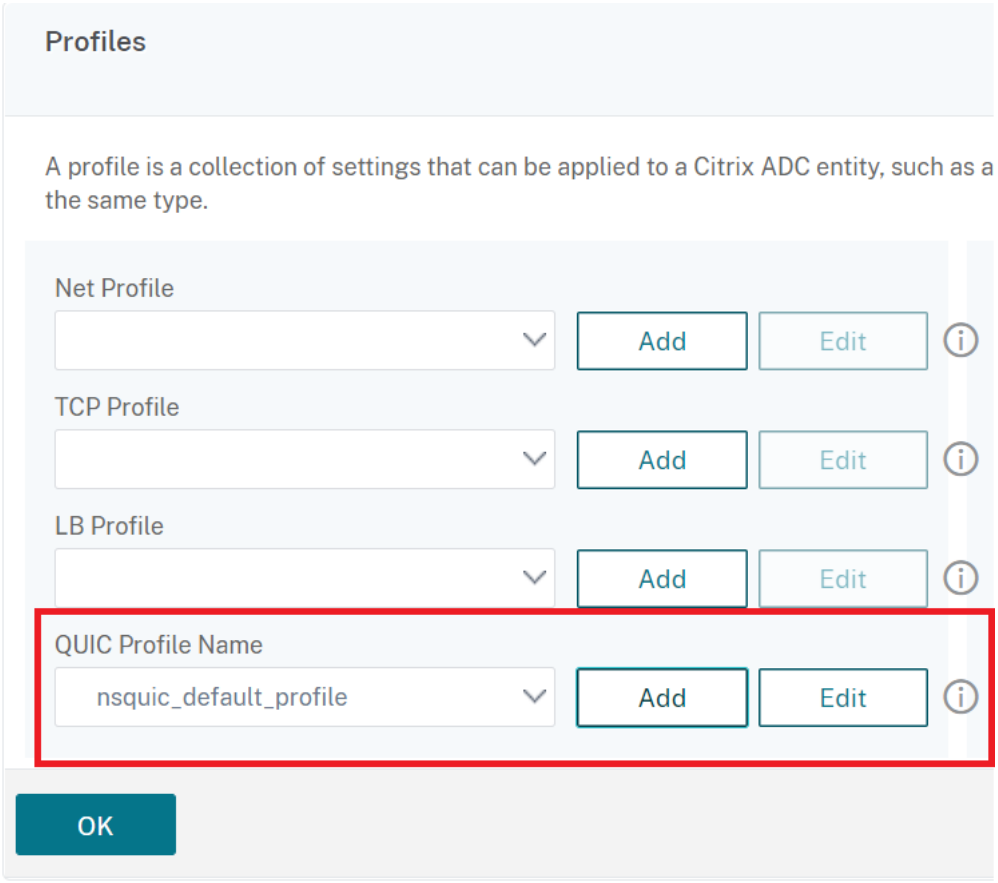
3. In the QUIC Profile page, set the following parameters. For detailed description of each parameter, see the Associate QUIC protocol CLI section.

   a) `Ack Delay` Exponent
   b) Active Connection ID Limit
   c) Active Connection Migration
   d) Congestion Control Algorithm
   e) Initial Maximum Data
   f) Initial Maximum Stream Data Bidi Local
   g) Initial Maximum Stream Data Bidi Remote
   h) Initial Maximum Stream Data Unit
   i) Initial Maximum Stream bidi
   j) Initial Maximum Stream Uni
   k) Maximum Acknowledgment Delay
   l) Maximum Idle Timeout
   m) Maximum UDP Data GramsperBurst
   n) New Token Validity Period
   o) Retry Token Validity Period
   p) Stateless Address Validation



Step 2: Associate QUIC profile with load balancing virtual server of type HTTP_QUIC

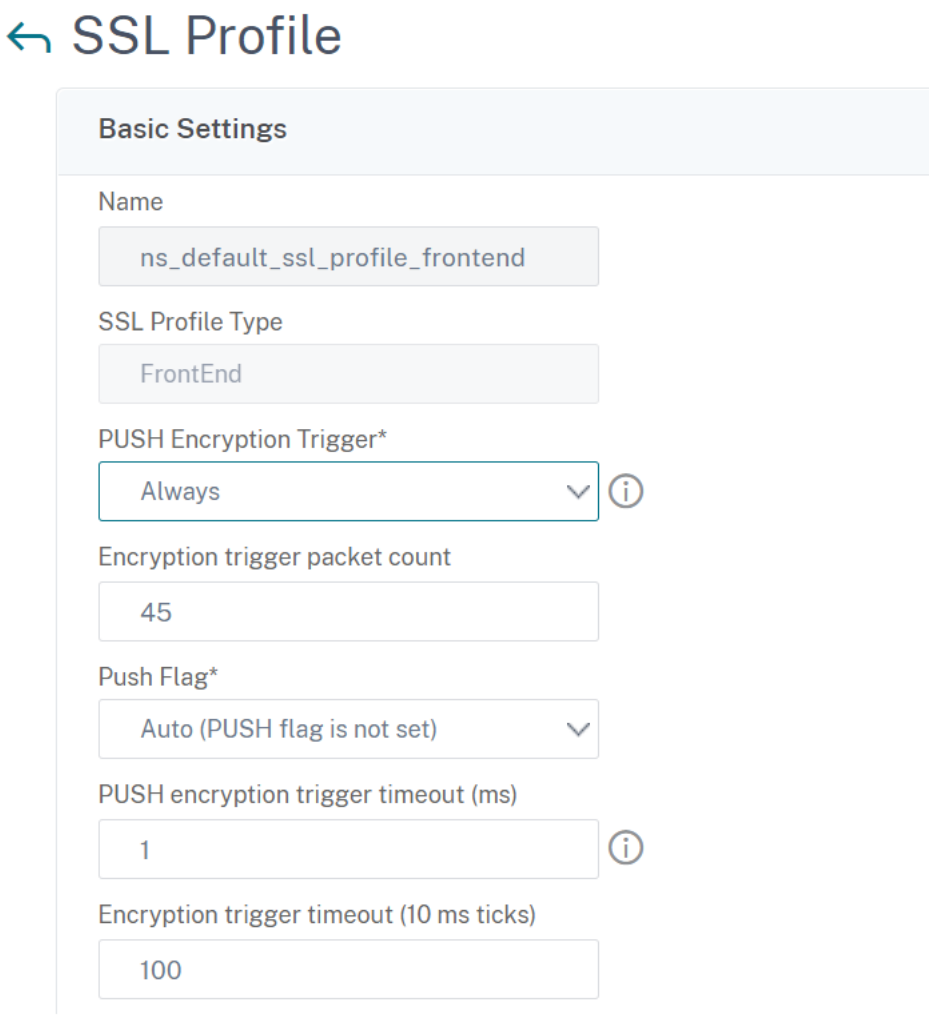
1. In the **Profiles** section, select the QUIC profile. Note: QUIC, HTTP/3 and SSL profiles are built-in ones.
2. Click **OK** and then **Done**.



Associate SSL/TLS protocol parameters with the virtual server of type SSL by using the GUI

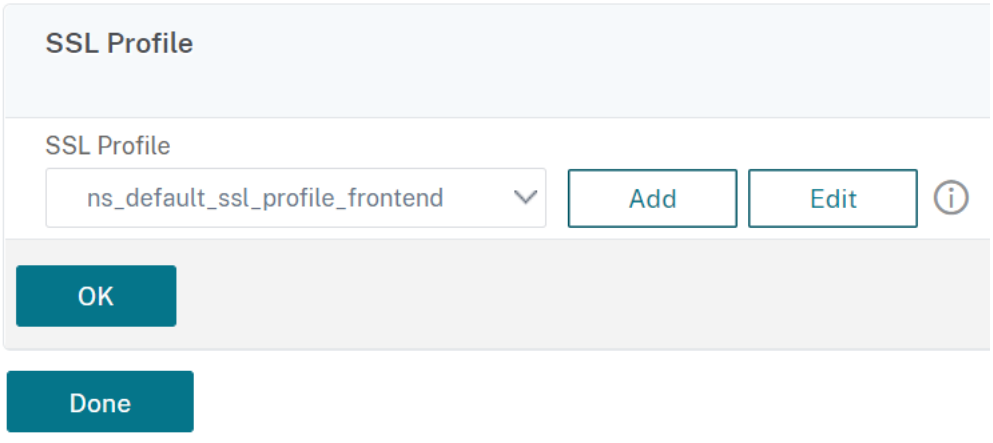
Step 1: Add SSL profile

1. Navigate to **System > Profiles > SSL Profile**.
2. Click **Add**.
3. In the **QUIC Profile** page, set the SSL parameters. For detailed description see, SSL Profile configuration topic.
4. Click **OK** and **Close**.



Step 2: Associate SSL profile with load balancing virtual server of type SSL.

1. In the **Profiles** section, select the SSL profile.
2. Click **OK** and then **Done**.



View QUIC, and HTTP/3 statistics

The following commands display a detailed summary of QUIC, and HTTP3 statistics. At the command prompt, type the following:

```
> stat quic
> stat quic  - detail
<!--NeedCopy-->
```

To clear the statistics display, type one of the following:

```
> stat quic -clearstats basic
> stat quic -clearstats full

<!--NeedCopy-->
```

To display a detailed summary of HTTP/3 statistics:

```
> stat http3
> stat http3  - detail
<!--NeedCopy-->
```

To clear the statistics display, type one of the following:

```
> stat http3 -clearstats basic
> stat http3 -clearstats full
<!--NeedCopy-->
```

Policy configuration for HTTP/3 traffic

September 14, 2021

HTTP/3 uses QUIC transport which is based on UDP. If you had policy expression defined for the HTTP or SSL virtual server that includes TCP policy expressions, it can no longer be used with a HTTP_QUIC virtual server. All other policies that do not have TCP or classic expressions can be bound with a HTTP_QUIC virtual server. For the policies to take effect, you must ensure that the feature policies are bound to the newly added global bind points as per the following.

- HTTPQUIC_REQ_DEFAULT
- HTTPQUIC_REQ_OVERRIDE
- HTTPQUIC_RES_DEFAULT
- HTTPQUIC_RES_OVERRIDE

Or, the policies can be bound to specific virtual server bind points:

- REQUEST
- RESPONSE

For more information, see Bind policy using advanced policy infrastructure topic.

Following are the policies supported for HTTP over QUIC configuration:

- Responder
- Rewrite
- HTTP Compression
- Integrated Caching
- Web Application Firewall
- URL transformation
- SSL
- Front end optimization (FEO)
- AppQoE

Responder policy configuration for HTTP/3 traffic

HTTP over QUIC type virtual servers have responder policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included.

New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 QUIC virtual servers or HTTP over QUIC global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Add responder action for redirecting URLs

To add a responder action, at the command prompt, type:

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
    expression>] [-headers <name(value)> ...]
<!--NeedCopy-->
```

Example:

```
add responder action redirectURL redirect "\"https://www.citrix.com/\""
```

Add responder policy

To add a responder policy, at the command prompt, type:


```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

Example:

```
add responder policy res-pol "CLIENT.IP.SRC.IN_SUBNET(10.10.10.10/32)"
redirectURL
```

Add responder policy based UDP expression

To add a responder policy based UDP expression, at the command prompt, type:

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

Example:

```
add responder policy redirectCitrixUdp "CLIENT.UDP.DSTPORT.EQ(443)"redirectURL
```

Bind responder policy based UDP expression with HTTP/3 QUIC based load balancing virtual server

To bind a responder policy based UDP expression to a load balancing virtual server, at the command prompt, type:

```
1 bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
      ) | <serviceGroupName>@ |         (-policyName <string>@  [-
     priority <positive_integer>]  [-gotoPriorityExpression <expression>]
       [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
     analyticsProfile <string>@)
2 <!--NeedCopy-->
```

Example:

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 9 -gotoPriorityExpres
 END -type REQUEST
```

Bind responder policy with HTTP/3 QUIC based load balancing virtual server

To bind a responder policy to a load balancing virtual server, at the command prompt, type:


```
bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
     ) | <serviceGroupName>@ |         (-policyName <string>@  [-
    priority <positive_integer>]  [-gotoPriorityExpression <expression>]
      [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
    analyticsProfile <string>@)
<!--NeedCopy-->
```

Example:

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 10 -
gotoPriorityExpression END -type REQUEST
```

Bind responder policy to HTTP/3 global bind point

To bind a responder policy with the HTTP/3 global bind point, at the command prompt, type:

```
bind responder global <policyName> <priority>  [<gotoPriorityExpression
    >] [-type <type>] [-invoke (<labelType>  <labelName>) ] bind
    responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
<!--NeedCopy-->
```

Example:

```
bind responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
```

> **Note:**
>
> For more information, see Responder policy documentation.

Rewrite policy configuration for HTTP/3 traffic

HTTP over QUIC type virtual servers have rewrite policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included.

New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Following are the configuration steps to configure the rewrite policy for HTTP3 over QUIC.

Add rewrite action for HTTP over QUIC

To add rewrite action, at the command prompt, type:


```
add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
    pattern <expression> | -search <expression>] [-refineSearch <
    expression>] [-comment <string>]
<!--NeedCopy-->
```

Example:

```
add rewrite action http3-altsvc-action insert_http_header Alt-Svc q/"h3
-29=\":443\"; ma=3600; persist=1"/
```

Add rewrite policy for HTTP over QUIC

To add a write action, at the command prompt, type:

```
add rewrite policy <name> <rule> <action> [<undefAction>] [-comment  <
    string>] [-logAction <string>]
<!--NeedCopy-->
```

Example:

```
add rewrite policy http3-altsvc-policy true http3-altsvc-action
```

Bind rewrite policy to load balancing virtual server of type HTTP/3_QUIC

To bind rewrite policy to the load balancing virtual server, at the command prompt, type:

```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] )
     | <serviceGroupName>@ | (-policyName <string>@ [-priority <
    positive_integer>] [-gotoPriorityExpression <expression>] [-type <
    type>] [-invoke (<labelType> <labelName>) ] ) | -analyticsProfile  <
    string>@)
<!--NeedCopy-->
```

Example:

```
bind lb vserver lb-http3 -policyName http3-altsvc-policy -priority 10 -type
 RESPONSE
```

Bind rewrite policy to HTTP/3 global bind point

```
To bind a responder policy with HTTP/3 global bind point, at the
    command prompt, type:
bind rewrite global <policyName> <priority> [<gotoPriorityExpression>]
     [-type <type>] [-invoke (<labelType> <labelName>)]
```



```
<!--NeedCopy-->
```

Example:

```
bind rewrite global http3-altsvc-policy 3 -type HTTPQUIC_RES_DEFAULT
```

> **Note:**
>
> For more information, see Rewrite policy documentation.

Compression policy configuration for HTTP/3 traffic

When the Citrix ADC receives an HTTP response from a server, it evaluates the built-in compression policies and any custom compression policies to determine whether to compress the response and, if so, the type of compression to apply. Priorities assigned to the policies determine the order in which the policies are matched against the requests.
HTTP over QUIC type virtual servers have compression policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included.
New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Add compression policy

To add compression policy, at the command prompt, type:

```
add cmp policy <name> -rule <expression> -resAction <string>
<!--NeedCopy-->
```

Example:

```
add cmp policy udp_port_cmp_policy -rule "CLIENT.UDP.DSTPORT.EQ(443)"-
resAction COMPRESS
```

Bind compression policy with load balancing virtual server of type HTTP/3_QUIC

To bind URL transformation policy with load balancing virtual server of type HTTP/3_QUIC, at the command prompt, type:


```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
    ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
    positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
     REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
    -analyticsProfile <string>@)
<!--NeedCopy-->
```

Example:

```
bind lb vserver lb-http3 -policyName udp_port_cmp_policy -priority 10 -type
 RESPONSE
```

Bind compression global to HTTP/3 global bind point

To bind a compression policy with the HTTP/3 global bind point, at the command prompt, type:

```
bind compression global <policyName> <priority>  [<
    gotoPriorityExpression>] [-type <type>] [-invoke (<labelType>  <
    labelName>) ] bind responder global redirectCitrixUdp 3 -type
    HTTPQUIC_REQ_DEFAULT
<!--NeedCopy-->
```

Example:

```
bind cmp global udp_port_cmp_policy -priority 100 -type HTTPQUIC_RES_DEFAULT
 Global built-in compression policies
```

After you upgrade your appliance to Citrix ADC release 13.0 build 82.x, the following compression policies will be automatically bound to the HTTP/3 default bind point.

```
> sho cmp global -type HTTPQUIC_RES_DEFAULT
        Policy Name: ns_adv_nocmp_xml_ie
        Priority: 8700
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_nocmp_mozilla_47
        Priority: 8800
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_mscss
        Priority: 8900
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

```



```
17          Policy Name: ns_adv_cmp_msapp
18          Priority: 9000
19          GotoPriorityExpression: END
20          Type: HTTPQUIC_RES_DEFAULT
21
22          Policy Name: ns_adv_cmp_content_type
23          Priority: 10000
24          GotoPriorityExpression: END
25          Type: HTTPQUIC_RES_DEFAULT
26  <!--NeedCopy-->
```

If not bound, the following commands can be configured through the command prompt and you can configuration on your appliance.

```
bind cmp global ns_adv_nocmp_xml_ie -priority 8700 -gotoPriorityExpression
END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_nocmp_mozilla_47 -priority 8800 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_mscss -priority 8900 -gotoPriorityExpression END
 -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_msapp -priority 9000 -gotoPriorityExpression END
 -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_content_type -priority 10000 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT
```

For more information, see Compression policy configuration.

Caching policy configuration for HTTP/3 traffic

The integrated cache provides in-memory storage on the Citrix ADC appliance and serves Web content to users without requiring a round trip to an origin server. For static content, the integrated cache requires little initial setup. After you enable the integrated cache feature and perform basic setup (for example, determining the amount of Citrix ADC appliance memory the cache is permitted to use), the integrated cache uses built-in policies to store and serve specific types of static content, including simple webpages and image files. You can also configure the integrated cache to store and serve dynamic content that is marked as non-cacheable by Web and application servers (for example, database records and stock quotes).
HTTP over QUIC type virtual servers have cache policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included.

New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can

be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Add cache content group

To add the cache content group, at the command prompt, type:

```
1 add cache contentGroup <name> [-weakPosRelExpiry <secs> | -relExpiry <
      secs> | -relExpiryMilliSec <msecs> | -absExpiry <HH:MM> ... | -
      absExpiryGMT <HH:MM> ...] [-heurExpiryParam <positive_integer>] [-
      weakNegRelExpiry <secs>] [-maxResSize <KBytes>] [-memLimit <MBytes>]
      ...
2 <!--NeedCopy-->
```

Example::

```
add cache contentGroup DEFAULT -maxResSize 500
```

Add cache policy

To add cache policy, at the command prompt, type:

```
1 add cache policy <policyName> -rule <expression> -action <action>  [-
      storeInGroup <string>] [-invalGroups <string> ...] [-invalObjects <
      string> ...] [-undefAction ( NOCACHE | RESET )] add cache policy <
      name> <rule> <profileName> [-comment <string>]  [-logAction <string
      >]
2 <!--NeedCopy-->
```

Example:

```
add cache policy ctx_doc_pdf -rule "HTTP.REQ.URL.ENDSWITH(\".pdf\")"-action
 CACHE -storeInGroup DEFAULT
```

Bind cache policy with load balancing virtual server of type HTTP/3_QUIC

To bind cache policy with load balancing virtual server of type HTTP/3_QUIC, at the command prompt, type:

```
1 bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
      ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
      positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
       REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
      -analyticsProfile <string>@)
2 <!--NeedCopy-->
```


Example:

```
bind lb vserver lb-http3 -policyName ctx_doc_pdf -priority 100 -type
REQUEST
```

Bind cache policy global to HTTP/3 global bind point

To bind a cache policy HTTP/3 global bind point:

```
1 bind cache global <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

Example:

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

For more information, see Integrated cache policy configuration.

Global built-in cache policies

After you upgrade your appliance to Citrix ADC release 13.0 build 82.x, the following cache policies will be automatically bound to the HTTP/3 default bind point.

On upgrade to the 13.0 82.x release, the following cache policies are automatically bound to the HTTP/3 default bind point.

```
1  > sho cache global -type HTTPQUIC_REQ_DEFAULT
2  1)      Policy Name: NOPOLICY
3          Priority: 185883
4          GotoPriorityExpression: USE_INVOCATION_RESULT
5          Invoke type: policylabel        Invoke name:
               _httpquicReqBuiltinDefaults
6          Global bindpoint: HTTPQUIC_REQ_DEFAULT
7
8   Done
9  > sho cache global -type HTTPQUIC_RES_DEFAULT
10 1)      Policy Name: NOPOLICY
11         Priority: 185883
12         GotoPriorityExpression: USE_INVOCATION_RESULT
13         Invoke type: policylabel        Invoke name:
               _httpquicResBuiltinDefaults
14         Global bindpoint: HTTPQUIC_RES_DEFAULT
15
16 <!--NeedCopy-->
```


After an upgrade, if the policies are not bound, you can use the following commands to manually bind and save the configuration.

```
add cache policylabel _httpquicReqBuiltinDefaults -evaluates
    HTTPQUIC_REQ

add cache policylabel _httpquicResBuiltinDefaults -evaluates
    HTTPQUIC_RES

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _nonGetReq -priority 100

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _advancedConditionalReq -priority 200

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _personalizedReq -priority 300

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableStatusRes -priority 100

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableVaryRes -priority 200

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableCacheControlRes -priority 300

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _cacheableCacheControlRes -priority 400

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheablePragmaRes -priority 500

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _cacheableExpiryRes -priority 600

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _imageRes -priority 700

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _personalizedRes -priority 800

bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
    USE_INVOCATION_RESULT -type HTTPQUIC_REQ_DEFAULT -invoke policylabel
     _httpquicReqBuiltinDefaults
```



```
28
29 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
       USE_INVOCATION_RESULT -type HTTPQUIC_RES_DEFAULT -invoke policylabel
        _httpquicResBuiltinDefaults
30
31 <!--NeedCopy-->
```

> **Note:**
>
> The first two commands in the list of commands, and the last two commands in the same list, are included for the sake of completeness. You might encounter an error when running the four commands, since the commands are already run at the time of appliance restart. But you can ignore these errors.

URL Transformation policy configuration for HTTP/3 traffic

The URL transformation modifies all URLs in designated requests from an external version seen by outside users to an internal URL seen only by your Web servers and administrators. You can redirect user requests seamlessly, without exposing your network structure to users. You can also modify complex internal URLs that users might find difficult to remember into simpler, more easily remembered external URLs.
HTTP over QUIC type virtual servers have cache policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included. New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Add URL Transform profile

To add a URL transformation profile, at the command prompt, type:

```
1 add transform profile <name> [-type URL]
2 <!--NeedCopy-->
```

Example:

```
add transform profile msapps
```

Add URL Transform action

To add URL transformation action, at the command prompt, type:


```
add transform action <name> <profileName> <priority> [-state ( ENABLED
    | DISABLED )]
<!--NeedCopy-->
```

Example:

```
add transform action docx2doc msapps 2
```

Add URL Transform action

To add URL transform action to replace URL, at the command prompt, type:

```
add transform action <name> <profileName> <priority> [-state ( ENABLED
    | DISABLED )]
<!--NeedCopy-->
```

Example:

```
add transform action docx2doc msapps 1
```

Add URL Transform policy

To add a URL transformation policy, at the command prompt, type:

```
add transform policy <name> <rule> <profileName> [-comment <string>]
   [-logAction <string>]
<!--NeedCopy-->
```

Example:

```
add transform policy urltrans_udp "CLIENT.UDP.DSTPORT.EQ(443)"msapps
```

Bind URL Transform policy with load balancing virtual server of type HTTP/3_QUIC

To bind URL transformation policy with load balancing virtual server of type HTTP/3_QUIC, at the command prompt, type:

```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
   ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
   positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
    REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
   -analyticsProfile <string>@)
<!--NeedCopy-->
```


Example:

```
bind lb vs lb-http3 -policyName urltrans_udp -type REQUEST -priority 8
```

Bind URL transform policy global with HTTP/3 QUIC based load balancing virtual server

To bind a URL transform policy HTTP/3 global bind point, at the command prompt, type:

```
1 bind transform global <policyName> <priority>  [<gotoPriorityExpression
     >] [-type <type>] [-invoke (<labelType>  <labelName>) ]
2 <!--NeedCopy-->
```

Example:

```
bind transform global urltrans_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

For more information, see URL transformation policy configuration.

Front end optimization (FEO) policy configuration for HTTP/3 traffic

The HTTP protocols that underlie web applications were originally developed to support the transmission and rendering of simple webpages. New technologies such as JavaScript and cascading style sheets (CSS), and new media types such as Flash videos and graphics-rich images, place heavy demands on front-end performance, that is, on performance at the browser level. The Citrix ADC front end optimization (FEO) feature addresses such issues and reduces the load time and render time of webpages.

> **Note:**
>
> `HTTP_QUIC _Override/Default_Request` Type is not supported for FEO policy global binding.

Add Front end optimization (FEO) action

To add a FEO action, at the command prompt, type:

```
1 add feo action <name> [-pageExtendCache] [<cacheMaxage>][-
     imgShrinkToAttrib] [-imgGifToPng] [-imgToWebp] [-imgToJpegXR] [-
     imgInline] [-cssImgInline] [-jpgOptimize] [-imgLazyLoad] [-cssMinify
     ] [-cssInline] [-cssCombine] [-convertImportToLink] [-jsMinify] [-
     jsInline] [-htmlMinify] [-cssMoveToHead] [-jsMoveToEND][-
     domainSharding <string> <dnsShards> ...] [-clientSideMeasurements]
2
3 <!--NeedCopy-->
```


Example:

```
add feo action feoact -imgGifToPng -pageExtendCache
```

Add Front end optimization (FEO) policy

To add a FEO policy, at the command prompt, type:

```
add feo policy <name> <rule> <action>
```

Example:

```
add feo policy udp_feo_img "CLIENT.UDP.DSTPORT.EQ(443)"IMG_OPTIMIZE
```

Bind FEO policy with load balancing virtual server of type HTTP/3_QUIC

To bind FEO policy with load balancing virtual server of type HTTP/3_QUIC, at the command prompt, type:

```
bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
     ) | <serviceGroupName>@ |         (-policyName <string>@  [-
    priority <positive_integer>]  [-gotoPriorityExpression <expression>]
      [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
    analyticsProfile <string>@)
<!--NeedCopy-->
```

Example:

```
bind lb vserver lb-http3 -policyName udp_feo_img -priority 4 -gotoPriorityExpression
 END -type REQUEST
```

Bind FEO policy to HTTP/3 global bind point

To bind a cache policy to the HTTP/3 global bind point, at the command prompt, type:

```
bind cache global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

Example:

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

For more information, see Front end optimization policy configuration.

SSL Policy configuration for HTTP/3 traffic

HTTP over QUIC type virtual servers have SSL policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included.
New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.
SSL policies with actions that are supported for TLSv1.3 are only applicable for HTTP/3 bind points or virtual servers.

Add SSL Policy

To add a FEO policy, at the command prompt, type:

```
1 add ssl policy <name> -rule <expression> [-action <string>] [-
     undefAction <string>] [-comment <string>]
2 <!--NeedCopy-->
```

Example:

```
add ssl policy ssl-pol -rule CLIENT.SSL.IS_SSL -action NOOP
```

Bind SSL Policy to HTTP/3 virtual server

To bind an SSL policy to the HTTP/3 virtual server, at the command prompt:

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

Example:

```
bind ssl vserver lb-http3 -policyName ssl-pol -priority 4 -type REQUEST
```

Add SSL policy with UDP expression for SSL Policy

To add an SSL policy with UDP expression, at the command prompt:

```
1 add ssl policy <name> -rule <expression> [-action <string>] [-
     undefAction <string>] [-comment <string>]
2 <!--NeedCopy-->
```

Example:


```
add ssl policy ssl_udp_clnt -rule "CLIENT.UDP.DSTPORT.EQ(443)"-action NOOP
```

Bind SSL Policy with UDP expression to HTTP/3 virtual server

To bind an SSL policy with UDP expression to the HTTP/3 virtual server, at the command prompt, type

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

Example:

```
bind ssl vs lb-http3 -policyName ssl_udp_clnt -priority 8 -type REQUEST
```

Add SSL policy for CLIENTHELLO bind point for HTTP/3 traffic

To bind SSL policy for CLIENTHELLO bind point for HTTP/3 traffic, at the command prompt, type:

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

Example:

```
add ssl policy ssl-pol-ch -rule "CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
(0x1301)"-action RESET
```

Bind SSL policy to CLIENTHELLO bind point

To bind an SSL policy to the CLIENTHELLO bind point, at the command prompt, type:

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

Example:

```
bind ssl vs lb-http3 -policyName ssl-pol-ch -type CLIENTHELLO_REQ -priority
 100
```

Bind SSL policy to HTTP/3 global bind point

To bind an SSL policy to the HTTP/3 global bind point, at the command prompt, type:


```
bind cache global <policy> -priority <positive_integer> [-gotoPriorityExpression
 <expression>] [-type <type>] [-invoke (<labelType> <labelName>)]
```

Example:

Following is an example of a DATA policy being bound to a HTTP/3 global bind point:

```
Bind ssl global -policyName ssl-pol-ch -priority 7 -type HTTPQUIC_DATA_DEFAULT
```

> **Note:**
>
> Forward action that can be set for CLIENTHELLO bind point for SSL virtual servers is currently not supported for HTTP_QUIC type virtual servers.

Application Firewall Policy configuration for HTTP/3 traffic

HTTP over QUIC type virtual servers have web application firewall policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included.
New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Add Web Application Firewall policy with UDP expression

To add Web Application Firewall policy with UDP expression, at the command prompt:

```
1 add appfw policy <name> <rule> <profileName> [-comment <string>] [-
     logAction <string>]
2 <!--NeedCopy-->
```

Example:

```
add appfw policy appfw_udp "CLIENT.UDP.DSTPORT.EQ(443)"APPFW_BYPASS
```

Bind log expressions with UDP based expression for Web Application Firewall profile

To bind log expressions with UDP for Web Application Firewall profile, at the command prompt:

Example:

```
bind appfw profile APPFW_BLOCK -logExpression logexp-1 "CLIENT.UDP.DSTPORT.
EQ(443)"
```


Bind Application Firewall policy with HTTP/3 virtual server

To bind Web Application Firewall policy with HTTP/3 virtual server, at the command prompt:

```
bind appfw policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

Example:

```
bind lb vs lb-http3 -policyName appfw_udp -priority 3 -type REQUEST
```

Bind Web Application Firewall policy to HTTP/3 global bind point

To bind a Web Application Firewall policy to the HTTP/3 global bind point, at the command prompt, type:

```
bind appfw global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

Example:

```
bind appfw global appfw_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

AppQoE Policy configuration for HTTP/3 traffic

HTTP over QUIC type virtual servers have AppQoE policy support. However, as QUIC uses UDP as its transport mechanism, TCP based expressions are excluded and UDP based expressions are included. New or existing policy configurations with TCP expressions cannot be bound to HTTP/3 virtual servers or to the newly added HTTP/3 global bind points. Instead of TCP expressions, UDP expressions can be included in the policy configurations that are bound to HTTP/3 QUIC virtual servers or HTTP over QUIC bind points.

Add AppQoE policy with UDP based expression

To add AppQoE policy with UDP expression, at the command prompt:

```
add AppQoE policy <name> <rule> <profileName> [-comment <string>] [-
    logAction <string>]
<!--NeedCopy-->
```

Example:


```
add appqoe policy appqoe-pol-udp -rule "CLIENT.UDP.DSTPORT.EQ(443)"-action
appqoe-act-basic-prhigh
```

Bind AppQoE policy with HTTP/3 virtual server

To bind the AppQoE policy with the HTTP/3 virtual server, at the command prompt, type:

```
1 bind appqoe policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

Example:

```
bind lb vs lb-http3 -policyName appqoe-pol-udp -type REQUEST -priority 3
```

Bind AppQoE policy to HTTP_QUIC virtual server

To bind AppQoE policy to `HTTP_QUIC` virtual server, at the command prompt, type:

```
1 bind appqoe <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

Example:

```
bind lb vs lb-http3 -policyName appqoe-pol-primd -priority 8 -type REQUEST
```

HTTP/3 service discovery

September 14, 2021

HTTP protocol relies on using HTTP Alternative Services for origin server to advertise the availability of an equivalent service. HTTP/3 service discovery also uses the same principle. An alternative HTTP/3 endpoint can be advertised using any one of the following methods:

- HTTP Alt-Svc response header
- HTTP/2 Alt-Svc Frame in the response
- Application Layer Protocol Negotiation (ALPN)

The alternative service advertises the usage of an HTTP Alt-Svc response header and the HTTP/2 Alt-Svc frame as HTTP/3 endpoint. Servers may serve HTTP/3 on any UDP port. An alternative service

advertisement includes an explicit port, and URLs contain either an explicit port or a default port associated with the scheme.

Clients receiving alternate service headers or frames are not bound to use them. The client, if made aware of an alternate service and if it supports the alternate service mechanism should use the appropriate alternate service advertised. In other words, a HTTP/1.1 service or a HTTP/2 service may advertise an equivalent endpoint that supports HTTP/3 protocol. The client on receiving this alternate service information can choose to establish a QUIC connection with the specified alternate service and once available, this connection can be used for any subsequent requests. If the establishment of the connection with the selected alternate service fails, the client can fall back to the original endpoint. When the client starts using the alternate service advertised, will indicate this by including a Alt-Used header.

Citrix ADC supports advertising equivalent HTTP/3 endpoints on HTTP and SSL type virtual servers.

Configure HTTP/3 service discovery

Complete the following steps to configure HTTP/3 service discovery:

1. Configure HTTP/3 alternative service endpoint by using an HTTP Alt-Svc Header
2. Configure HTTP/3 alternative service endpoint by using an HTTP/2 Alt-Svc frame
   Configure HTTP/3 alternative service endpoint by using an HTTP Alt-Svc Header
   To advertise an HTTP/3 endpoint by using a HTTP Alt-Svc Header, type the following command:

Note: The main purpose of advertising alternative service is to let user know that HTTP/3 capability can be accessed on HTTP/1.1 or HTTP/2 service on a.b.c.d:443 also.

```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

Example:

```
add ns httpProfile http-profile -altsvc ENABLED -altSvcValue  "h3
   -29=\":443\"; ma=3600; persist=1"
<!--NeedCopy-->
```

or

```
set ns httpProfile http-custom -altsvc ENABLED -altSvcValue  "h3
   -29=\":443\"; ma=3600; persist=1"
<!--NeedCopy-->
```

Configure HTTP/3 alternative service endpoint by using an HTTP/2 Alt-Svc frame

To advertise an HTTP/3 endpoint by using a HTTP/2 Alt-svc frame, type the following command:


```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ] -
    http2AltSvcFrame [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

Example:

```
add ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -http2AltSvcFrame
 ENABLED -altsvc ENABLED -altSvcValue "h3-29=\":443\"; ma=3600; persist=1"
```

or

```
set ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -http2AltSvcFrame
 ENABLED -altsvc ENABLED -altSvcValue "h3-29=\":443\"; ma=3600; persist=1"
```

Configure HTTP/3 alternative service with HTTP Alt-Svc header value by using GUI

1. Navigate to **System > Profiles > HTTP Profiles**.
2. Click **Add**.
3. In **Create HTTP Profile** page, go to HTTP/3 section and select **Alternative Service** checkbox.
4. The system displays the **Alternative Service Value** text box in the http2 section.
5. Enter the alternative service value as “h3-29=”:443”; ma=3600; persist=1”
6. Click **OK** and **Close**.

gRPC

September 14, 2021

gRPC in a Citrix ADC appliance is a lightweight, high performance, and open-source universal Remote Procedure Call (RPC) framework. The framework is optimal to work across multiple languages running on any operation system. Also when compared to other protocols, gRPC offers better performance and security.

gRPC for Citrix ADC is preferred for the following reasons:

- Build distributed applications for data-center and public/private cloud infrastructure.
- Provide client-server communication for mobile, web, or cloud.
- Access cloud services and applications
- Microservice deployments

Why gRPC in Citrix ADC

gRPC in Citrix ADC is implemented over HTTP/2 to support highly performance and scalable APIs. The use of binary than text keeps the payload compact and efficient. In Citrix ADC, the HTTP/2 requests are multiplexed over a single TCP connection, allowing multiple concurrent messages to be in flight without compromising the network resource usage. It also uses header compression to reduce the size of requests and responses.

gRPC supports the following types of service methods for a client to remotely invoke parameters and return types.

1. **Unary RPC**. Client sends a single request to the gRPC server and gets a single response back.

   Example:

   ```
   rpc SayHello(HelloRequest)returns (HelloResponse);
   ```

2. **Server streaming RPC**. Client sends a single request to the gRPC server and gets a stream response.

   Example:

   ```
   rpc StreamingResponse(HelloRequest)returns (HelloResponse);
   ```

3. **Client streaming RPC**. Client sends a sequence of messages and waits for the server to read and return its response.

   Example:

   ```
   rpc IntroduceYourself(stream HelloRequest)returns (HelloResponse)
   ```

4. **Bidirectional streaming RPC**. Both client and server from both sides send a stream of messages using the read-write stream. The two streams operate independently.

   Example:

   ```
   rpc ChatSession (stream HelloRequest)returns (stream HelloResponse)
   ```

Citrix ADC supports the following capabilities for its services with gRPC endpoints:

- Load balancing
- Content switching
- Secure end-point services like Web Application Firewall, authentication .
- Policy configuration

- Stats and logging
- Content rewriting, content filtering
- Layer 4 and Layer 7 optimizations, TLS offering
- Gateway solutions for protocol translations

gRPC end-to-end configuration

September 14, 2021

The gRPC end-to-end configuration works by sending a gRPC request from a client over HTTP/2 protocol and again forwarding gRPC messages responded by the gRPC server.

How end-to-end gRPC configuration works

The following diagram shows a gRPC configuration works in a Citrix ADC appliance.

1. To deploy the gRPC configuration, you must first enable HTTP/2 in the HTTP profile and also enable HTTP/2 support globally on server side.
2. When a client sends a gRPC request, the load balancing virtual server evaluates the gRPC traffic using policies.
3. Based on policy evaluation, the load balancing virtual server (with gRPC service bound to it) terminates the request and forwards it as a gRPC request to the back-end gRPC server.
4. Similarly, when the gRPC server responds to the client, the appliance terminates the response and forwards it as a gRPC response to the client.

Example for gRPC request sent to gRPC server

The request header is sent as HTTP/2 headers in HEADERS+CONTINUATION Frames.

```
1 ```
```



```
HEADERS (flags = END_HEADERS)
: method = POST
: scheme = http
: path = /helloworld.citrix-adc/SayHello
: authority = 10.10.10.10.:80
grpc-timeout = 15
content-type = application/grpc+proto
grpc-encoding = gzip
DATA (flags = END_STREAM)
<Length-Prefixed Message>
<!--NeedCopy--> ```
```

Example for gRPC response header from gRPC server to Citrix ADC appliance

Response-Headers & Trailers-Only are delivered in a single HTTP/2 HEADERS frame block. Most responses are expected to have both headers and trailers but Trailers-Only is permitted for calls that produce an immediate error. Status must be sent in Trailers even if the HTTP status code is OK.

```
```
HEADERS (flags = END_HEADERS)
: status = 200
Grpc-encoding= gzip
Content-type = application/grpc+proto
DATA
<Length-Prefixed Message>
HEADERS (flags = END_STREAM, END_HEADERS)
grpc-status = 0 # OK

<!--NeedCopy--> ```
```

### Configure gRPC by using the CLI

To configure an end-to-end gRPC deployment, you must complete the following:

- Add HTTP profile with HTTP/2 and HTTP/2 direct enabled.
- Enable global back end HTTP/2 support in HTTP parameter
- Add load balancing virtual server of type SSL/HTTP and set HTTP profile
- Add Service for gRPC endpoint and set HTTP profile
- Bind gRPC end point service to load balancing virtual server

 
```

Add HTTP profile with HTTP/2 and HTTP/2 direct enabled

You must enable HTTP/2 and HTTP/2 direct parameters in the HTTP profile. Also, you must enable the HTTP/2 direct parameter if gRPC over HTTP/2 cleartext is required.

At the command prompt, type:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

Example:

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

Enable global back-end HTTP/2 support through HTTP parameter

To enable HTTP/2 support globally on the server side by using the Citrix ADC command line.

At the command prompt, type:

```
set ns httpParam -http2ServerSide( ON | OFF )
```

Example:

```
set ns httpParam -http2ServerSide ON
```

Add load balancing virtual server of type SSL/HTTP and set HTTP profile

To add a load balancing virtual server by using the **Citrix ADC** command interface:

At the command prompt, type:

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

Example:

```
add lb vserver lb-grpc HTTP 10.10.10.11 80 -httpProfileName http2gRPC
```

> **Note:**
>
> If you are using a load balancing virtual server of type SSL, then you must bind the server certificate. See Bind server certificate topic for more information.

Add Service for gRPC endpoint and set HTTP profile

To add a gRPC service with HTTP profile by using the **Citrix ADC** command interface:
At the command prompt, type:


```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

Example:

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

Bind gRPC end point service to load balancing virtual server

To bind a gRPC service to load balancing virtual server by using the **Citrix ADC** command interface:

At the command interface, type:

```
bind lb vserver <name> <serviceName>
```

Example:

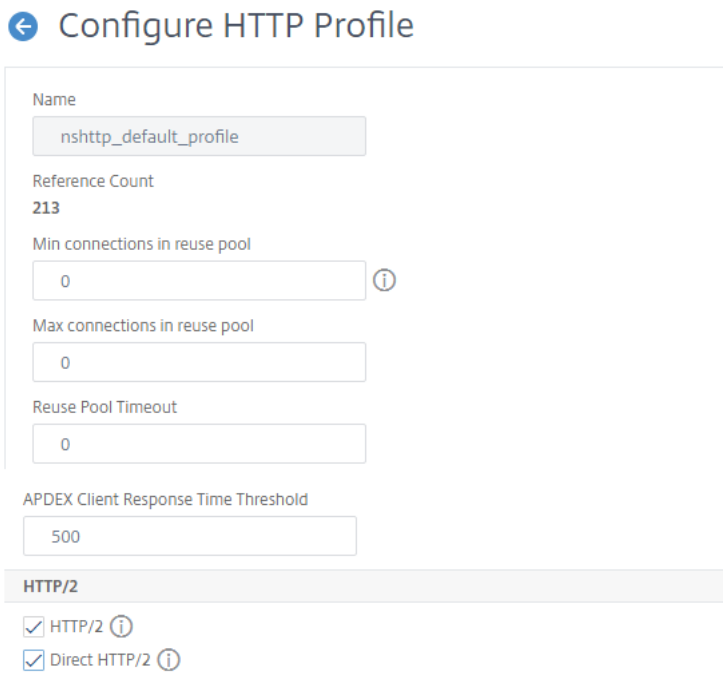
```
bind lb vserver lb-grpc svc-grpc
```

Configure end-to-end gRPC deployment by using the GUI

Complete the following steps to configure gRPC by using the GUI.

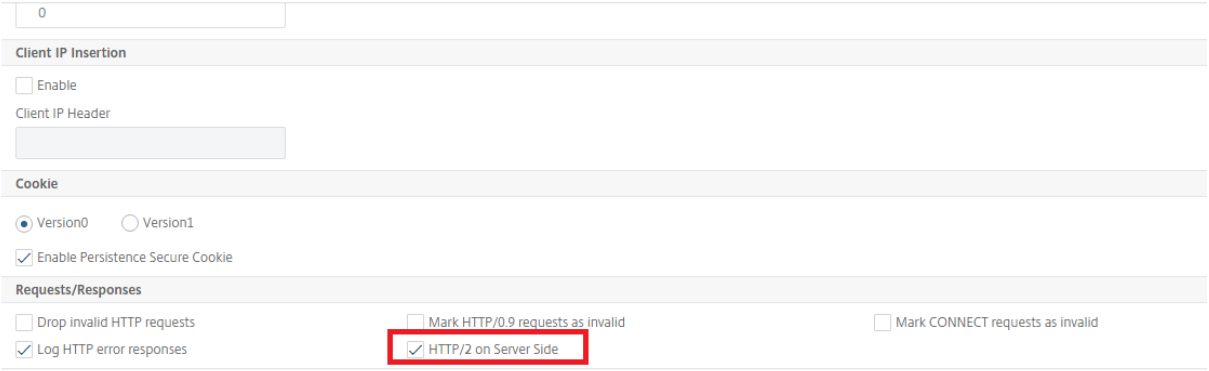
Add HTTP profile with HTTP/2 and HTTP/2 direct enabled

1. Navigate to **System > Profiles** and click **HTTP Profiles**.
2. Enable HTTP/2 option in a new HTTP profile or existing HTTP profile



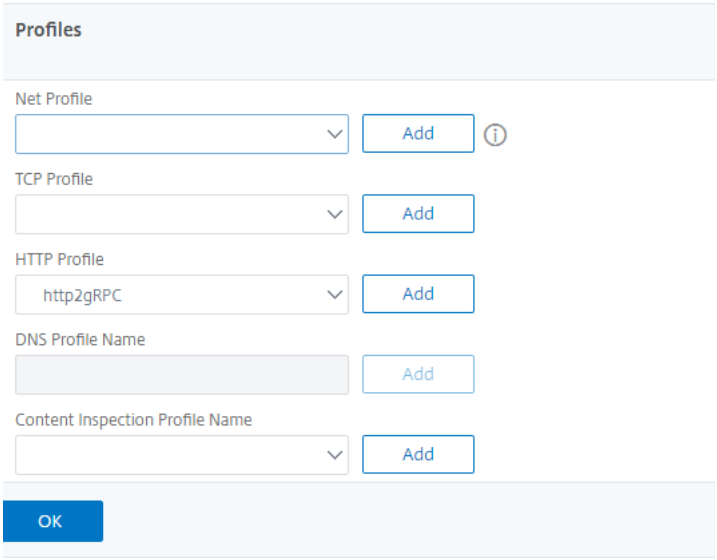
Enable global back end HTTP/2 support in HTTP parameter

1. Navigate to **System > Settings > HTTP Parameters**.
2. In the Configure HTTP Parameter page, select HTTP/2 on Server Side.
3. Click **OK**.



Add load balancing virtual server of type SSL/HTTP and set HTTP profile

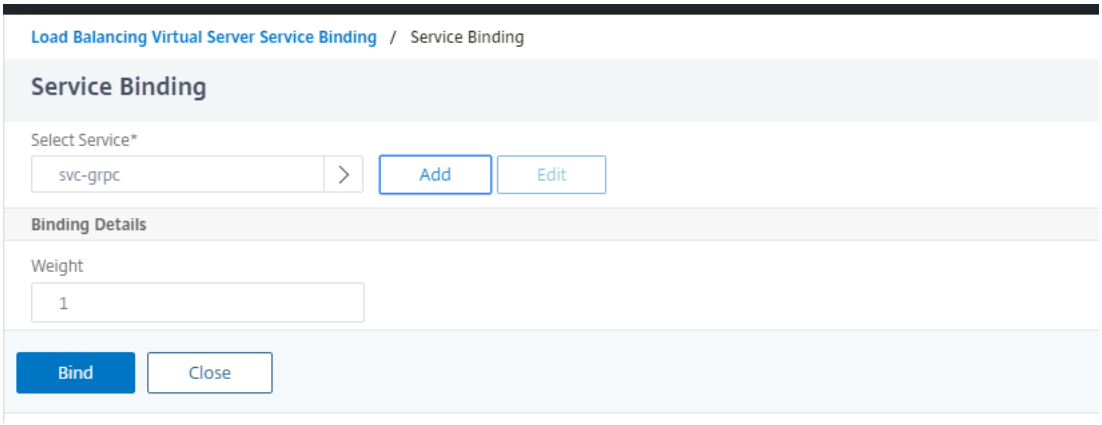
1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Click Add to create a load balancing virtual server for gRPC traffic.
3. In Load Balancing Virtual Server page, click Profiles.
4. In the Profiles section, select the profile type as HTTP.
5. Click OK and then Done.



Add Service for gRPC endpoint and set HTTP profile

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Click Add to create an application server for gRPC traffic.

3. In Load Balancing Service page, go to Profile section.
4. Under Profiles, add HTTP profile for gRPC endpoint.
5. Click OK and then Done.



For detailed GUI procedures related to loading balancing, see Load Balancing topic.

gRPC bridging

September 14, 2021

When a client sends a request over HTTP/1.1 protocol, the Citrix ADC appliance supports bridging of the gRPC requests over HTTP/1.1 protocol which is in compliance with the gRPC server over HTTP/2 protocol. Similarly, in reverse bridging, the appliance receives the client gRPC request over the HTTP/2 protocol and performs reverse bridging for the gRPC requests in compliance with the gRPC server of the HTTP/1.1 protocol.

How gRPC bridging works

In this scenario, the Citrix ADC appliance seamlessly bridges gRPC content received on an HTTP/1.1 connection and forwards it to the back-end gRPC server over HTTP/2.

The following diagram shows how components interact with each other in a gRPC bridging configuration.

1. When a gRPC request is sent, the Citrix ADC appliance checks if the connection is HTTP/1.1 and the content type is application/grpc. The HTTP/1.1 requests translate to the following pseudo headers.
2. On receiving a gRPC request on HTTP/1.1. connection as indicated by the Content-Type header, the ADC appliance transforms the request into gRPC over HTTP/2 as given below:

```
1     :method: Method-name in HTTP/1.1 request
2     :path: Path is HTTP/1.1 request
3     content-type: application/grpc
4 <!--NeedCopy-->
```

1. Based on policy evaluation, the load balancing virtual server (with the gRPC service bound to it) terminates the request or forwards it over HTTP/2 frames to the back-end gRPC server.
2. On receiving the response on a HTTP/2 connection from the gRPC server, the appliance buffers until it receives the HTTP/2 trailer and then checks for the gRPC-status code. If it is non-zero gRPC error status, the appliance looks for the mapping HTTP Status code and send a suitable HTTP/1.1 error response.

Configure gRPC bridging by using the CLI

To configure gRPC bridging, you must complete the following steps:

1. Add HTTP profile with HTTP/2 and HTTP/2 direct enabled
2. Enable global back-end HTTP/2 support in the HTTP parameter
3. Add load balancing virtual server of type SSL/HTTP and set the HTTP profile
4. Add Service for gRPC endpoint and set the HTTP profile
5. Bind gRPC end point service to load balancing virtual server
6. Map gRPC status code to the HTTP response for non-zero gRPC status
7. Configure gRPC buffering by time and/or size

Add HTTP profile with the HTTP/2 and HTTP/2 direct enabled

To begin the configuration, you must enable the HTTP/2 feature in the HTTP profile. If the client sends the HTTP 1.1 requests, the appliance bridges the request and forward it to the back-end server.

At the command prompt, type:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

Example:


```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

Enable global back end HTTP/2 support in the HTTP parameter

To enable the HTTP/2 support globally on the server side by using the Citrix ADC command line.

At the command prompt, type:

```
set ns httpParam -http2ServerSide( ON | OFF )
```

Example:

```
set ns httpParam -http2ServerSide ON
```

Add load balancing virtual server of type SSL/HTTP and set the HTTP profile

To add a load balancing virtual server by using the **Citrix ADC** command interface

At the command prompt, type:

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

Example:

```
add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName http2gRPC
```

> **Note:**
>
> If you are using a load balancing virtual server of type SSL, then you must bind the server certificate. See Bind server certificate topic for more information.

Add Service for gRPC endpoint and set the HTTP profile

To add a gRPC service with the HTTP profile by using the **Citrix ADC** command interface.

At the command prompt, type:

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

Example:

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

Bind gRPC end point service to load balancing virtual server

To bind a gRPC end point service to the load balancing virtual server by using the CLI.

At the command interface, type:


```
bind lb vserver <name> <serviceName>
```

Example:

```
bind lb vserver lb-grpc svc-grpc
```

Map gRPC status code to HTTP status-code in the HTTP/1.1 response

In gRPC bridging scenario, the gRPC service responds to the request with a gRPC status-code. The appliance maps the gRPC status code to a corresponding HTTP response code and reason phrase. The mapping is done based on the table provided below. The Citrix ADC appliance when sending the HTTP/1.1 response to the client sends the HTTP status code and the reason phrase.

gRPC status-code	HTTP response status-code	HTTP response reason-phrase
OK = 0	200	OK
CANCELLED = 1	499	*
UNKNOWN = 2	500	Internal Server Error
INVALID_ARGUMENT = 3	400	Bad Request
DEADLINE_EXCEEDED = 4	504	Gateway Timeout
NOT_FOUND = 5	404	*
ALREADY_EXISTS = 6	409	Conflict
PERMISSION_DENIED = 7	403	Forbidden
UNAUTHENTICATED = 16	401	Unauthorized
RESOURCE_EXHAUSTED = 8	429	*
FAILED_PRECONDITION = 9	400	Bad Request
ABORTED = 10	409	Conflict
OUT_OF_RANGE = 11	400	Bad Request
UNIMPLEMENTED = 12	501	Not Implemented
INTERNAL = 13	500	Internal Server Error
UNAVAILABLE = 14	503	Service Unavailable
DATA_LOSS = 15	500	Internal Server Error

Configure gRPC buffering by time and/or size

The Citrix ADC appliance buffers the gRPC response from the back-end server until the response trailer is received. This breaks bi-directional gRPC calls. Also, if the gRPC response is huge, it consumes a significant amount of memory to buffer the response completely. To resolve the issue, the gRPC bridging configuration is enhanced to limit buffering by time and/or size. If the buffer size or time limit exceeds threshold, the appliance stops buffering and forwards the response to the client even when any one of the limitations triggers (either the trailer is not received within the configured buffer size or if the configured timeout occurs). As a result, the configured policies and its expressions (based on grpc-status code) do not work as expected.

To limit gRPC buffering by time and/or size by the CLI, you can configure when you add a new HTTP profile or configure when you modify an existing profile.

At the command prompt, type:

```
add ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout
 <positive_integer>]
```

Or

```
set ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout
 <positive_integer>]
```

Where,

`grpcholdlimit`. Maximum size in bytes allowed to buffer gRPC packets until trailer is received. You can configure both the parameters and any one.

Default value: 131072
Minimum value: 0
Maximum value: 33554432

`grpcholdtimeout`. Maximum time in milliseconds allowed to buffer gRPC packets until trailer is received. The value should be in multiples of 100.
Default value: 1000
Minimum value: 0
Maximum value: 180000

Example:

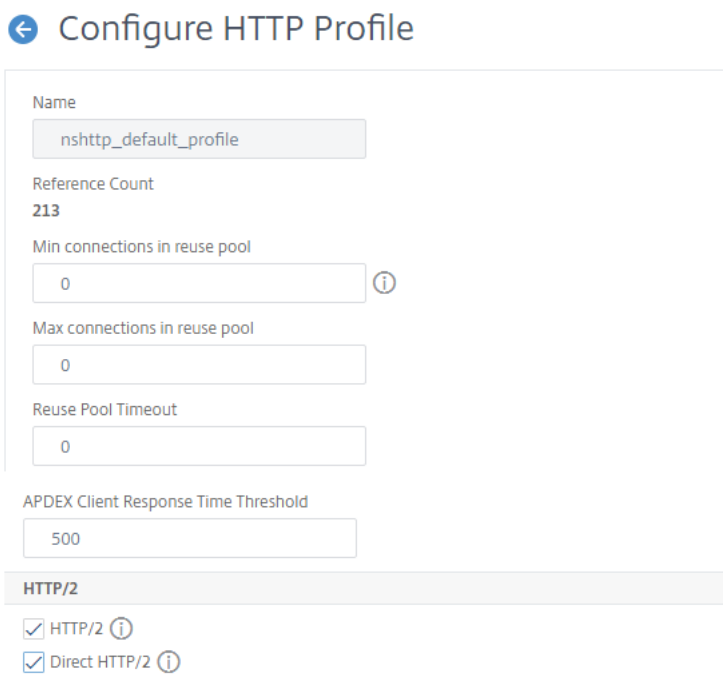
```
add httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
set httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
```

Configure gRPC bridging by using the GUI

Complete the following steps to configure gRPC bridging by using the Citrix ADC GUI.

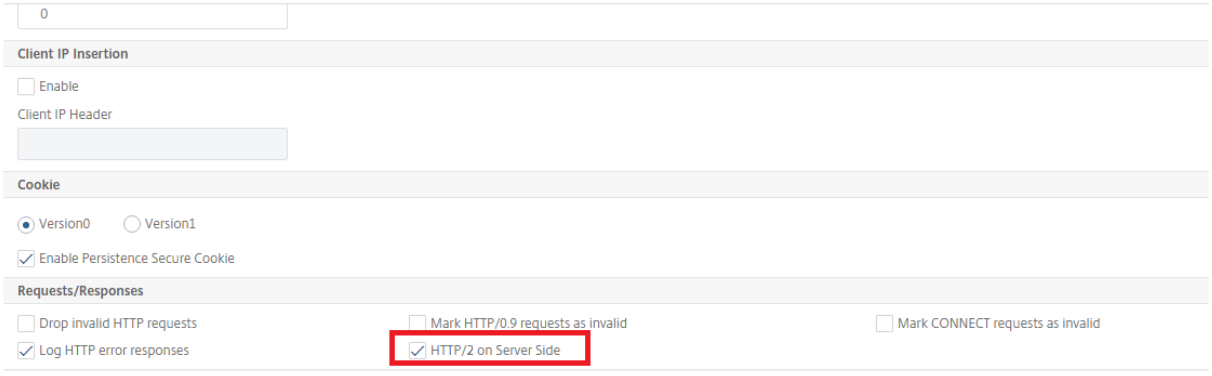
Add HTTP profile with HTTP/2 and HTTP/2 direct enabled

1. Navigate to **System > Profiles** and click **HTTP Profiles**.
2. Select **HTTP/2** in the HTTP profile.



Enable global back-end HTTP/2 support in the HTTP parameter

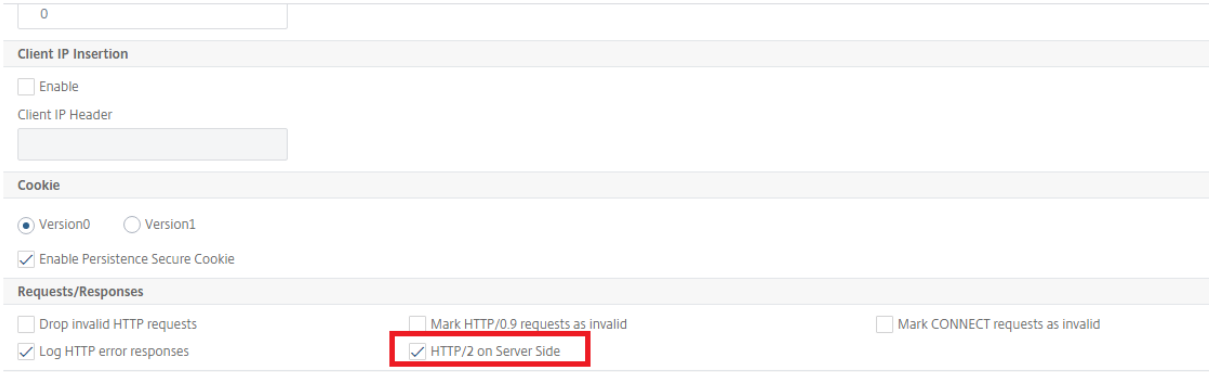
1. Navigate to **System > Settings > HTTP Parameters**.
2. In the **Configure HTTP Parameter** page, select **HTTP/2 on Server Side** option.
3. Click **OK**.



Add load balancing virtual server of type SSL/HTTP and set HTTP profile

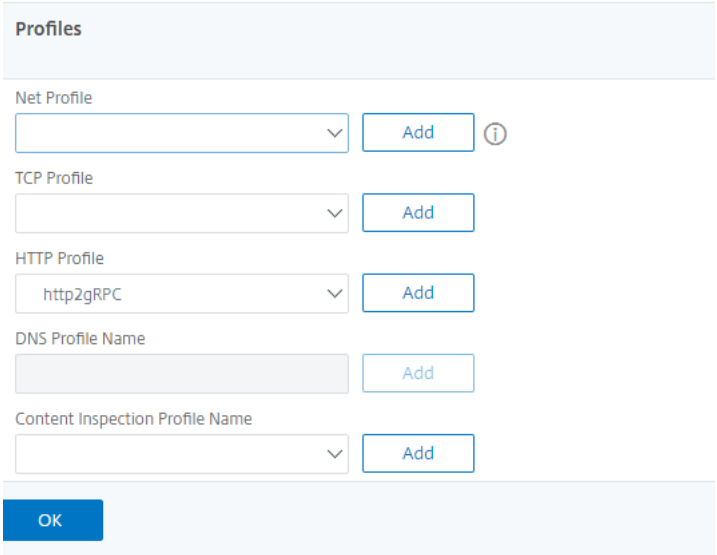
1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.

2. Click **Add** to create a load balancing virtual server for gRPC traffic.
3. In **Load Balancing Virtual Server** page, click **Profiles**.
4. In the **Profiles** section, select the profile type as HTTP.
5. Click **OK** and then **Done**.



Add Service for gRPC endpoint and set HTTP profile

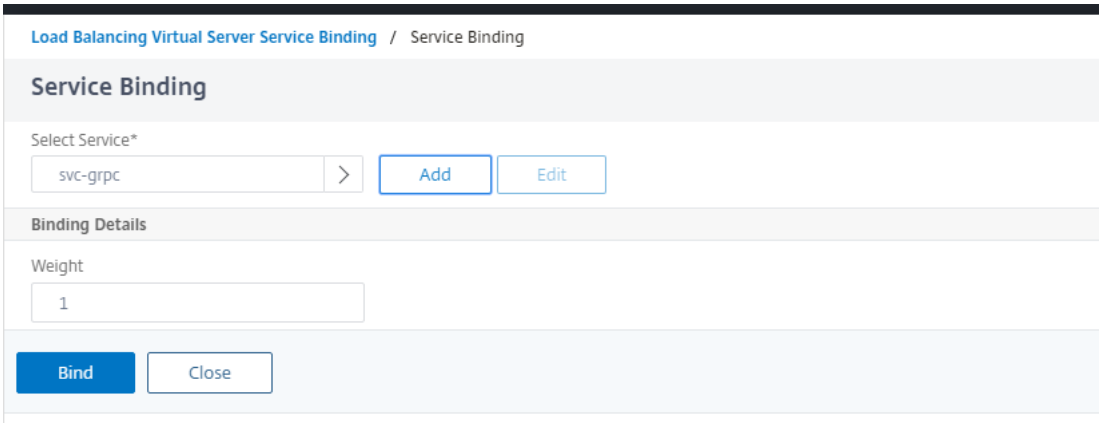
1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Click **Add** to create an application server for gRPC traffic.
3. In **Load Balancing Service** page, go to **Profile** section.
4. Under **Profiles**, add **HTTP profile** for gRPC endpoint.
5. Click **OK** and then **Done**.



Bind Service for gRPC endpoint to load balancing virtual server

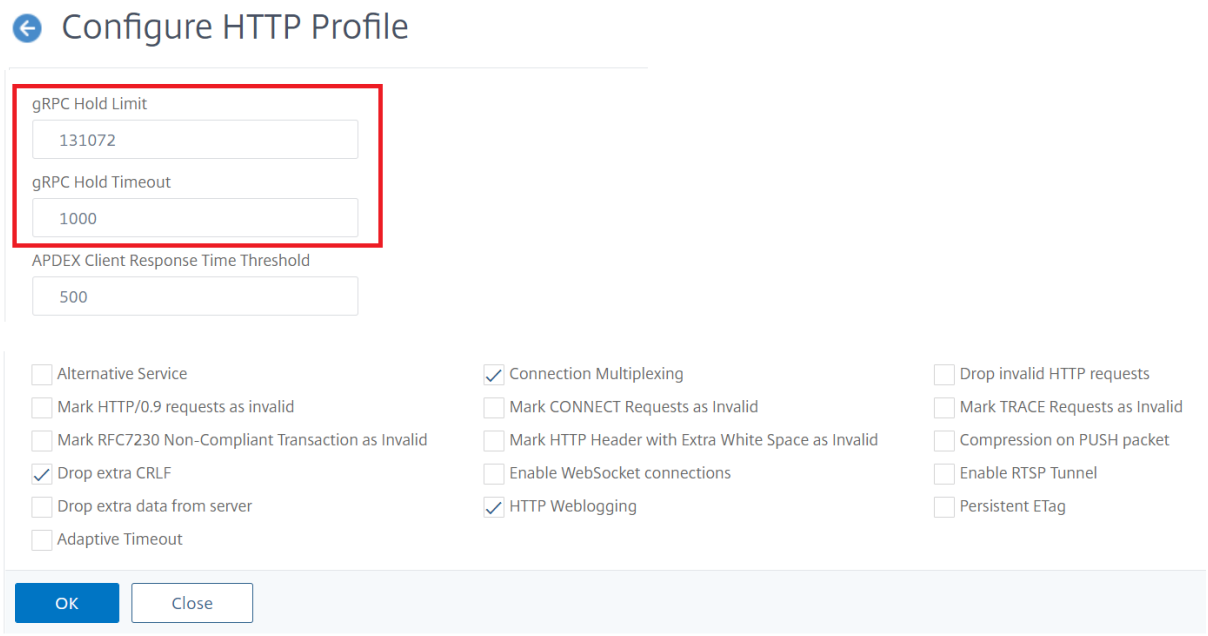
1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Click **Add** to create a load balancing virtual server for gRPC traffic.

3. In **Load Balancing Virtual Server** page, click **Service and Service Groups** section.
4. In the **Load Balancing Virtual Server Service Binding** page, select the gRPC service to bind.
5. Click **Close** and then **Done**.



Configure gRPC buffering by time and size by using the GUI

1. Navigate to **System > Profiles** and click **HTTP Profiles**.
2. Select **HTTP/2** in the HTTP profile.
3. In the **Configure HTTP Profile** page, set the following parameters:
   a) grpcHoldTimeout. Enter the time in milliseconds to buffer gRPC packets until the trailer is received.
   b) grpcHoldLimit. Enter the maximum size in bytes to buffer gRPC packets until the trailer is received.
4. Click **OK** and **Close**.



For detail GUI procedures for binding service and load balancing virtual servers, see Load Balancing topic.

gRPC reverse bridging

September 14, 2021

In this scenario, the Citrix ADC appliance seamlessly bridges gRPC content received on an HTTP/2 connection and forwards it to the back-end gRPC server over HTTP/1.1.

How reverse bridging works

The following diagram shows how components interact with each other in a gRPC bridging configuration.

1. Client sends a gRPC request on HTTP/2 connection with gRPC headers in HTTP/2 frames and proto-buf payload.
2. Based on policy evaluation, the load balancing virtual server (with gRPC service bound to it) translates and forwards the request over HTTP/1.1 connection to backend server.
3. On receiving the HTTP/1.1 response, if there is no grpc-status code in the response, ADC derives a grpc status-case from the HTTP response code.
4. The appliance then inserts the gRPC headers into HTTP/2 trailer before forwarding the response to the client.

Configure gRPC reverse bridging by using the CLI

To configure gRPC reverse bridging, you must complete the following steps:

- Add HTTP profile 1 with HTTP/2 and HTTP/2 direct enabled for load balancing virtual server
- Add HTTP profile 2 with HTTP/2 disabled for back-end server
- Add load balancing virtual server of type SSL/HTTP and set to HTTP profile 1
- Add service for gRPC endpoint and set to HTTP profile 2
- Bind Service for gRPC endpoint to load balancing virtual server
- Map HTTP-status code to gRPC status code if the response does not have a grpc status code

Add HTTP profile 1 with HTTP/2 and HTTP/2 direct enabled for load balancing virtual server

To begin the reverse bridging configuration, you must add two HTTP profiles. One profile for enabling HTTP/2 for gRPC client requests and another profile for disabling HTTP/2 for non-gRPC server response.

At the command prompt, type:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

Example:

add ns httpProfile profile1 –http2 ENABLED -http2Direct ENABLED

Add HTTP profile 2 with HTTP/2 disabled for back-end server

To disable HTTP/2 support on the HTTP profile for back-end server response by using the Citrix ADC command line.

At the command prompt, type:

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```


Example:

add ns httpProfile profile2 –http2 DISABLED http2Direct DISABLED

Add load balancing virtual server of type SSL/HTTP and set to HTTP profile 1

To add a load balancing virtual server by using the Citrix ADC command interface.

At the command prompt, type:

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

Example:

add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName profile1

> **Note:**
>
> If you are using a load balancing virtual server of type SSL, then you must bind the server certificate. See Bind server certificate topic for more information.

Add service for gRPC endpoint and set to HTTP profile 2

To add a service with gRPC endpoint and set HTTP profile 2 by using the Citrix ADC command interface.

At the command prompt, type:

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

Example:

```
add service svc-grpc 10.10.10.11 HTTP 80 -httpProfileName profile2
```

Bind service for gRPC endpoint to load balancing virtual server

To bind a gRPC service to load balancing virtual server by using the Citrix ADC command interface.

At the command interface, type:

```
bind lb vserver <name> <serviceName>
```

Example:

```
bind lb vserver lb-grpc svc-grpc
```


Map HTTP response code to gRPC status code

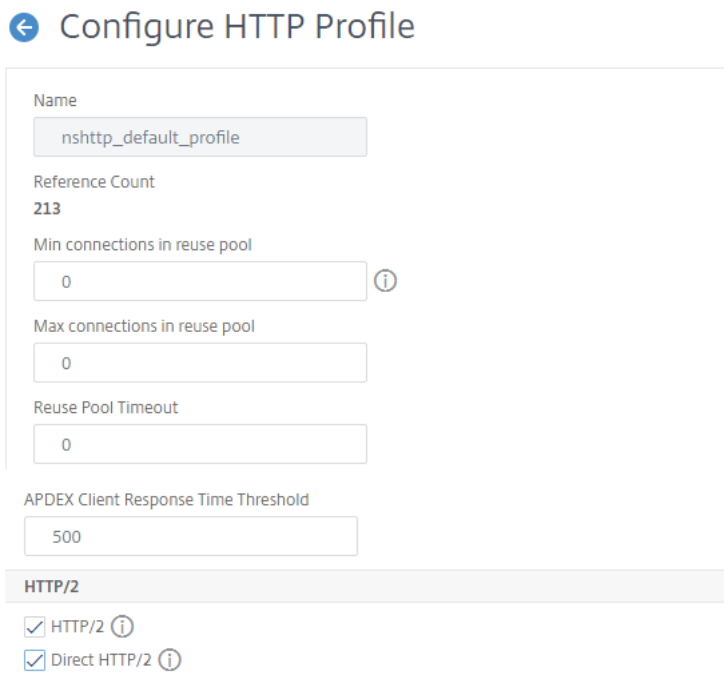
If the server does not generate a gRPC status code, the Citrix ADC appliance generates a suitable gRPC status code based on the HTTP response received. The status codes are listed in the below mapping table.

HTTP Response status-code	gRPC status code
200	OK
400	INTERNAL = 13
403	PERMISSION_DENIED = 7
401	UNAUTHENTICATED = 16
429, 502, 503, 504	UNAVAILABLE = 14
404	UNIMPLEMENTED = 12

Configure gRPC reverse bridging by using the GUI

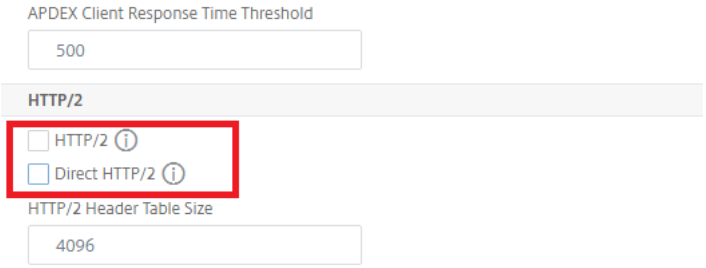
Add HTTP profile 1 with HTTP/2 and HTTP/2 direct enabled for load balancing virtual server

1. Navigate to System > Profiles and click HTTP Profiles.
2. Enable HTTP/2 option in a HTTP profile 1.



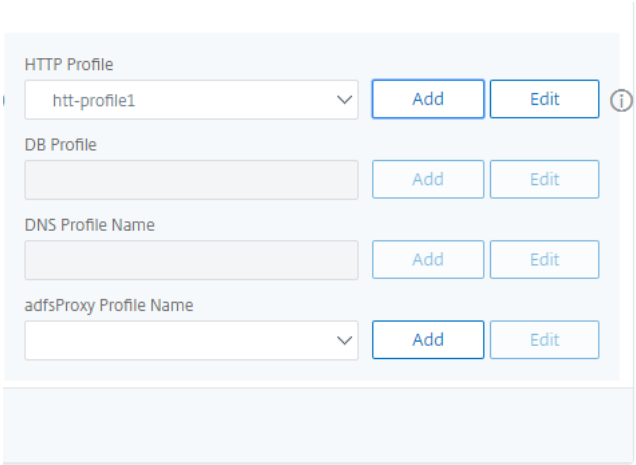
Add HTTP profile 2 with HTTP/2 disabled for back-end server

1. Navigate to **System > Profiles** and click **HTTP Profiles**.
2. Enable **HTTP/2** option in a HTTP profile 2.
3. Click **OK**.



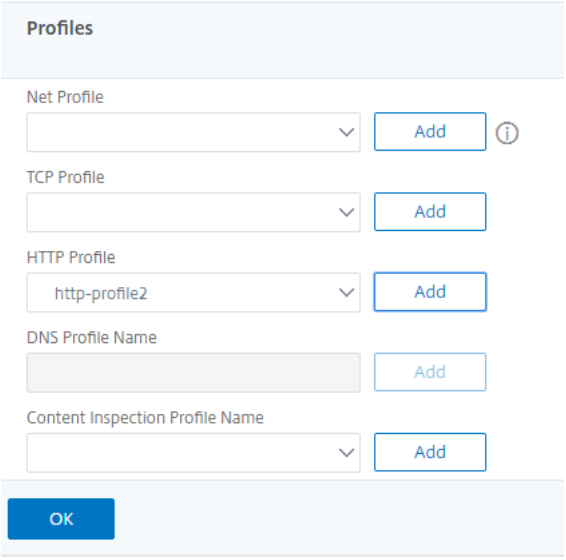
Add load balancing virtual server of type SSL/HTTP and set to HTTP profile 1

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Click **Add** to create a load balancing virtual server for gRPC traffic.
3. In **Load Balancing Virtual Server** page, click **Profiles**.
4. In the **Profiles** section, select the profile type as HTTP.
5. Click **OK** and then **Done**.



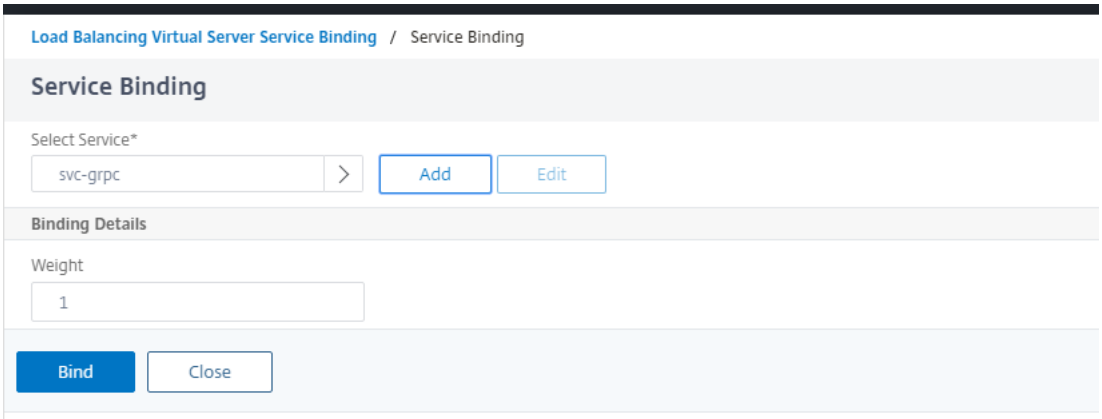
Add service with gRPC endpoint and set to HTTP profile 2

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Click **Add** to create an application server for gRPC traffic.
3. In **Load Balancing Service** page, go to **Profile** section.
4. Under **Profiles**, add **HTTP profile** for gRPC endpoint.
5. Click **OK** and then **Done**.



Bind Service for gRPC endpoint to load balancing virtual server

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Click **Add** to create a load balancing virtual server for gRPC traffic.
3. In **Load Balancing Virtual Server** page, click **Service** and **Service Groups** section.
4. In the **Load Balancing Virtual Server Service Binding** page, select the gRPC service to bind.
5. Click **Close** and then **Done**.



For detail GUI procedures, see Load Balancing topic.

gRPC call termination

September 14, 2021

When a Citrix ADC appliance has policies such as rate limiting, Web App Firewall security configured

and if a policy evaluates to true, the appliance can terminate the call and respond with a computable gRPC error message to the client.

gRPC with rewrite policy

September 14, 2021

The gRPC with rewrite policy use case explains how Citrix ADC appliance works in rewriting some information in the gRPC requests or responses. The following diagram shows the components interact.

The following diagram shows how components interact with each other in a gRPC with rewrite policy configuration.

1. Enable rewrite feature on the appliance.
2. Configure rewrite action to modify, add, or delete gRPC headers.
3. Configure rewrite policy for determining the gRPC requests (traffic) on which an action has to be taken.
4. Bind the rewrite policy to the load balancing virtual server to examine if the traffic matches the policy expression.
5. By using a rewrite policy, you can perform the following based on gRPC status code.
    a) Modify responses from gRPC web server.
    b) Modify, add, or delete gRPC headers.
    c) Modify the URL of the request to the gRRC server.

Configure gRPC call termination with rewrite policy

To configure gRPC call termination with rewrite policy, you must complete the following steps:

1. Enable rewrite feature
2. Add rewrite policy
3. Bind rewrite policy to load balancing virtual server

Enable rewrite feature

To use the rewrite feature, you must first enable it.

At the command prompt, type:

```
enable ns rewrite
```

Add rewrite policy

After you configure a rewrite action, you must next configure a rewrite policy to select the gRPC requests to which the Citrix ADC appliance must rewrite.

At the command prompt, type:

```
add rewrite policy <name> <expression> <action> [<undefaction>]-appFlowaction
 <actionName>
```

Example:

```
add rewrite policy grpc-rewr_pol1 "http.res.header(\"grpc-status\").NE
(\"0\")"RESET
```

Bind rewrite policy to load balancing virtual server

To put a policy into effect, you must bind it to the load balancing virtual server with the gRPC service.

At the command prompt, type:

```
bind rewrite global <policyName> <priority> [<gotoPriorityExpression> [-
type <type>] [-invoke (<labelType> <labelName>)]
```

Example:

```
bind lb vserver lb-grpc -policyName grpc-rewr_pol1 -priority 100
```

gRPC with the responder policy

September 14, 2021

The gRPC with responder policy configuration explains how a Citrix ADC appliance provides different responses to gRPC requests over the HTTP/2 protocol. When users request a website home page, you might want to provide a different home page depending on where each user is located or the browser the user is using.

The following diagram shows the components that interact.

1. Enable the responder feature on the appliance.
2. Configure the responder action to generate a custom response, redirect a request to a different webpage, or reset a connection.
3. Configure the responder policy for determining the gRPC requests (traffic) on which an action has to be taken.
4. Bind the responder policy to the load balancing virtual server to examine if the traffic matches the policy expression.
5. By using a responder policy, you can perform the following based on the gRPC status code.

Configure gRPC call termination with the responder policy by using the CLI

To configure gRPC call termination with the responder policy, you must complete the following steps:

1. Enable the responder feature
2. Add a responder action
3. Add a responder policy and associate responder action
4. Bind the responder policy to load balancing virtual server

Enable the responder feature

To use the responder feature, you must first enable it.

At the command prompt, type:

```
enable ns responder
```

Add the responder action

After enabling the feature, you must configure the responder action for handling the gRPC response based on the status code returned by the back-end server.

At the command prompt, type:

```
add responder action <name> <type>
```


Example:

```
add responder action grpc-act respondwith "HTTP/1.1 200 OK\r\nServer: NS
-Responder\r\nContent-Type:application/grpc\r\ngrpc-status: 12\r\ngrpc
-message: Not Implemented\r\n\r\n"+ "Method: "+ HTTP.REQ.URL+ "is not
implemented."
```

Adding responder policy

After you configure a responder action, you must next configure a responder policy to select the gRPC request to which the Citrix ADC appliance must respond.

At the command prompt, type:

```
add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction
 <actionName>
```

Example:

```
add responder policy grpc-resp-pol1 HTTP.REQ.URL.NE( “/helloworld.Greeter/
SayHello” )grpc-act
```

Bind responder policy to load balancing virtual server

To put a policy into effect, you must bind it to the load balancing virtual server with the gRPC service.

At the command prompt, type:

```
bind responder global <policyName> <priority> [<gotoPriorityExpression> [-
type <type>] [-invoke (<labelType> <labelName>)]
```

Example:

```
bind lb vserver lb-grpc svc-grpc -policyName grpc-resp-pol1 -priority 100
```

For more information about the responder policy, see Responder Policy topic.

Policy expressions for matching gRPC protocol buffer fields

The Citrix ADC appliance supports the following policy expressions in the gRPC configuration:

- **gRPC protocol buffer field access.** The arbitrary gRPC API call matches the message field number with the new policy expressions. In a PI configuration, the matches are done using only the ‘field numbers’ and ‘API path’.
- **gRPC header filtering.** The “HttpProfile” parameters for gRPC is used to adjust the default behavior of gRPC parsing (including gRPC policy expressions). The following parameters are applicable to gRPC policy expressions:

- **gRPCLengthDelimitation.** It is enabled by default and expects the protocol buffers to be presented with a length delimited message.
- **gRPCHoldLimit.** The default value is 131072. It is the maximum protocol buffer message size in bytes. It is also the maximum string length and the maximum 'byte' field length as well.

Configure gRPC advance policy expressions by using the CLI

At the command prompt, type:

```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED ) -
    gRPCLengthDelimitation ( ENABLED | DISABLED ) -gRPCHoldLimit <int>
```

Example:

```
set ns httpProfile http2gRPC -http2 ENABLED -gRPCLengthDelimitation
    ENABLED -gRPCHoldLimit 131072
```

Configure gRPC header filtering parameters by using the GUI

1. Navigate to **System > Profiles** and click **HTTP Profiles**.
2. On the **Create HTTP Profile** page, scroll down to **HTTP/3** section, select **gRPC Length Delimitation**.

The following policy expression example shows a value in message 5, submessage 4, and field 3. It is a 32-bit int equal to 2.

```
http.req.body(1000).grpc.message(5).message(4).int32(3).eq(2)
```

The following policy expressions are added for matching gRPC protocol buffer message fields by number:

- message
- double
- float
- int32
- int64
- uint32
- uint64
- sint64
- sint32
- fixed32

- fixed64
- sfixed32
- sfixed64
- bool
- string
- enum
- bytes

API path matching

The API path matching is used to match the correct gRPC API call when more than one API is used. Match the API path, which can be found in the ': path' pseudo header in the HTTP request.

Example:

```
http.req.header(":path").eq("acme.inventory.v1/ListBooks")
```

QUIC

September 14, 2021

Quick UDP Internet Protocol (QUIC) is a combination of (TCP+TLS+HTTP/2) protocols implemented on UDP. The QUIC transport protocol multiplexes the connections between two endpoints using UDP. Also when compared to other protocols, QUIC provides a high-performance in terms of security, fast delivery of traffic, and lower latency.

A QUIC bridge is configured in a Citrix ADC appliance for load balancing QUIC traffic between a QUIC client and QUIC back-end server. The QUIC bridge enables you to have persistent QUIC connections between client and server if there is a NAT rebinding or a connection migration. This configuration however does not process data. It is used only for load balancing of QUIC traffic through the Citrix ADC appliance.

QUIC packets contain connection ID to allow endpoints to associate the packets with different address or 4-tuple to the same connection. The connection ID contains the details of the server ID that are shared to the Citrix ADC appliance and to the back end servers. The Citrix ADC appliance extracts the connection ID details of the server ID and sends the traffic back to the back end server. The connection IDs are in protected packets that makes the connections robust in the event of connection migration.

> **Important**
>
> The back end servers must have support to encode server ID in QUIC connection ID.

Benefits of QUIC bridge

QUIC bridge for the Citrix ADC appliance is preferred for the following reasons:

- No expensive crypto operations.
- Stateless routing is possible (no 4-tuple based load balancing).

QUIC bridge configuration

September 14, 2021

To configure QUIC bridge, you must complete the following:

- Add QUIC bridge profile
- Add QUIC back-end servers
- Add QUIC service on the appliance
- Add load balancing virtual server of type QUIC bridge
- Bind QUIC bridge to load balancing virtual server of type QUIC bridge

> **Important**
>
> Before you configure the QUIC bridge, ensure you first enable the load balancing feature on the appliance. For more information, see Set up basic load balancing.

Configure QUIC bridge by using the CLI

The following sections must be configured by using the CLI.

Add a QUIC bridge profile

You must add a QUIC bridge profile.

At the command prompt, type:

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```

Example:

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```

> **Note**
>
> The `serveridlen` parameter configured in the example is the length of a custom server ID,

> which is the hex string of IP and PORT.

Add QUIC back-end application server

You must add QUIC back-end application servers.

At the command prompt, type:

```
1 -  add server <name> (<IPAddress>)
2 -  add server <name> (<IPAddress>)
```

Example:

```
1 -  add server s1 192.0.2.20
2 -  add server s2 192.0.2.30
```

Add QUIC bridge service

You must add QUIC bridge service to the application servers.

At the command prompt, type:

```
1 -  add service  <name>  (<IP> | <serverName>) <serviceType> <port> [-
     CustomServerID <string>]
2
3 -  add service  <name>  (<IP> | <serverName>) <serviceType> <port> [-
     CustomServerID <string>]
```

Example:

```
1 -  add service src1 s1 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A8026401BB
2
3 -  add service src2 s2 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A802C801BB
```

> **Note**
>
> The `CustomServerID` parameters configured in the preceding example are the hex string of a corresponding IP and the PORT of the server (s1 and s2). For the QUIC bridge feature, Citrix recommends you to configure the `CustomServerID` parameter in the hex string format only.

Add a load balancing virtual server of type QUIC bridge

You must add a load balancing virtual server of type QUIC bridge.

At the command prompt, type:


```
add lb vserver <name> [<IPAddress>@ <port> [-persistenceType <
    persistenceType >] [-lbMethod < lbMethod > [-cltTimeout <secs>] [-
    quickBridgeProfilename <name>]
```

Example:

```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -cltTimeout 120 -
    quicBridgeProfilename q1
```

> **Note**
>
> While configuring the QUIC bridge virtual server, you must configure `persistenceType` parameter as `CUSTOMSERVERID` and "LbMethod" parameter as `TOKEN`.

Bind QUIC bridge service to the load balancing virtual server of type QUIC bridge

You must bind the QUIC bridge service to the load balancing virtual server of type QUIC bridge.

At the command prompt, type:

```
-  bind lb vserver <name> (<serviceName>)

-  bind lb vserver <name> (<serviceName>)
```

Example:

```
-  bind lb vserver quic_bridge_vip src1

-  bind lb vserver quic_bridge_vip src2
```

Configure QUIC bridge for service groups

You can also configure QUIC bridge capabilities to service groups. The following steps guide you to configure QUIC bridge for service groups.

To configure QUIC bridge for service groups, you must complete the following:

Add QUIC bridge profile

At the command prompt, type:

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```


Example:

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```

Add server of type QUIC

At the command prompt, type:

```
- add server <name> (<IPAddress>)
- add server <name> (<IPAddress>)
```

Example:

```
- add server s1 192.0.2.20
- add server s2 192.0.2.30
```

Add QUIC bridge service group

At the command prompt, type:

```
add serviceGroup <serviceGroupName> (<IP> | <serverName>) <serviceType>
```

Example:

```
add serviceGroup svg1 QUIC_BRIDGE
```

Bind the QUIC servers to the service group

At the command prompt, type:

```
- bind serviceGroup <serviceGroupName> (<IP>@ | (<serverName>) [-
  CustomServerID <string>]
- bind serviceGroup <serviceGroupName> (<IP>@ | (<serverName>) [-
  CustomServerID <string>]
```

Example:

```
- bind serviceGroup svg1 s1 443 -customServerID C0A8026401BB
- bind serviceGroup svg1 s2 443 -customServerID C0A802C801BB
```


Add load balancing virtual server of type QUIC bridge

At the command prompt, type:

```
add lb vserver <name> [<IPAddress>@ <port> [-persistenceType <
    persistenceType >] [-lbMethod < lbMethod > [-cltTimeout <secs>] [-
    quickBridgeProfilename <name>]
```

Example:

```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -cltTimeout 120 -
    quicBridgeProfilename q1
```

Bind the load balancing virtual server of type QUIC bridge to the service group

At the command prompt, type:

```
bind lb vserver <name>@ (<serviceName>@ <serviceGroupName>
```

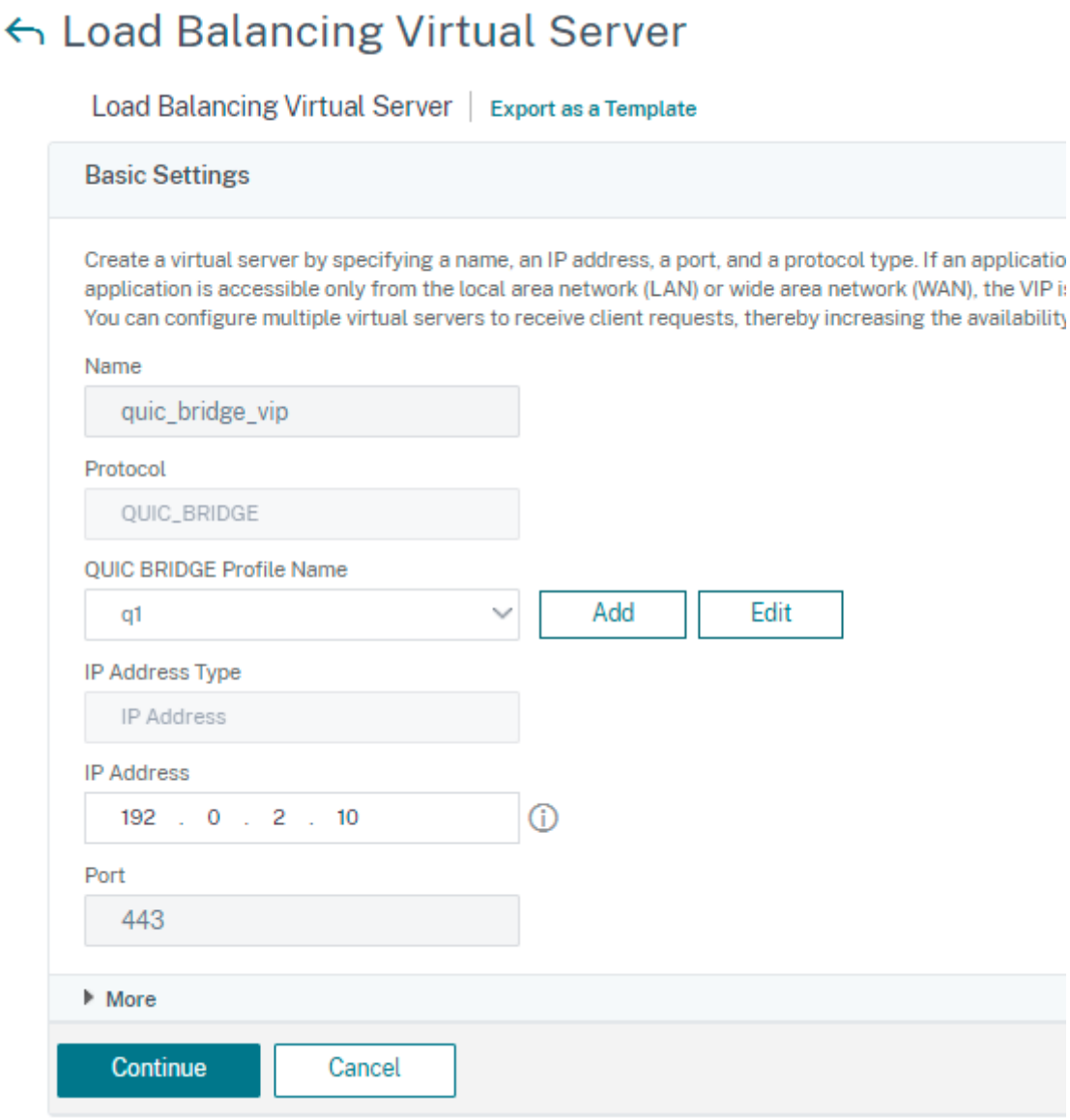
Example:

```
bind lb vserver quic_bridge_vip svg1
```

Configure QUIC bridge using the GUI

Complete the following steps to configure QUIC bridge by using the GUI.

1. Navigate to **Traffic Management > Load balancing > Virtual Servers**.
2. On the **Virtual Servers** page, click **Add**.
3. On the **Load Balancing Virtual Server** page, select the Protocol as QUIC_BRIDGE and enter the details. Click **OK**.

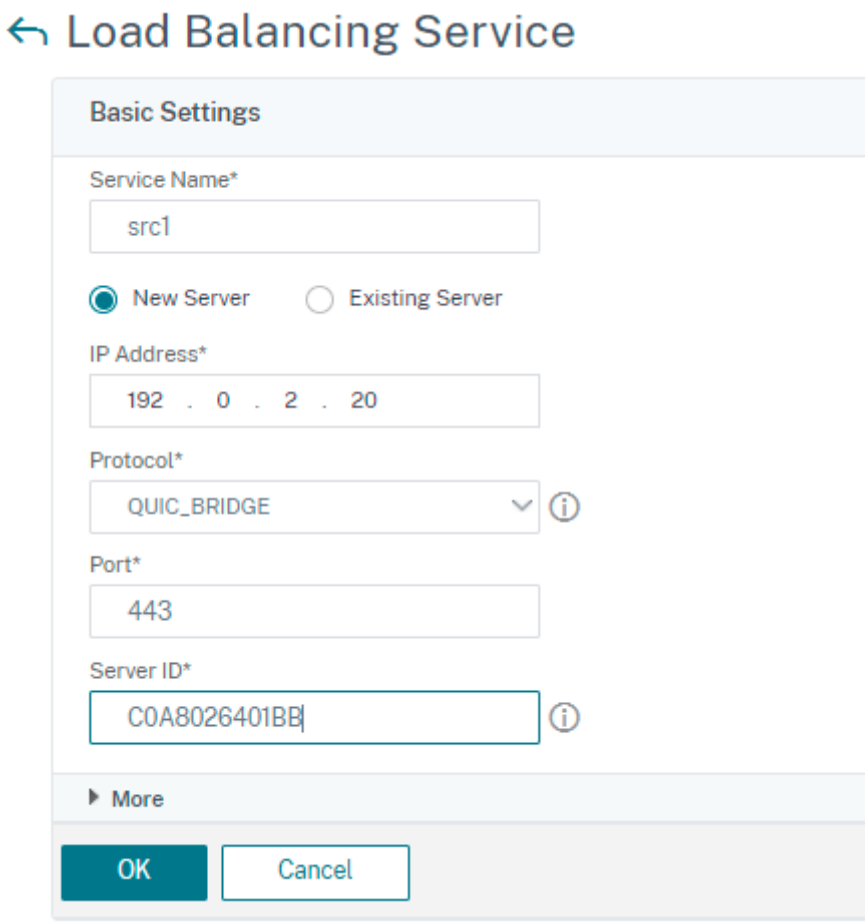


4. On the **Load Balancing Virtual Server** page, click **Continue** and **Done**.

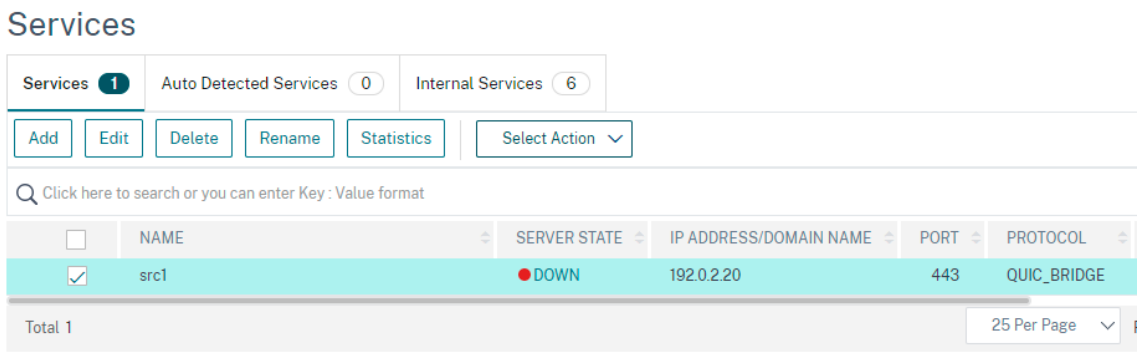
Configure load balancing for the services by using the GUI

Complete the following steps to configure load balancing for the services by using the GUI.

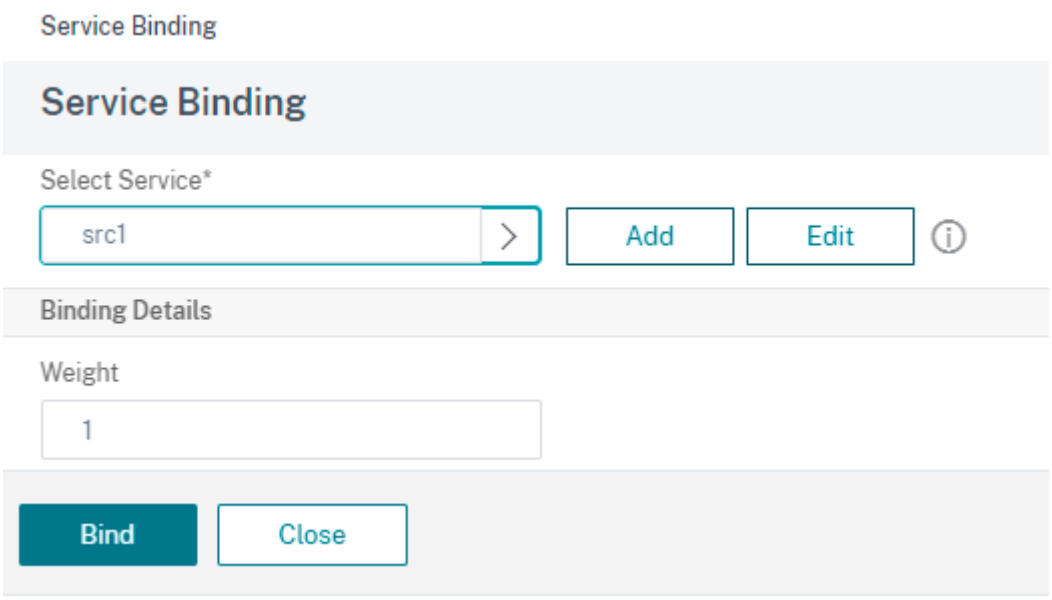
1. Navigate to **Traffic Management > Load Balancing > Services**. On the **Services** page, click **Add**.
2. On the **Load Balancing Service** page, enter the details and click **OK**.



3. On the **Virtual Servers** page, select the created virtual server to bind the service.
4. Scroll down on the **Load Balancing Virtual Server** page and select the **Services and Service Groups**.
5. On the **Service Binding** screen, click **Select Service** field.
6. On the **Service** screen, select the service to bind to the load balancing virtual server, and click **Select**.



7. The src1 service is selected and on the **Service Binding** screen, click **Bind**.



8. On the **Load Balancing Virtual Server** page, click **Done**.

Proxy protocol

October 8, 2021

Proxy protocol safely transports client details from client to server across Citrix ADC appliances. The appliance adds a proxy protocol header with client details and forwards it to the back-end server. Following are some of the usage scenarios for proxy protocol in a Citrix ADC appliance.

- Learning original client IP address
- Selecting a language for a website
- Block listing selected IP addresses
- Logging and collecting statistics.

Following are the three modes of operation:

- Insert. The appliance inserts the client details and sends it to the back-end server.
- Forward. The appliance forwards the client details to the back-end server.
- Stripped. The appliance stores the client details for logging purpose. Also, if the proxy protocol is not supported on the back-end server, sends the client details to the server by using the rewrite policy configuration

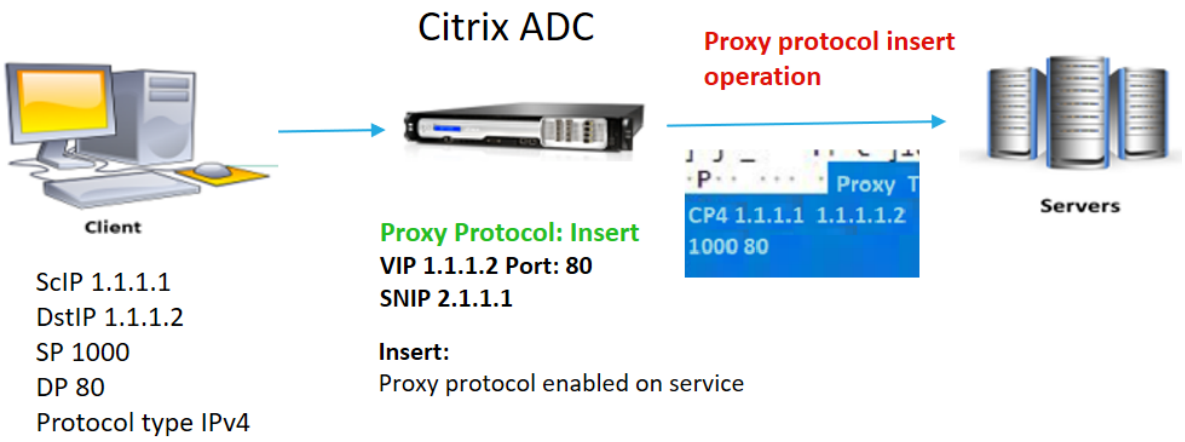
Limitations

The proxy protocol is not supported for the TCP Fast Open (TFO) and MultiPath TCP features. The feature is supported only for services for which the Citrix ADC appliance does TCP connection termination. It is not support for other services, for example, “ANY”.

How proxy protocol works in a Citrix ADC appliance

The following flow diagrams show how you can configure the proxy protocol across Citrix ADC appliances for Insert, Forwards, and Stripped operation:

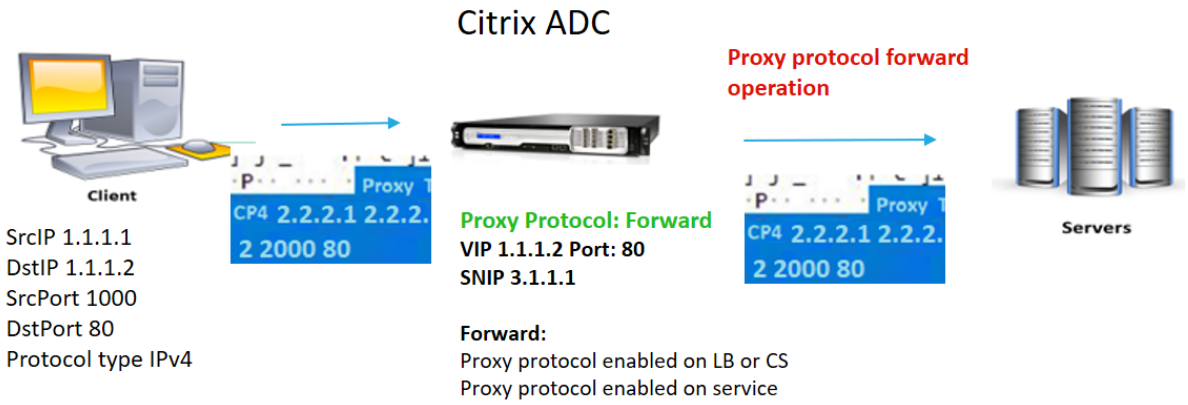
Insert operation



The component interact is as follows:

- At the Citrix ADC instance, you must enable proxy protocol in the net profile and bind it to the service.
- In the Insert operation, Citrix ADC adds a proxy header with client connection details and forwards it to the back-end server.
- On the sending side, the appliance decides the proxy protocol version based on CLI configuration.

Forward operation

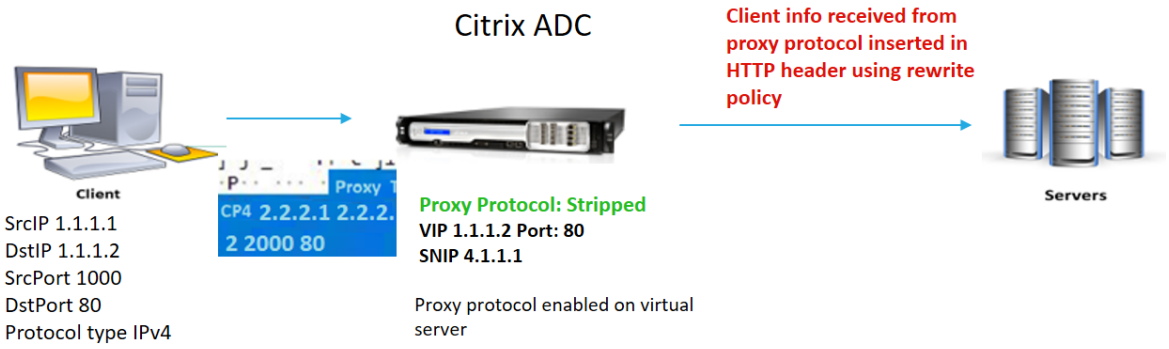


*** The original client details 2.2.2.1, 2.2.2.2, 2000, 80 in the proxy header is forwarded to the back-end server**

The component interact is as follows:

- A client sends a request along with the proxy header to the Citrix ADC. The appliance dynamically identifies the version.
- In the Citrix ADC appliance, it is a Forward operation. The proxy protocol is enabled on the load balancing virtual server or content switching virtual server and enabled on the service. The appliance receives the proxy header and forwards the header details to the back-end server.
- If the proxy header details are in invalid format, the appliance resets the connection.
- On the sending side, the appliance decides the proxy protocol version based on CLI configuration.

Stripped operation



The component interact is as follows:

- A client sends a request along with a proxy header to the Citrix ADC appliance.

- In the Citrix ADC appliance, if it is a Stripped operation, the appliance forwards the client information obtained from the proxy protocol and inserts it into the HTTP header using rewrite policy expressions.
- The client details such as source IP address, destination IP address, source port, and destination port are added in an HTTP header using rewrite policy expressions. The rewrite policy evaluates the expression and if "true," the corresponding rewrite policy action is triggered. And the client details are forwarded to the back-end server in an HTTP header.
- If the proxy header details are in invalid format, the appliance resets the connection.

Proxy protocol version formats

The Proxy protocol version is available as two formats. The appliance decides to use a format based on the incoming data length. For detailed information, see Proxy Protocol RFP.

1. Proxy protocol version-1 format

   ```
   PROXY TCP4/TCP6/UNKNOWN <SRC IP> <DST IP> <SRC PORT> <DST PORT>
   ```

   - PROXY -> Unique string format for Proxy header version -1.
   - Support protocols TCP over IPv4 and TCP over IPv6. For the remaining protocols, this is UNKNOWN.
   - SRC IP – Source IP (Original Client IP) address of a packet.
   - DST IP – Destination IP address of a packet.
   - SRC port – Source port of a packet.
   - DST port – Destination port of a packet.

2. Proxy protocol version-2 format

   ```
   0D 0A 0D 0A 00 0D 0A 51 55 49 54 0A <13th byte> <14th byte> <15-16th
   byte> <17th byte onwards>
   ```

   - D 0A 0D 0A 00 0D 0A 51 55 49 54 0A -> Unique binary string for Proxy header version -2.
   - Support protocols TCP over IPv4 and TCP over IPv6. For the remaining protocols, this is UNKNOWN.
   - Thirteenth byte – protocol version and command.
   - Fourteenth byte – address and protocol family.
   - 15-16th byte – Address length in network order.
   - Seventeenth byte onwards – Addresses info present in network order- src IP, dst IP, src port, dst port.

Configure Proxy protocol in Citrix ADC appliance

Complete the following steps to configure the Proxy protocol in your Citrix ADC appliance.

1. Enable proxy protocol as global.
2. Configure proxy protocol for Insert operation
3. Configure proxy protocol for Forward operation
4. Configure proxy protocol for Strip operation
5. Configure proxy protocol for no operation

Enable the proxy protocol as global

At the command prompt, type the following:

```
set ns param -proxyProtocol ENABLED
```

Configure proxy protocol for Insert operation

To configure the proxy protocol for Insert operation, you must enable or disable the protocol on the load balancing virtual server and enable it on the service.

Add net profile with Proxy protocol disabled for load balancing virtual server

At the command prompt, type the following:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

Example:

```
Add netprofile proxyprofile-1 -proxyProtocol DISABLED -proxyprotocoltxversion
 V1
```

> **Note:**
>
> If you disable proxy protocol on your appliance, you need not set the protocol version parameter.

Add net profile with a proxy protocol enabled for service

At the command prompt, type the following:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

Example:

```
add netprofile proxyprofile-2 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```


Add load balancing virtual server for Citrix ADC appliance in the proxy layer

At the command prompt, type the following:

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

Example:

```
add lb vserver lbvserver-1 http 1.1.1.1 80
```

Add HTTP service for Citrix ADC appliance in the proxy layer

At the command prompt, type the following:

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

Example:

```
Add service http-service-1 2.2.2.1 http 80
```

Set net profile with load balancing virtual server in Citrix ADC appliance

At the command prompt, type the following:

```
set lb vserver <vserver name> -netprofile <name>
```

Example:

```
set lb vserver lbvserver-1 -netprofile proxyProfile-1
```

Set net profile with HTTP service in Citrix ADC appliance

At the command prompt, type the following:

```
set service <service name> -netprofile <name>
```

Example:

```
set service http-service-1 -netprofile proxyProfile-1
```

Configure proxy protocol for forward operation

To configure the proxy protocol for Forward operation for the next Citrix ADC instance in the proxy layer. You must enable or disable the protocol and bind to the virtual server or service.

Add net profile with proxy protocol enabled for load balancing virtual server

At the command prompt, type the following:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

Example:

```
add netprofile proxyprofile-3 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

Add net profile with proxy protocol enabled for service

At the command prompt, type the following:

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

Example:

```
add netprofile proxyprofile-4 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

Add load balancing virtual server for Citrix ADC appliance in the proxy layer

At the command prompt, type the following:

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

Example:

```
add lb vserver lbvserver-2 http 2.2.2.2 80
```

Add HTTP service for Citrix ADC appliance in the proxy layer

At the command prompt, type the following:

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

Example:

```
Add service http-service-2 3.3.3.1 http 80
```

Set net profile with load balancing virtual server in Citrix ADC appliance

At the command prompt, type the following:

```
set lb vserver <vserver name> -netprofile <name>
```


Example:

```
set lb vserver lbvserver-2 -netprofile proxyProfile-3
```

Set net profile with HTTP service in Citrix ADC appliance

At the command prompt, type the following:

```
set service <service name> -netprofile <name>
```

Example:

```
set service http-service-2 -netprofile proxyProfile-4
```

Configure proxy protocol for strip operation

To configure the proxy protocol for strip operation, you must enable the proxy protocol on the load balancing virtual server and disable the proxy protocol on the service.

Add net profile with proxy protocol enabled for virtual server

At the command prompt, type the following:

```
add netprofile <name> -proxyProtocol ENABLED> -proxyprotocoltxversion <V1/
V2>
```

Example:

```
add netprofile proxyprofile-5 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

Add load balancing or content switching virtual server for Citrix ADC appliance in the proxy layer

At the command prompt, type the following:

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

Example:

```
add lb vserver lbvserver-3 http 2.2.2.2 80
```

Add HTTP service for Citrix ADC appliance in the proxy layer

At the command prompt, type the following:

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```


Example:

```
Add service http-service-3 3.3.3.1 http 80
```

Set net profile with load balancing or content switching virtual server in Citrix ADC appliance

At the command prompt, type the following:

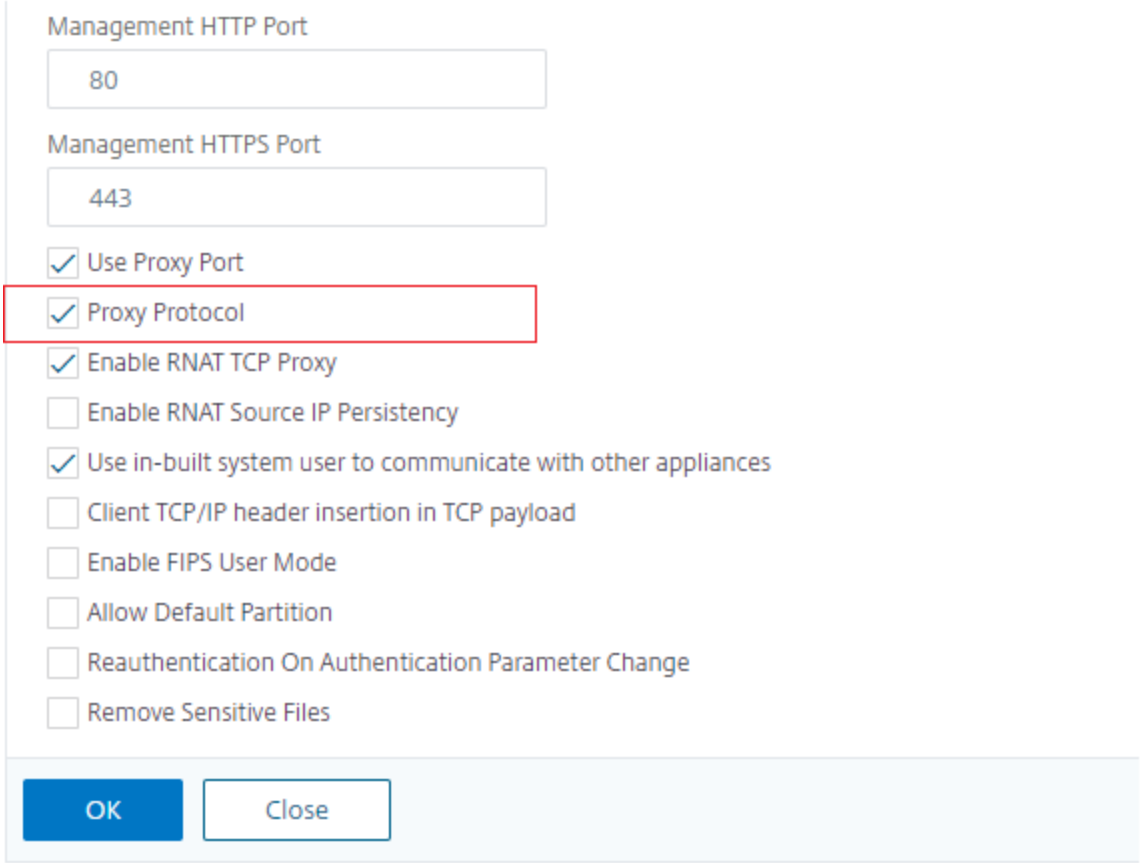
```
set lb vserver <vserver name> -netprofile <name>
```

Example:

```
set lb vserver lbvserver-3 -netprofile proxyProfile-5
```

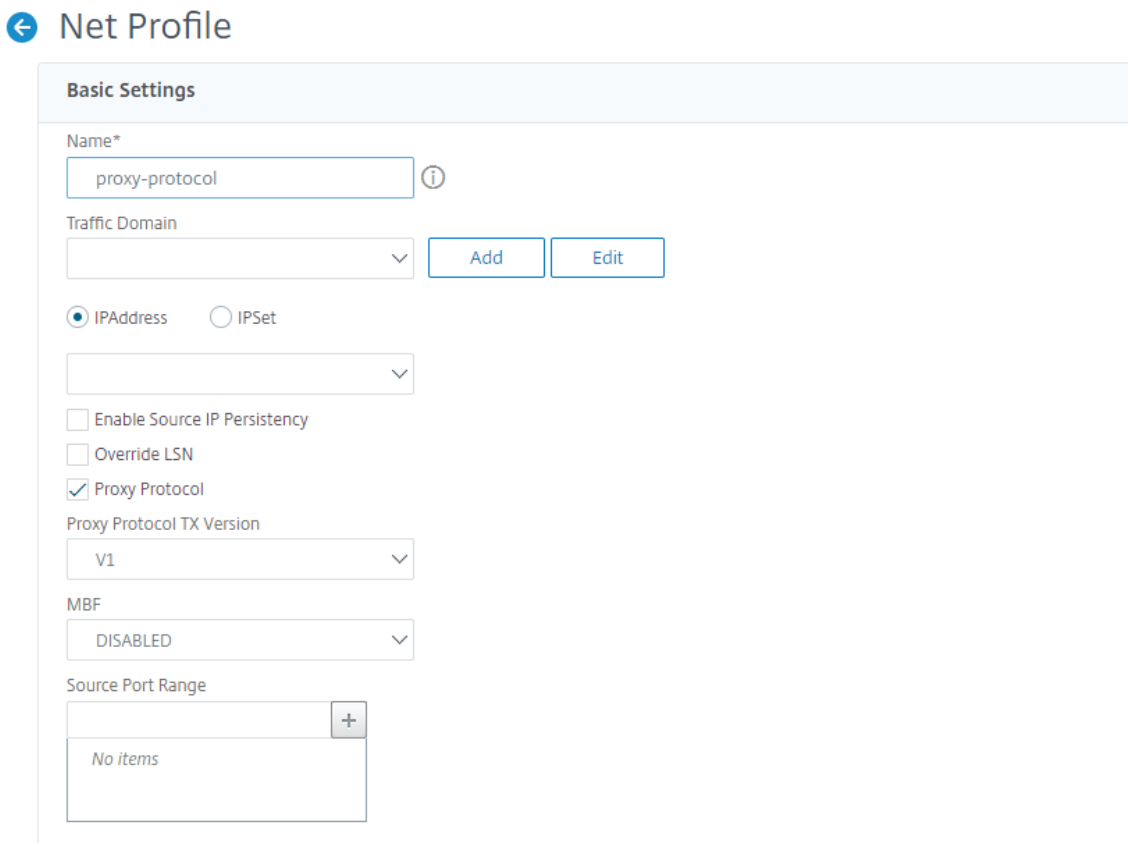
Configure Proxy protocol by using Citrix ADC GUI

1. Navigate to **System** > **Settings** > **Change Global System Settings**.
2. In the **Configure Global System Settings Parameters** page, select **Proxy Protocol** check box.
3. Click **OK** and **Close**.

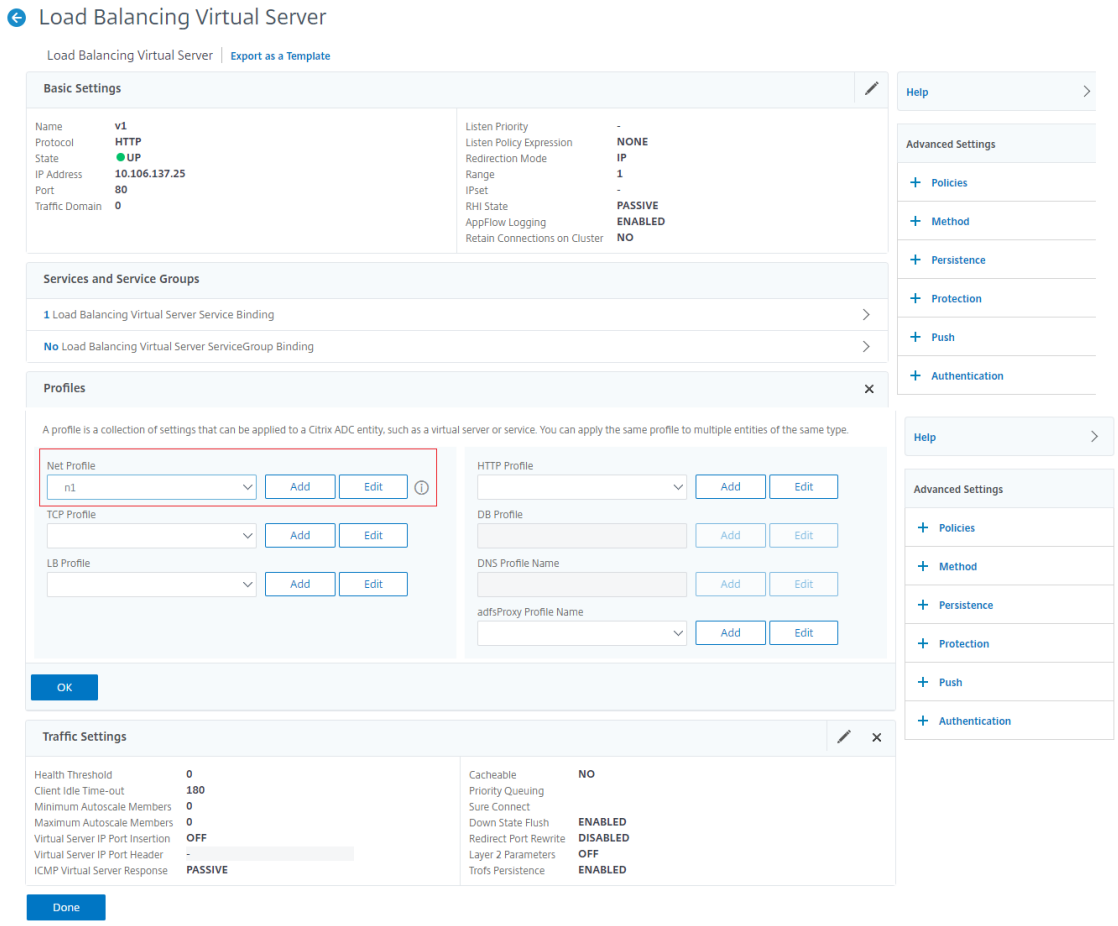


4. Navigate to **System** > **Network** > **Net Profiles**.

5. In the details pane, click **Add** to create a net profile for the load balancing virtual server.
6. In the **Net Profile** page, set the following parameters:
   a) Name. Name of the net profile.
   b) Proxy Protocol. Enable or disable proxy protocol for the load balancing virtual server.
   c) Proxy Protocol TX Version. Set proxy protocol version as V1 or V2 based on incoming data format.
7. Click **OK**.

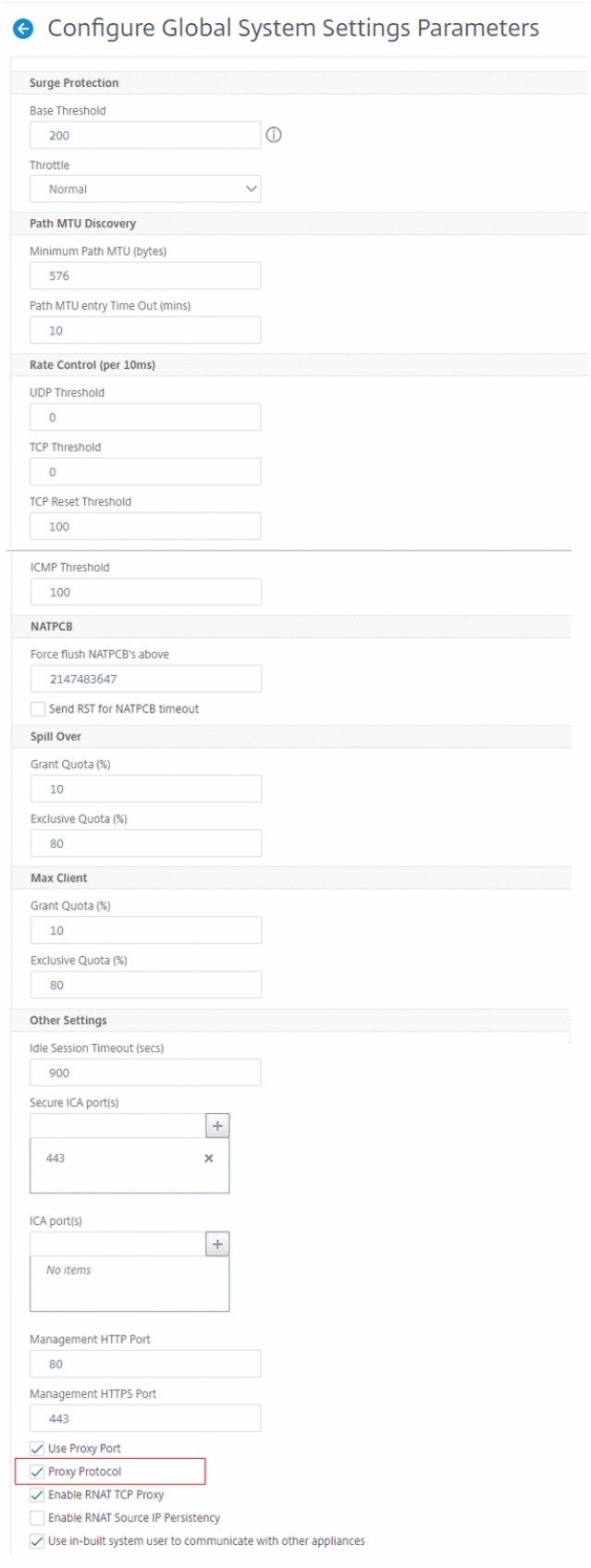


8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
9. In the details pane, click **Add**.
10. In the **Load Balancing Virtual Server** page, the set the basic parameters.
11. In the **Advanced Settings** section, select **Profiles**.
12. In the **Profiles** section, click the pencil icon.
13. Select a net profile and click **OK**.
14. Click **Done**.



15. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
16. In the details pane, click **Add**.
17. In the **Load Balancing Service** page, set the basic parameters.
18. In the **Advanced Settings** section, select **Profiles**.
19. In the **Profiles** section, click the pencil icon.
20. Select a net profile and click **OK**.
21. Click **Done**.

> **Note:**
>
> If you have more than one Citrix ADC appliance as part of the proxy layer, you must set the proxy protocol configuration on each appliance for the Forward operation.



Client IP address in TCP option

September 14, 2021

The Citrix ADC appliance uses many ways to send the client information to the back-end server. One such method is by sending the client IP address in the TCP option of the first data packet. The appliance uses the TCP option number in the TCP profile, if the back-end server using TCP option to read the client IP address. The IP address is carried in the TCP option number 28 (configurable on the appliance service).

The TCP option method includes both insert and forward functionality in carrying the client IP address to the back-end server.

In the TCP option configuration, the appliance adds a TCP option, 28 to insert the client IP address and forward it to the back-end server. Following are some the of usage scenarios for TCP option configuration in a Citrix ADC appliance.

Multiplexing is disabled if this feature is enabled for traffic coming to TCP profile. Also, if nsapimgr and clientip tcp-options in TCP profile are enabled, clientip tcp-option takes precedence.

> **Note:**
>
> However, multiplexing is disabled on the appliance if Client IP TCP option is enabled for the traffic that comes to the TCP profile.

- Learning original client IP address
- Selecting a language for a website
- Block listing selected IP addresses

Following are the two modes of operation:

- Insert. The appliance adds the client details in the TCP option 28 (configurable but preferable value is 28) field and sends it to the back-end server.
- Forward. The appliance forwards the client details in the TCP option 28 (configurable on the front-end of the appliance service). However, the option number at the back-end can be modified based on the value configured in the back-end

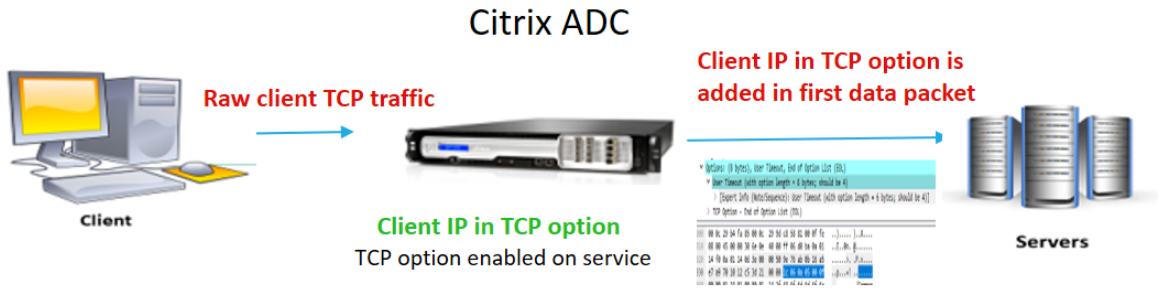
> **Note:**
>
> In case of TCP or HTTP virtual server, the TCP option number is forwarded with or without this feature enabled in transparent mode.

Limitations

The TCP option configuration feature is not supported in TFO, MultiPath TCP, and HTTP2 features.

How TCP option configuration in a Citrix ADC appliance

The following flow diagrams show how you can configure TCP option in the Citrix ADC appliances for Insert and forward operations.

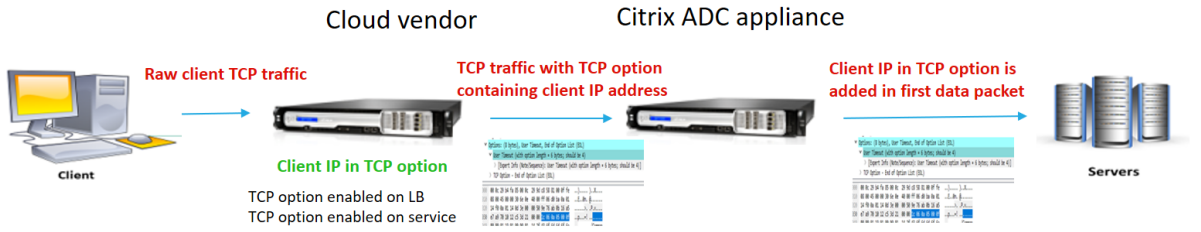


The component interact is as follows:

- A client sends a request to Citrix ADC.
- In Citrix ADC appliance, you must create a TCP profile, enable the TCP option feature, and specify the TCP option number.

  Note: It is advisable to configure TCP option number as 28 in the TCP profile.
- In Insert operation, Citrix ADC inserts the client details in the TCP option 28 bound to the service. The client details are then sent it to the back-end server. If the incoming traffic is HTTPS, the client IP address in the TCP option will be sent in the SSL client hello message which is the first data packet at the TCP level

Forward operation:



The component interact is as follows:

- A client sends a HTTP/HTTPS request to Citrix ADC.
- At Citrix ADC appliance, if is a Forward operation, the TCP option is enabled on load balancing virtual server or content switching virtual server and also enabled on the service. The appliance receives the client info in the TCP option number specified in the virtual server and forwards it to the back-end server in the TCP option number (configurable in the service) added in the first data packet

Configure TCP option for Insert operation

Following the procedure given below to configure TCP option in your Citrix ADC appliance.

1. Add a TCP profile.
2. Configure TCP option for Insert operation
3. Bind TCP profile to service

Add a TCP profile

At the command prompt, type:

```
add tcpprofile <name> -clientIpTcpOption (enabled | disabled> -clientIpTcpOptionNumber
 <positive_integer>
```

Example:

```
add tcprofile p1
```

Configure TCP option for Insert operation

At the command prompt, type:

```
add tcpprofile <name> -clientIpTcpOption (enabled | disabled> -clientIpTcpOptionNumber
 <positive_integer>
```

Example:

```
add tcpprofile p1 -clientIpTcpOption ENABLED -clientIpTcpOptionNumber 28
```

Add service

At the command prompt, type:

```
add service <name> <server name> <service type> <port>
```

Example:

```
add service service-http1 1.1.1.1 HTTP 80
```

Bind TCP profile to service

At the command prompt, type:

```
set service <name> -tcpprofileName <name>
```

Example:

```
set service s1 -tcpprofileName p1
```


> **Note:**
>
> The basic configuration for service must be taken care.

Configure TCP option for Forward operation

Following the procedure given below to configure TCP option in the TCP profile for Forward operation.

1. Add TCP profile with TCP option number
2. Bind TCP profile to virtual server
3. Bind TCP profile to service.

Add TCP profile with TCP option number

At the command prompt, type:

```
add tcpprofile <name> -clientIpTcpOption (enabled | disabled> -clientIpTcpOptionNumber
 <positive_integer>
```

Example:

```
add tcpprofile p1 -clientIpTcpOption ENABLED -clientIpTcpOptionNumber 29
```

Bind TCP profile to virtual server (load balancing or content switching)

At the command prompt, type:

```
set lb vserver <name> -tcpprofileName <name>
```

Example:

```
set lb vservice s1 -tcpprofileName p1
```

Bind TCP profile to service

At the command prompt, type:

```
set service <name> -tcpprofileName p1
```

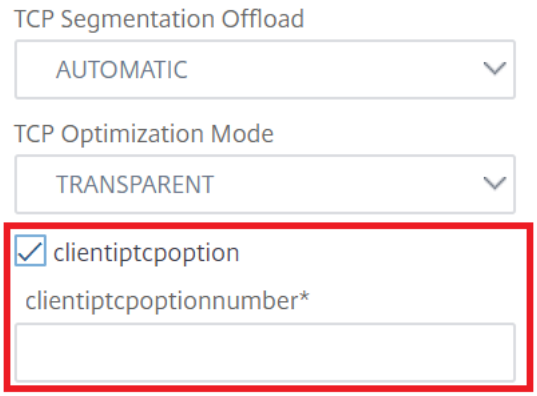
Example:

```
set service s1 -tcpprofileName p1
```

Configure TCP option by using Citrix ADC GUI

1. Navigate to **System** > **Profiles**.

2. In the **TCP Profile** tab page, click **Add**.
3. In the **Configure TCP profile** page, configure the following parameters:
   a. clientIptcpoption. TCP option to send or receive client IP address.
   b. clientiptcpoptionnumber. Configurable TCP option number to receive the client IP address.



4. Click **OK** and **Close**.

SNMP

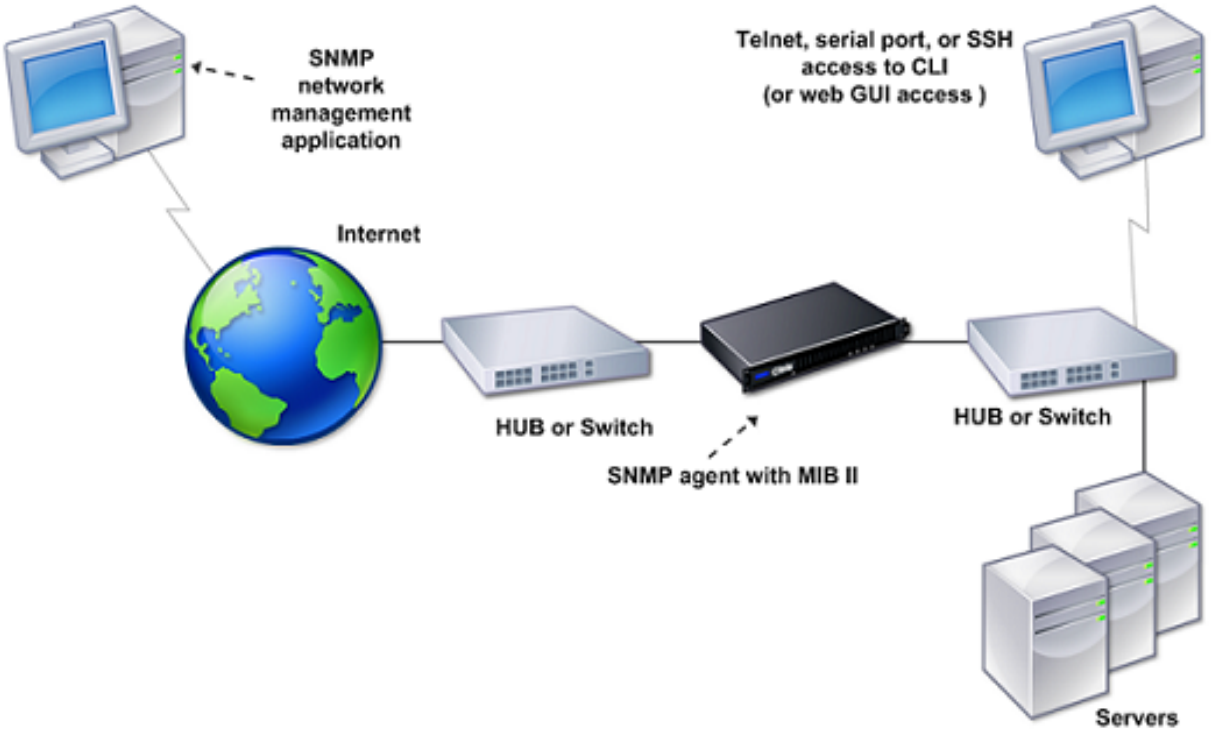
September 14, 2021

You can use Simple Network Management Protocol (SNMP) to configure the SNMP agent on the Citrix ADC appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the Citrix ADC. The traps are then sent to a remote device called a *trap listener*, which signals the abnormal condition on the Citrix ADC appliance. Or, you can query the SNMP agent for System-specific information from a remote device called an *SNMP manager*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The SNMP agent on the Citrix ADC can generate traps compliant with SNMPv1, SNMPv2, and SNMPv3. For querying, the SNMP agent supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

For information about SNMP parameters, traps, and its descriptions, see Citrix ADC SNMP OID Reference.

The following figure illustrates a network with a Citrix ADC that has SNMP enabled and configured. In the figure, each SNMP network management application uses SNMP to communicate with the SNMP agent on the Citrix ADC. The SNMP agent searches its management information base (MIB) to collect the data requested by the SNMP Manager and provides the information to the application.



> **Important**
>
> The SNMP module in a Citrix ADC appliance supports a maximum length of 128 bytes (as compliant with RFC 3416) for an SNMP OID. A long index variable name for an object can result in an SNMP OID exceeding 128 bytes in length.
>
> To resolve this issue, the Citrix ADC SNMP module supports a maximum length of 31 characters for an index variable name. If an index variable name exceeds 31 characters in length, the SNMP module using an hash algorithm converts the name to a 31 characters hash value. This hashed value is used in the SNMP OID for that variable.
>
> The original index variable name is stored in another variable, which has the following name format: `<variable type>FullName`. For example, When the name of a load balancing virtual server has more the 31 characters, `vserverName` SNMP OID contains the hashed value and `vsvrFullName` SNMP OID contains the full (original) name of the virtual server.
>
> Similarly, for SNMP traps, the index variable displays a hashed valued. `<variable type>FullName`, which stores the full name of the original index variable name, is also part of the trap messages.

Importing MIB Files to the SNMP Manager and Trap Listener

To monitor a Citrix ADC appliance, you must download the MIB object definition files. The Citrix ADC appliance supports the following enterprise-specific MIBs:

- **A subset of standard MIB-2 groups**. Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.
- **A system enterprise MIB**. Provides system-specific configuration and statistics.

You can obtain the MIB object definition files from the /netscaler/snmp directory or from the Downloads tab of the GUI.

Configuring the Citrix ADC to generate SNMP traps

September 14, 2021

You can configure the Citrix ADC appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the appliance. The traps are sent to a remote device called a *trap listener*. It helps administrators monitor the appliance and respond promptly to any issues.

The Citrix ADC appliance provides a set of condition entities called *SNMP alarms*. When the condition in any SNMP alarm is met, the appliance generates SNMP trap messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the appliance.

To configure the Citrix ADC appliance to generate traps, you need to enable and configure alarms. Then, you specify the trap listeners to which the appliance sends the generated trap messages.

Enabling an SNMP alarm

The Citrix ADC appliance generates traps only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them.

When you enable an SNMP alarm, the appliance generates corresponding trap messages when some events occur. Some alarms are enabled by default.

To enable an SNMP alarm by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `enable snmp alarm <trapName>`
- `show snmp alarm <trapName>`

To enable an SNMP alarm by using the GUI

1. Navigate to **System > SNMP > Alarms**, and select the alarm.
2. Click **Actions** and select **Enable**.

Configuring alarms

The Citrix ADC appliance provides a set of condition entities called *SNMP alarms*. When the condition set for an SNMP alarm is met, the appliance generates SNMP traps messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the appliance.

You can assign an SNMP alarm with a severity level. When you do this, the corresponding trap messages are assigned that severity level.

The following are the severity levels, defined on the appliance, in decreasing order of severity.

- Critical
- Major
- Minor
- Warning
- Informational

For example, if you set a warning severity level for the SNMP alarm named LOGIN-FAILURE, the trap messages generated when there is a login failure is assigned with the warning severity level.

> **Note**
>
> Citrix ADC supports various SNMP alarms. For more information, see SNMP alarms.

You can also configure an SNMP alarm to log the corresponding trap messages generated whenever the condition on that alarm is met.

To configure an SNMP alarm by using the CLI

At the command prompt, type the following commands to configure an SNMP alarm and verify the configuration:

- `set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm <trapName>`

Where,

ThresholdValue: Value for the high threshold. The Citrix ADC appliance generates an SNMP trap message when the value of the attribute associated with the alarm is greater than or equal to the specified high threshold value.

NormalValue: Value for the normal threshold. A trap message is generated if the value of the respective attribute falls to or below this value after exceeding the high threshold.

To configure SNMP alarms by using the GUI

Navigate to **System > SNMP > Alarms**, select an alarm, and configure the alarm parameters.

Configuring SNMPv1 or SNMPv2 traps

After configuring the alarms, you need to specify the trap listener to which the appliance sends the trap messages. Apart from specifying parameters such as IP or IPv6 address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

You can also configure the appliance to send SNMP trap messages with a source IP address other than the Citrix ADC IP (NSIP or NSIP6) address to a particular trap listener. For a trap listener that has an IPv4 address, you can set the source IP to either a mapped IP (MIP) address or a subnet IP (SNIP) address configured on the appliance. For a trap listener that has an IPv6 address, you can set the source IP to a subnet IPv6 (SNIP6) address configured on the appliance.

You can also configure the appliance to send trap messages to a trap listener based on a severity level. For example, if you set the severity level as Minor for a trap listener, all trap messages of the severity level equal to or greater than Minor (Minor, Major, and Critical) are sent to the trap listener.

If you have defined a community string for the trap listener, you must also specify a community string for each trap that is to be sent to the listener. A trap listener for which a community string has been defined accepts only trap messages that include a community string matching the community string defined in the trap listener. Other trap messages are dropped.

To add an SNMP trap by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 )-destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>`
- `show snmp trap`

Example:

```
1 > `add snmp trap specific 192.0.2.10 -version V2 -destPort 80 -
     communityName com1 -severity Major`
2 <!--NeedCopy-->
```


To configure SNMP traps by using the GUI

Navigate to **System > SNMP > Traps**, and create the SNMP trap.

Configuring SNMPv3 traps

SNMPv3 provides security capabilities such as authentication and encryption by using the credentials of SNMP users. An SNMP manager can receive SNMPv3 trap messages only if its configuration includes the password assigned to the SNMP user.

The trap destination can now receive SNMPv1, SNMPv2, and SNMPv3 trap messages.

To configure an SNMPv3 trap by using the CLI

At the command prompt, do the following:

1. Add an SNMPv3 trap.

   ```
   add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 | V3)
   -destPort <port> -communityName <string> -srcIP <ip_addr> -severity <
   severity>
   ```

   > **Note**
   >
   > Once set, the SNMP trap version cannot be modified.

   Example

   ```
   1 > add snmp trap specific 192.0.2.10 -version V3 -destPort 80 -
       communityName com1 -severity Major
   2 <!--NeedCopy-->
   ```

2. Add an SNMP user.

   ```
   add snmp user <name> -group <string> [ -authType ( MD5 | SHA ){ -
   authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]
   ```

   Example

   ```
   1 > add snmp user edocs_user -group edocs_group
   2 <!--NeedCopy-->
   ```

3. Bind the SNMPv3 trap to the SNMP user.

```
bind snmp trap <trapClass> <trapDestination> [-version <version>] (-userName
 <string> [-securityLevel <securityLevel>])
```

Example


```
1      > bind snmp trap specific 192.0.2.10 -version V3 -userName
          edocs_user -securityLevel authPriv
2  <!--NeedCopy-->
```

To configure an SNMPv3 trap by using the GUI

1. Add an SNMPv3 trap.

   Navigate to **System > SNMP > Traps**, and create the SNMP trap by selecting V3 as the SNMP version.

2. Add an SNMP user.

   Navigate to **System > SNMP > Users** and create the SNMP user.

3. Bind the SNMPv3 trap to the SNMP user.

   - Navigate to **System > SNMP > Traps**, and select the SNMP version 3 trap.
   - Select the user to which the trap should be bound and define the appropriate Security Level.

SNMP trap logging

A Citrix ADC appliance can log SNMP trap messages (for SNMP alarms in which logging capability is enabled) when you enable the SNMP trap logging option and at least one trap listener is configured on the appliance. Now, you can specify the audit log level of trap messages sent to an external log server. The default log level is Informational. Possible values are Emergency, Alert, Critical, Error, Warning, Debug, and Notice.

For example, you can set the audit log level to Critical for an SNMP trap message generated by a logon failure. That information is then available on the NSLOG or SYSLOG server for troubleshooting.

To enable SNMP trap logging and configure trap log level by using the CLI

At the command prompt, type the following commands to configure SNMP trap logging and verify the configuration:

- `set snmp option [-snmpTrapLogging (ENABLED | DISABLED)][-snmpTrapLoggingLevel <snmpTrapLoggingLevel>]`
- `show snmp option`

To enable SNMP trap logging and configure SNMP trap log level by using the GUI

Navigate to **System > SNMP**, click Change SNMP Options, and set the following parameters:

1. SNMP Trap Logging—Select this check box to enable SNMP trap logging when at least one trap listener is configured on the appliance.
2. SNMP Trap Logging Level—Select an audit log level for the SNMP trap. By default, the audit level for an SNMP trap is set to "Informational."

Configuring Citrix ADC for SNMP v1 and v2 queries

September 14, 2021

You can query the Citrix ADC SNMP agent for system-specific information from a remote device called *SNMP managers*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The following types of SNMP v1 and v2 queries are supported by the SNMP agent:

- GET
- GET NEXT
- ALL
- GET BULK

You can create strings called community strings and associate each of these to query types. You can associate one or more community strings to each query type. Community string are passwords and used to authenticate SNMP queries from SNMP managers.

For example, if you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the Citrix ADC appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

Specifying an SNMP manager

You must configure the Citrix ADC appliance to allow the appropriate SNMP managers to query it. You must also provide the SNMP manager with the required Citrix ADC-specific information. You can add up to a maximum of 100 SNMP managers or networks.

For an IPv4 SNMP manager you can specify a host name instead of the manager's IP address. If you do so, you must add a DNS name server that resolves the host name of the SNMP manager to its IP address. You can add up to a maximum of five host-name based SNMP managers.

> **Note**:
> The appliance does not support use of host names for SNMP managers that have IPv6 addresses. You must specify the IPv6 address.

If you do not configure at least one SNMP manager, the appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds only to SNMP queries from those specific IP addresses.

If you remove an SNMP manager from the configuration, that manager can no longer query the appliance.

To add SNMP managers by specifying IP addresses by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp manager <IPAddress> ... [-netmask <netmask>]`
- `show snmp manager`

Example

```
> add snmp manager 10.102.29.10 10.102.29.15 10.102.29.30
```

To add an SNMP manager by specifying its host name by using the command line interface

Important: If you specify the SNMP manager's host name instead of its IP address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see "Adding a Name Server."

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp manager <IPAddress> [-domainResolveRetry ****<integer>]`
- `show snmp manager`

Example

```
add nameserver 10.103.128.15
add snmp manager engwiki.eng.example.net -domainResolveRetry 10
```

To add an SNMP manager by using the GUI

1. Navigate to **System** > **SNMP** > **Managers**, and create the SNMP manager.

> **Important**:
>
> If you specify the SNMP manager's host name instead of its IPv4 address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address.

> **Note**:
> The appliance does not support host names for SNMP managers that have IPv6 addresses.

Specifying an SNMP community

You can create strings called community strings and associate them with the following SNMP query types on the appliance:

- GET
- GET NEXT
- ALL
- GET BULK

You can associate one or more community strings to each query types. For example, when you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

If you do not associate any community string to a query type then the SNMP agent responds to all SNMP queries of that type.

To specify an SNMP community by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp community <communityName> <permissions>`
- `show snmp community`

Example

```
> add snmp community com all
```

To configure an SNMP community string by using the GUI

Navigate to **System** > **SNMP** > **Community**, and create the SNMP community.

Configuring Citrix ADC for SNMPv3 queries

September 14, 2021

Simple Network Management Protocol Version 3 (SNMPv3) is based on the basic structure and architecture of SNMPv1 and SNMPv2. However, SNMPv3 enhances the basic architecture to incorporate administration and security capabilities, such as authentication, access control, data integrity check, data origin verification, message timeliness check, and data confidentiality.

To implement message level security and access control, SNMPv3 introduces the user-based security model (USM) and the view-based access control model (VACM).

- **User-Based Security Model.** The user-based security model (USM) provides message-level security. It enables you to configure users and security parameters for the SNMP agent and the SNMP manager. USM offers the following features:
  - **Data integrity:** To protect messages from being modified during transmission through the network.
  - **Data origin verification:** To authenticate the user who sent the message request.
  - **Message timeliness:** To protect against message delays or replays.
  - **Data confidentiality:** To protect the content of messages from being disclosed to unauthorized entities or individuals.
- **View-Based Access Control Model.** The view-based access control model (VACM) enables you to configure access rights to a specific subtree of the MIB based on various parameters, such as security level, security model, user name, and view type. It enables you to configure agents to provide different levels of access to the MIB to different managers.

Citrix ADC supports the following entities that enable you to implement the security features of SNMPv3:

- SNMP Engines
- SNMP Views
- SNMP Groups
- SNMP Users

These entities function together to implement the SNMPv3 security features. Views are created to allow access to subtrees of the MIB. Then, groups are created with the required security level and access to the defined views. Finally, users are created and assigned to the groups.

> **Note**:
>
> The view, group, and user configuration are synchronized and propagated to the secondary node in a high availability (HA) pair. However, the engine ID is neither propagated nor synchronized as it is unique to each Citrix ADC appliance.

To implement message authentication and access control, you need to do the following:

Setting the engine ID

SNMP engines are service providers that reside in the SNMP agent. They provide services such as sending, receiving, and authenticating messages. SNMP engines are uniquely identified using engine IDs.

The Citrix ADC appliance has a unique engineID based on the MAC address of one of its interfaces. It is not necessary to override the engineID. However, if you want to change the engine ID, you can reset it.

To set the engine ID by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `set snmp engineId <engineID>`
- `show snmp engineId`

Example

```
> set snmp engineId 8000173f0300c095f80c68
```

To set the engine ID by using GUI

Navigate to **System** > **SNMP** > **Users**, click **Configure Engine ID** and type an engine ID.

Configure a view

SNMP views restrict user access to specific portions of the MIB. SNMP views are used to implement access control.

To add an SNMP view by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp view <name> <subtree> -type ( included | excluded )`
- `show snmp view <name>`
- `rm snmp view <name> <subtree>`

Where,

Name. Name for the SNMPv3 view. It can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=),

colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 view.

Subtree. A particular branch (subtree) of the MIB tree that you want to associate with this SNMPv3 view. You must specify the subtree as an SNMP OID. This is an argument of maximum Length: 99.

type. Include or exclude the subtree, specified by the subtree parameter, in or from this view. This setting can be useful when you have included a subtree, such as A, in an SNMPv3 view and you want to exclude a specific subtree of A, such as B, from the SNMPv3 view. This is a mandatory argument. Possible values: included, excluded.

Examples

add snmp view SNMPv3test 1.1.1.1 -type included
sh snmp view SNMPv3test
rm snmp view SNMPv3test 1.1.1.1

To configure an SNMP view by using the GUI

Navigate to **System** > **SNMP** > **Views**,and create the SNMP view.

Configure a group

SNMP groups are logical aggregations of SNMP users. They are used to implement access control and to define the security levels. You can configure an SNMP group to set access rights for users assigned to that group, thereby restricting the users to specific views.

You need to configure an SNMP group to set access rights for users assigned to that group.

To add an SNMP group by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp group <name> <securityLevel> -readViewName <string>`
- `show snmp group <name> <securityLevel>`

Where,

Name. Name for the SNMPv3 group. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 group.

securityLevel. Security level required for communication between the Citrix ADC appliance and the SNMPv3 users who belong to the group. Specify one of the following options:

noAuthNoPriv. Require neither authentication nor encryption.

authNoPriv. Require authentication but no encryption.

authPriv. Require authentication and encryption. Note: If you specify authentication, you must specify an encryption algorithm when you assign an SNMPv3 user to the group. If you also specify encryption, you must assign both an authentication and an encryption algorithm for each group member. This is a mandatory argument. Possible values: noAuthNoPriv, authNoPriv, authPriv.

readViewName. Name of the configured SNMPv3 view that you want to bind to this SNMPv3 group. An SNMPv3 user bound to this group can access the subtrees that are bound to this SNMPv3 view as type INCLUDED, but cannot access the ones that are type EXCLUDED. If the Citrix ADC appliance has multiple SNMPv3 view entries with the same name, all such entries are associated with the SNMPv3 group. This is a mandatory argument. Maximum Length: 31

To configure an SNMP group by using the GUI

Navigate to **System** > **SNMP** > **Groups**, and create the SNMP group.

Configuring a user

SNMP users are the SNMP managers that the agents allow to access the MIBs. Each SNMP user is assigned to an SNMP group.

You need to configure users at the agent and assign each user to a group.

To configure a user by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp user <name> -group <string> [-authType ( MD5 | SHA ){ -authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]`
- `show snmp user <name>`

Where,

authType is the authentication option available while configuring an user. There are two authentication types such as MD5 and SHA.

privType is the encryption option available while configuring an user. There are two types of encryption such as DES of key size 128 bit, and AES of key size 128 bit.

Example

```
> add snmp user edocs_user -group edocs_group
<!--NeedCopy-->
```

To configure an SNMP user by using the GUI

Navigate to **System** > **SNMP** > **Users**, and create the SNMP user.

Configuring SNMP Alarms for rate limiting

September 14, 2021

Citrix ADC appliances such as the Citrix ADC MPX 10500, 12500, and 15500 are rate limited. The maximum throughput (Mbps) and packets per second (PPS) are determined by the license purchased for the appliance. For rate-limited platforms, you can configure SNMP traps to send notifications when throughput and PPS approach their limits and when they return to normal.

Throughput and PPS are monitored every seven seconds. You can configure traps with high-threshold and normal-threshold values, which are expressed as a percentage of the licensed limits. The appliance then generates a trap when throughput or PPS exceeds the high threshold, and a second trap when the monitored parameter falls to the normal threshold. In addition to sending the traps to the configured destination device, the Citrix ADC logs the events associated with the traps in the /var/log/ns.log file as EVENT ALERTSTARTED and EVENT ALERTENDED.

Exceeding the throughput limit can result in packet loss. You can configure SNMP alarms to report packet loss.

For more information about SNMP alarms and traps, see "Configuring the Citrix ADC to generate SNMP v1 and v2 Traps."

This document includes the following details:

- Configuring an SNMP Alarm for Throughput or PPS
- Configuring SNMP Alarm for Dropped Packets

Configuring an SNMP alarm for throughput or PPS

To monitor both throughout and PPS, you must configure seperate alarms and set threshold pps value in Mbps.

To configure an SNMP alarm for the throughput rate by using the command line interface

At the command prompt, type the following commands to configure the SNMP alarm, set threshold value in Mbps and verify the configuration:

- `set snmp alarm PF-RL-RATE-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( **ENABLED** | **DISABLED** )] [-severity <severity>] [-logging ( **ENABLED** | **DISABLED** )]`
- `show snmp alarm PF-RL-RATE-THRESHOLD`

Example

```
1 > set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue
     50
2 <!--NeedCopy-->
```

To configure an SNMP alarm for PPS by using the command line interface

At the command prompt, type the following commands to configure the SNMP alarm for PPS and verify the configuration:

- `set snmp alarm PF-RL-PPS-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( **ENABLED** | **DISABLED** )] [-severity <severity>] [-logging ( **ENABLED** | **DISABLED** )]`
- `show snmp alarm PF-RL-PPS-THRESHOLD`

Example

```
1 > set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
2 <!--NeedCopy-->
```

To configure an SNMP alarm for throughput or PPS by using the GUI

1. Navigate to **System** > **SNMP** > **Alarms**, and select **PF-RL-RATE-THRESHOLD** (for throughput rate) or **PF-RL-PPS-THRESHOLD** (for packets per second).
2. Set the alarm parameters and enable the selected SNMP alarm.

Configuring SNMP alarm for dropped packets

You can configure an alarm for packets dropped as a result of exceeding the throughput limit and an alarm for packets dropped as a result of exceeding the PPS limit.

To configure an SNMP alarm for packets dropped because of excessive throughput, by using the command line interface

At the command prompt, type:

```
set snmp alarm PF-RL-RATE-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

To configure an SNMP alarm for packets dropped because of excessive PPS, by using the command line interface

At the command prompt, type:

```
set snmp alarm PF-RL-PPS-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

To configure an SNMP alarm for dropped packets by using the GUI

1. Navigate to **System** > **SNMP** > **Alarms**, and select **PF-RL-RATE-PKTS-DROPPED** (for packets dropped because of excessive throughput) or **PF-RL-PPS-PKTS-DROPPED** (for packets dropped because of excessive PPS).
2. Set the alarm parameters and enable the selected SNMP alarm.

Configuring SNMP in FIPS mode

September 14, 2021

FIPS mode requires Simple Network Management Protocol version 3 (SNMPv3) with the authentication and privacy (authPriv) option. SNMP version 1 and version 2 use a community string mechanism to provide secured access to management data. The community string is sent as clear text between an SNMP manager and an SNMP agent. This type of communication is unsecure, allowing intruders to access SNMP information on the network.

The SNMPv3 protocol uses the User-based Security Model (USM) and View-based Access Control Model (VACM) to authenticate and control management access to SNMP messaging data. SNMPv3 has three security levels: no authentication no privacy (noAuthNoPriv), authentication and no privacy (authNoPriv), and authentication and privacy (authPriv).

Enabling FIPS mode and restarting the Citrix ADC appliance removes the following SNMP configurations from the appliance:

1. Community configuration for SNMPv1 and SNMPv2 protocols.
2. SNMPv3 groups configured with the noAuthNoPriv or authNoPriv security-level option.

3. Traps configured for SNMPv1 or SNMPv2, or SNMPv3 with the noAuthNoPriv security-level option.

After restarting the appliance, configure SNMPv3 with the authPriv option. For more information about configuring authPriv option in SMNP v3, see SNMPV3 topic

> **Note**:
>
> Enabling FIPS mode and restarting your appliance blocks execution of the following SNMP trap and group commands:

```
1.  add snmp community <communityName> <permissions>

2.  add snmp trap <trapClass> <trapDestination> ... [-version: v1/
   v2]   [-td <positive_integer>] [-destPort <port>] [-
   communityName <string>] [-srcIP <ip_addr|ipv6_addr>] [-severity
   <severity>] [-allPartitions ( ENABLED | DISABLED )]

3.  add snmp group <name> <securityLevel : noAuthNoPriv/ authNoPriv
    > -readViewName <string>

4.  bind snmp trap specific <TrapIp>-userName <v3 user name> -
   securityLevel <noAuthNoPriv/ authNoPriv>
<!--NeedCopy-->
```

Audit logging

September 14, 2021

> **Important**
>
> Citrix recommends you to update a SYSLOG or NSLOG configuration only during maintenance or downtime. If you update a configuration after creating a session, the changes are not applied to the existing session logs.

Auditing is a methodical examination or review of a condition or situation. The audit logging feature enables you to log the Citrix ADC states and status information collected by various modules. The log information can be in the kernel and in the user-level daemons. For audit logging, you can use the SYSLOG protocol, the native NSLOG protocol, or both.

SYSLOG is a standard protocol for logging. It has two components:

- **SYSLOG auditing module.** Runs on the Citrix ADC appliance.
- **SYSLOG server.** Runs on the underlying FreeBSD operating system (OS) of the Citrix ADC appliance or on a remote system.

SYSLOG uses a user data protocol (UDP) for data transfer.

Similarly, the native NSLOG protocol has two components:

- **NSLOG auditing module.** Runs on the Citrix ADC appliance.
- **NSLOG server.** Runs on the underlying FreeBSD OS of the Citrix ADC appliance or on a remote system.

NSLOG uses TCP for data transfer.

When you run a SYSLOG or NSLOG server, it connects to the Citrix ADC appliance. The Citrix ADC appliance then starts sending all the log information to the SYSLOG or NSLOG server. And the server filters the log entries before storing them in a log file. An NSLOG or SYSLOG server receives log information from more than one Citrix ADC appliance. The Citrix ADC appliance sends log information to more than one SYSLOG server or NSLOG server.

If multiple SYSLOG servers are configured, the Citrix ADC appliance sends its SYSLOG events and messages to all the configured external log servers. It results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the Citrix ADC appliance offers load balancing algorithms. The appliance can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastPackets, and AuditlogHash.

> **Note**
>
> The Citrix ADC appliance can send audit log messages up to 16 KB to an external SYSLOG server.

The log information that a SYSLOG or NSLOG server collects from a Citrix ADC appliance is stored in a log file in the form of messages. These messages typically contain the following information:

- The IP address of a Citrix ADC appliance that generated the log message.
- A time stamp
- The message type
- The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency)
- The message information

To configure audit logging, you first configure the audit modules on the Citrix ADC appliance. The appliance involves creating audit policies and specifying the NSLOG server or SYSLOG server information. You then install and configure the SYSLOG or the NSLOG server on the underlying FreeBSD OS of the Citrix ADC appliance or on a remote system.

> **Note**
>
> SYSLOG is an industry standard for logging program messages, and various vendors provide support. The documentation does not include SYSLOG server configuration information.

The NSLOG server has its own configuration file (auditlog.conf). You can customize logging on the NSLOG server system by making extra modifications to the configuration file (auditlog.conf).

Configuring Citrix ADC appliance for audit logging

November 24, 2021

> **Warning:**
>
> Classic policy expressions and its usage are deprecated (discouraged from use but still supported) from Citrix ADC 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies.

Audit-logging displays status information from different modules so that an administrator can see event history in the chronological order. Main components of an Audit framework are 'audit action', 'audit policy'. 'Audit action' describes Audit Server configuration information whereas 'audit policy' links a bind entity to an 'audit action'. The audit policies use 'Classic Policy Engine'(CPE) framework or Progress Integration (PI) framework to link 'audit action' to 'system global bind entities'.

However, the policy frameworks differ from each other in binding audit-log policies to global entities. Previously, the audit module supported only Classic expression but now it supports both Classic and Advanced policy expressions. Currently, the Advanced expression can bind audit-log policies only to System global entities.

> **Note**
>
> When you bind a policy to global entities, you must bind it to a system global entity of the same expression. For example, you cannot bind a classic policy to an advanced global entity or bind an advanced policy to a classic global entity.
>
> Also, you cannot bind both classic audit-log policy and advanced audit-log policy to a load balancing virtual server.

Configuring audit-log policies in a Classic policy expression

Configuring audit-logging in Classic policy consists of the following steps:

1. **Configuring an audit-log action**. You can configure an audit action for different servers and for different log levels. 'Audit action' describes Audit Server configuration information whereas 'audit policy' links a bind entity to an 'audit action'. By default, the SYSLOG, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data. When using TCP for SYSLOG, you can set the buffer limit on the Citrix ADC appliance to store the logs. After which the logs are sent to the SYSLOG server.

2. **Configuring audit-log policy**. You can either configure SYSLOG policies to log messages to a SYSLOG server or NSLOG policy to log messages to an NSLOG server. Each policy includes a rule identifying the messages to be logged, and a SYSLOG or NS LOG action.
3. **Binding audit-log policies to global entities**. You must globally bind the audit log policies to global entities such SYSTEM, VPN, Citrix ADC AAA and so on. You can do it to enable logging of all Citrix ADC system events. By defining the priority level, you can set the evaluation order of the audit server logging. Priority 0 is the highest and is evaluated first. The higher the priority number, the lower is the priority of evaluation.

Each of these steps is explained in the following sections.

Configuring audit-log action

To configure SYSLOG action in Advanced Policy infrastructure by using the command line interface.

> **Note**
>
> The Citrix ADC appliance allows you to configure only one SYSLOG action to SYSLOG server IP address and port. The appliance does not allow you to configure multiple SYSLOG actions to the same server IP address and port.

A syslog action contains a reference to a syslog server. It specifies which information to log and mentions how to log that information.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]`
2 -  show audit syslogAction [<name>]
3
4 <!--NeedCopy-->
```

To configure NSLOG action in Advanced Policy infrastructure by using the command line interface

A ns log action contains a reference to a nslog server. It specifies which information to log and mentions how to log that information.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 -  show audit nslogAction [<name>]
3 <!--NeedCopy-->
```


Configuring audit-log policies

To configure audit-log Policies in Classic Policy infrastructure by using the command line interface

At the command prompt, type:

```
1 -  add audit syslogpolicy <name> <-rule> <action>
2 -  add audit nslogpolicy <name> < rule> <action>rm audit nslogpolicy <
     name>show audit nslogpolicy [<name>]set audit nslogpolicy <name>  [-
     rule <expression>] [-action <name>]
3 <!--NeedCopy-->
```

Binding audit syslog policies to audit syslog global

To bind audit-log policy in Classic policy framework by using the command line interface

At the command prompt, type:

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
```

```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```

Configuring audit-log policies using advanced policy expression

Configuring audit-logging in Advanced policy consists of the following steps:

1. **Configuring an audit-log action**. You can configure an audit action for different servers and for different log levels. 'Audit action' describes Audit Server configuration information whereas 'audit policy' links a bind entity to an 'audit action'. By default, the SYSLOG, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data. When using TCP for SYSLOG, you can set the buffer limit on the Citrix ADC appliance to store the logs. After which the logs are sent to the SYSLOG server.
2. **Configuring audit-log policy**. You can either configure SYSLOG policies to log messages to a SYSLOG server or NSLOG policy to log messages to an NSLOG server. Each policy includes a rule identifying the messages to be logged, and a SYSLOG or NS LOG action.
3. **Binding audit-log policies to global entities**. You must globally bind the audit log policies to SYSTEM global entity to enable logging of all Citrix ADC system events. By defining the priority level, you can set the evaluation order of the audit server logging. Priority 0 is the highest and is evaluated first. The higher the priority number, the lower is the priority of evaluation.

> **Note**
>
> The Citrix ADC appliance evaluates all the policies that are bind to true.

Configuring audit-log action

To configure syslog action in Advanced Policy infrastructure by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]
2 -  show audit syslogAction [<name>]
3 <!--NeedCopy-->
```

To configure NSLOG action in Advanced Policy infrastructure by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
1 -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 -  show audit nslogAction [<name>]
3 <!--NeedCopy-->
```

Configuring audit-log policies

To add a syslog audit action by using the command line interface

At the command prompt, type:

```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>])
2     | -lbVserverName <string>))[-serverPort <port>] -logLevel <logLevel
          >[-dateFormat <dateFormat>]
3      [-logFacility <logFacility>][-tcp ( NONE | ALL )] [-acl ( ENABLED
           | DISABLED )]
4     [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES |
          NO )]
5      [-appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED
           )][-alg ( ENABLED | DISABLED )]
6     [-subscriberLog ( ENABLED | DISABLED )][-transport ( TCP | UDP )]
          [-tcpProfileName <string>][-maxLogDataSizeToHold
7 <!--NeedCopy-->
```

Example


```
1      > add audit syslogaction audit-action1 10.102.1.1 -loglevel
          INFORMATIONAL -dateformat MMDDYYYY
2      > add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -
          loglevel INFORMATIONAL -dateFormat MMDDYYYY
3      > add audit syslogpolicy syslog-pol1 TRUE audit-action1
4      > add audit nslogPolicy nslog-pol1 TRUE nslog-action1
5      > bind system global nslog-pol1 -priority 20
6  <!--NeedCopy-->
```

To add a nslog audit action by using the command line interface

At the command prompt, type:

```
1      add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
          domainResolveRetry <integer>])) [-serverPort <port>]       -
          logLevel <logLevel> ... [-dateFormat <dateFormat>][-logFacility
          <logFacility>] [-tcp ( NONE | ALL )][-acl ( ENABLED | DISABLED )
          ] [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog (
          YES | NO )][-appflowExport ( ENABLED | DISABLED )] [-lsn (
          ENABLED | DISABLED )][-alg ( ENABLED | DISABLED )] [-
          subscriberLog ( ENABLED | DISABLED )]`
2  <!--NeedCopy-->
```

Binding audit-log policies to global entities

To bind syslog audit-log policy in Advanced policy framework by using the command line interface

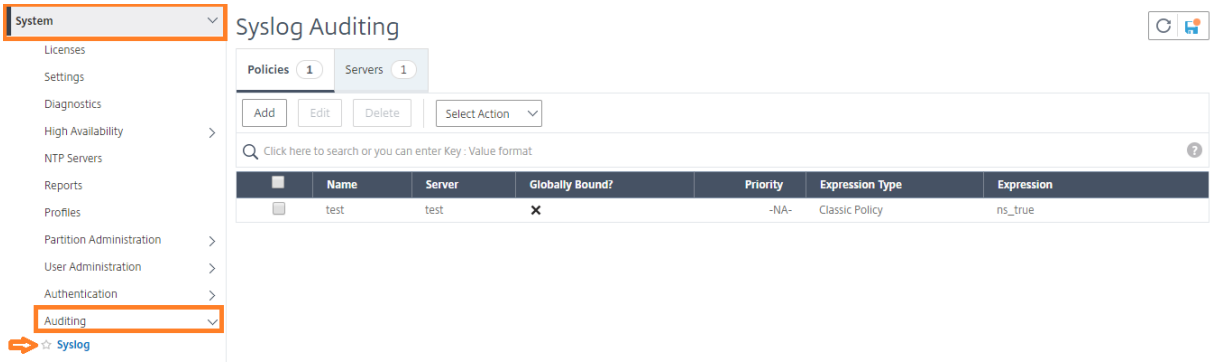
At the command prompt, type:

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
```

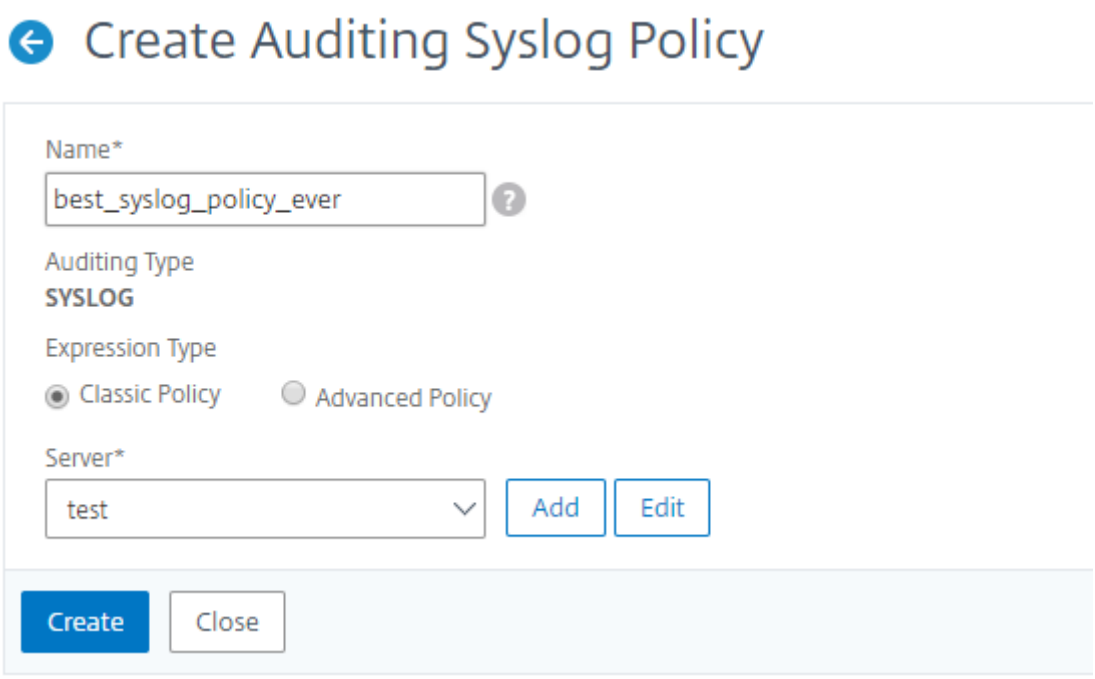
```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```

Configuring audit-log policy by using the GUI

1. Navigate to **Configuration** > **System** > **Auditing** > **Syslog**.



1. Select **Servers** tab.
2. Click **Add**.
3. In the **Create Auditing Server** page, populate the relevant fields, and click **Create**.
4. To add the policy, select the **Policies** tab, and click **Add**.
5. In the **Create Auditing Syslog Policy** page, populate the relevant fields, and click **Create**.



6. To bind the policy globally, select **Advanced Policy Global Bindings** from the drop-down list. Select the **best_syslog_policy_ever** policy. Click **Select**.
7. From the drop-down list, select the bind point as **SYSTEM_GLOBAL** and click **Bind**, and then click **Done**.

Configuring policy-based logging

You can configure policy-based logging for rewrite and responder policies. Audit messages are then logged in a defined format when the rule in a policy evaluates to TRUE. To configure policy-based logging, you configure an audit-message action that uses default syntax expressions to specify the format of the audit messages. And associate the action with a policy. The policy can be bound either globally or to a load balancing or content switching virtual server. You can use audit-message actions to log messages at various log levels, either in syslog format only or in both syslog and new nslog formats

Prerequisites

- User Configurable Log Messages (userDefinedAuditlog) option is enabled for when configuring the audit action server to which you want to send the logs in a defined format.
- The related audit policy is bound to system global.

Configuring an audit message action

You can configure audit message actions to log messages at various log levels, either in syslog format only or in both syslog and new ns log formats. Audit-message actions use expressions to specify the format of the audit messages.

To create an audit message action by using the command line interface

At the command prompt, type:

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-
    logtoNewnslog (YES|NO)] [-bypassSafetyCheck (YES|NO)]
<!--NeedCopy-->
```

```
add audit messageaction log-act1 CRITICAL '"Client:"+CLIENT.IP.SRC+"
    accessed "+HTTP.REQ.URL' -bypassSafetyCheck YES
<!--NeedCopy-->
```

To configure an audit message action by using the GUI

Navigate to **System > Auditing > Message Actions**, and create the audit message action.

Binding audit message action to a policy

After you have created an audit message action, you must bind it to a rewrite or responder policy. For more information about binding log message actions to a rewrite or responder policy, see Rewrite or Responder.

Installing and configuring the NSLOG server

September 14, 2021

During installation, the NSLOG server executable file (auditserver) is installed along with other files. The auditserver executable file includes options for performing several actions on the NSLOG server,

including running and stopping the NSLOG server. In addition, you use the auditserver executable to configure the NSLOG server with the IP addresses of the Citrix ADC appliances from which the NSLOG server will start collecting logs. Configuration settings are applied in the NSLOG server configuration file (auditlog.conf).

Then, you start the NSLOG server by executing the auditserver executable. The NSLOG server configuration is based on the settings in the configuration file. You can further customize logging on the NSLOG server system by making additional modifications to the NSLOG server configuration file (auditlog.conf).

> **Attention**:
>
> The version of the NSLOG server package must be the same as that of the Citrix ADC. For example, if the version of the Citrix ADC is 10.1 Build 125.9, the NSLOG server must also be of the same version.

The following table lists the operating systems on which the NSLOG server is supported.

Operating system	Software requirements	Remarks
Windows	Windows XP Professional, Windows Server 2003, Windows 2000/NT, Windows Server 2008, Windows Server 2008 R2	
Linux	RedHat Linux 4 or later, SUSE Linux Enterprise 9.3 or later	
FreeBSD	FreeBSD 6.3 or later	For Citrix ADC 10.5, use only FreeBSD 8.4.
Mac OS	Mac OS 8.6 or later	Not supported on Citrix ADC 10.1 and later releases.

The minimum hardware specifications for the platform running the NSLOG server are as follows:

- Processor- Intel x86 ~501 megahertz (MHz)
- RAM - 512 megabytes (MB)
- Controller - SCSI

Installing NSLOG server on the Linux operating system

Log on to the Linux system as an administrator. Use the following procedure to install the NSLOG server executable files on the system.

To install the NSLOG server package on a Linux operating system

1. At a Linux command prompt, type the following command to copy the NSauditserver.rpm file to a temporary directory:

   ```
   cp <path_to_cd>/Utilities/auditserver/Linux/NSauditserver.rpm /tmp
   ```

2. Type the following command to install the NSauditserver.rpm file.

   rpm -i NSauditserver.rpm

   This command extracts the files and installs them in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

To uninstall the NSLOG server package on a Linux operating system

1. At a command prompt, type the following command to uninstall the audit server logging feature:

   ```
   rpm -e NSauditserver
   ```

2. For more information about the NSauditserver RPM file, use the following command:

   ```
   rpm -qpi \*.rpm
   ```

3. To view the installed audit server files use the following command:

   ```
   rpm -qpl *.rpm
   ```

   *.rpm: Specifies the file name.

Installing NSLOG server on the FreeBSD operating system

Before you can install the NSLOG server, you have to copy the NSLOG package from the Citrix ADC product CD or download it from www.citrix.com. The NSLOG package has the following name format:

```
AuditServer_<release number>-<build number>.zip
```

For example: `AuditServer_10.5-58.11.zip`

This package contains files for all supported platforms: Linux, Windows, and FreeBSD. On a FreeBSD operating system, install the NSLOG package that has the following name format:

```
audserver_bsd-<release number>-<build number>.tgz
```

For example: `audserver_bsd-10.5-58.11.tgz`

To download NSLOG package from www.citrix.com:

1. In a web browser, go to www.citrix.com.
2. In the menu bar, click **Log In**.
3. Enter your login credentials, and then click **Log In**.
4. In the menu bar, click **Downloads**.
5. From the **Select a product** list, select **Citrix ADC**.
6. On the **Citrix ADC** page, select the release for which you want to download the NSLOG package (for example, Release 10.5), and then select **Firmware**.
7. Under **Firmware**, select the Citrix ADC firmware for the build number for which you want to download the NSLOG package.
8. On the page that appears, scroll down, select **Audit Servers**, and click **Download File** next to the package that you want to download.

To install the NSLOG server package on a FreeBSD operating system

1. On the system to which you have downloaded the NSLOG package `AuditServer_<release number>-<build number>.zip` (for example, `AuditServer_9.3-51.5.zip`), extract the `FreeBSD NSLOG server package audserver_bsd-<release number>-<build number>.tgz` (for example, `audserver_bsd-9.3-51.5.tgz`) from the package.
2. Copy the FreeBSD NSLOG server package `audserver_bsd-<release number>-<build number>.tgz` (for example, `audserver_bsd-9.3-51.5.tgz`) to a directory on a system running FreeBSD OS.
3. At a command prompt for the directory into which the FreeBSD NSLOG server package was copied, run the following command to install the package:

   `pkg_add audserver_bsd-<release number>-<build number>.tgz`

   Example:

   ```
   pkg_add audserver_bsd-9.3-51.5.tgz
   <!--NeedCopy-->
   ```

   The following directories are extracted:

   - `<root directory extracted from the FreeBSD NSLOG server package tgz file>Citrix ADCbin` (for example, `/var/auditserver/netscaler/bin`)
   - `<root directory extracted from the FreeBSD NSLOG server package tgz file>netscaler/etc` (for example, `/var/auditserver/netscaler/etc`)
   - `<root directory extracted from the FreeBSD NSLOG server package tgz file>\netscaler\samples` (for example, `/var/auditserver/samples`)
4. At a command prompt, type the following command to verify that the package is installed:

   `pkg_info | grep NSaudserver`

To uninstall the NSLOG server package on a FreeBSD operating system

At a command prompt, type:

```
pkg_delete NSaudserver
```

Installing NSLOG Server Files on the Windows Operating System

Before you can install the NSLOG server, you have to copy the NSLOG package from the Citrix ADC product CD or download it from www.citrix.com. The NSLOG package has the following name format `AuditServer _<release number>-<build number>.zip` (for example, `AuditServer_9.3-51.5.zip`). This package contains NSLOG installation packages for all supported platforms.

To download NSLOG package from www.Citrix.com

1. In a web browser, go to www.citrix.com.
2. In the menu bar, click Log In.
3. Enter your login credentials, and then click Log In.
4. In the menu bar, click Downloads.
5. Search to find the page that provides the appropriate release number and build.
6. On that page, under Audit Servers, click Download to download the NSLOG package, having the format `AuditServer_<release number>-<build number>.zip`, to your local system (for example, `AuditServer_9.3-51.5.zip`).

To install NSLOG server on a Windows operating system

1. On the system, where you have downloaded the NSLOG package `AuditServer_<release number>-<build number>.zip` (for example, `AuditServer_9.3-51.5.zip`), extract `audserver_win-<release number>-<build number>.zip` (for example, `audserver_win-9.3-51.5.zip`) from the package.
2. Copy the extracted file `audserver_<release number>-<build number>.zip` (for example, `audserver_win-9.3-51.5.zip`) to a Windows system on which you want to install the NSLOG server.
3. Unzip the `audserver_<release number>-<build number>.zip` file (for example, `audserver_win-9.3-51.5.zip`).
4. The following directories are extracted:
   a) `<root directory extracted from the Windows NSLOG server package zip file>\bin`(for example, `C:\audserver_win-9.3-51.5\bin`)
   b) `<root directory extracted from the Windows NSLOG server package zip file>\etc` (for example, `C:\audserver_win-9.3-51.5\etc`)

c) <root directory extracted from the Windows NSLOG server **package** zip file>\samples (for example, C:\audserver_win-9.3-51.5\samples)

5. At a command prompt, run the following command from the <root directory extracted from the Windows NSLOG server **package** zip file>\bin path

   audserver -install -f <directorypath>\auditlog.conf

   <directorypath>: Specifies the path to the configuration file (auditlog.conf). By default, log.conf is under \<root directory extracted from Windows NSLOG server **package** zip file\>\samples directory. But you can copy auditlog.conf to your desired directory.

To uninstall the NSLOG server on a Windows operating system

At a command prompt, run the following from the <root directory extracted from Windows NSLOG server **package** zip file>\bin path:

```
audserver -remove
```

NSLOG Server Command Options

For information about NSLOG server commands, see Audit Server Options.

Run the audserver command from the directory in which the audit server executable is present:

- On Windows: \ns\bin
- On Solaris and Linux:\usr\local\netscaler\bin

The audit server configuration files are present in the following directories:

- On Windows: \ns\etc
- On Linux: \usr\local\netscaler\etc

The audit server executable is started as ./auditserver in Linux and FreeBSD.

Adding the Citrix ADC Appliance IP Addresses on the NSLOG Server

In the configuration file (auditlog.conf), add the IP addresses of the Citrix ADC appliances whose events must be logged.

To add the IP addresses of the Citrix ADC appliance

At a command prompt, type the following command:

```
audserver -addns -f <directorypath\>\auditlog.conf
```


<directorypath>: Specifies the path to the configuration file (auditlog.conf).

You are prompted to enter the information for the following parameters:

NSIP: Specifies the IP address of the Citrix ADC appliance, for example, 10.102.29.1.

Userid: Specifies the user name, for example, nsroot.

Password: Specifies the password, for example, nsroot.

If you add multiple Citrix ADC IP addresses (NSIP), and later you do not want to log all of the Citrix ADC appliance event details, you can delete the NSIPs manually by removing the NSIP statement at the end of the auditlog.conf file. For a high availability (HA) setup, you must add both primary and secondary Citrix ADC IP addresses to auditlog.conf by using the audserver command. Before adding the IP address, make sure the user name and password exist on the system.

Verifying the NSLOG Server Configuration File

Check the configuration file (audit log.conf) for syntax correctness to enable logging to start and function correctly.

To verify configuration, at a command prompt, type the following command:

```
audserver -verify -f <directorypath>\auditlog.conf
```

<directorypath>: Specifies the path to the configuration file (audit log.conf).

Running the NSLOG server

September 14, 2021

To start audit server logging

Type the following command at a command prompt:

audserver -start -f <directorypath>\auditlog.conf

<directorypath>: Specifies the path to the configuration file (audit log.conf).

To stop audit server logging that starts as a background process in FreeBSD or Linux

Type the following command:

```
audserver -stop
```


To stop audit server logging that starts as a service in Windows

Type the following command:

```
audserver -stopservice
```

Customizing logging on the NSLOG server

September 14, 2021

You can customize logging on the NSLOG server by making additional modifications to the NSLOG server configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the server system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information from a Citrix ADC appliance or a set of Citrix ADC appliances.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

This document includes the following details:

- Creating Filters
- Specifying Log Properties

Creating filters

You can use the default filter definition located in the configuration file (audit log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

> Note:
> For consolidated logging, if a log transaction occurs for which there is no filter definition, the default filter is used (if it is enabled.) The only way you can configure consolidated logging of all the Citrix ADC appliances is by defining the default filter.

To create a filter

At the command prompt, type the following command in the configuration file (auditlog.conf):

```
1 filter <filterName> [IP <ip>] [NETMASK <mask>] ON | OFF]
2 <!--NeedCopy-->
```

filterName: Specify the name of the filter (maximum of 64 alphanumeric characters).

ip: Specify the IP addresses.

mask: Specify the subnet mask to be used on a subnet.

Specify ON to enable the filter to log transactions, or specify OFF to disable the filter. If no argument is specified, the filter is ON.

Examples:

```
filter F1 IP 192.168.100.151 ON
<!--NeedCopy-->
```

To apply the filter F2 to IP addresses 192.250.100.1 to 192.250.100.254:

```
filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON
<!--NeedCopy-->
```

filterName is a required parameter if you are defining a filter with other optional parameters, such as IP address, or the combination of IP address and Netmask.

Specifying log properties

Log properties associated with the filter are applied to all the log entries present in the filter. The log property definition starts with the key word BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
    logFilenameFormat ...
    logDirectory ...
    logInterval ...
    logFileSizeLimit ....
END
<!--NeedCopy-->
```

Entries in the definition can include the following:

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:
  - Static: A constant string that specifies the absolute path and the file name.
  - Dynamic: An expression that includes the following format specifiers:
    - Date (%{format}t)
    - creates file name with NSIP

  Example:


```
LogFileNameFormat Ex%`{
`%m%d%y }
t.log
<!--NeedCopy-->
```

This creates the first file name as Exmmddyy.log. New files are named: Exmmddyy.log.0, Exmmddyy.log.1, and so on. In the following example, the new files are created when the file size reaches 100MB.

Example:

```
LogInterval size
LogFileSize 100
LogFileNameFormat Ex%`{
`%m%d%y }
t
<!--NeedCopy-->
```

> **Caution**
>
> The date format %t specified in the LogFilenameFormat parameter overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat parameter.

- **logDirectory** specifies the directory name format of the log file. The name of the file can be either of the following:
  - Static: Is a constant string that specifies the absolute path and file name.
  - Dynamic: Is an expression containing the following format specifiers:
    - Date (%{format}t)
    - creates directory with NSIP

  The directory separator depends on the operating system. In Windows, use the directory separator.

  Example:

  ```
  LogDirectory dir1\dir2\dir3
  <!--NeedCopy-->
  ```

  In the other operating systems (Linux, FreeBsd, and so forth.), use the directory separator.

- **LogInterval** specifies the interval at which new log files are created. Use one of the following values:
  - Hourly: A file is created every hour. Default value.

- Daily: A file is created very day at midnight.
- Weekly: A file is created every Sunday at midnight.
- Monthly : A file is created on the first day of the month at midnight.
- None: A file is created only once, when audit server logging starts.
- Size: A file is created only when the log file size limit is reached.

Example:

```
1  LogInterval Hourly
2  <!--NeedCopy-->
```

- **LogFileSizeLimit** specifies the maximum size (in MB) of the log file. A new file is created when the limit is reached.

> **Note**
>
> You can override the loginterval property by assigning size as its value.

The default LogFileSizeLimit is 10 MB.

Example:

```
1  LogFileSizeLimit 35
2  <!--NeedCopy-->
```

SYSLOG Over TCP

September 14, 2021

Syslog is a standard for sending event notification messages. These messages can be stored locally or on an external log server. Syslog enables network administrators to consolidate log messages and derive insights from the collected data.

Syslog is originally designed to work over UDP, which can transmit a huge amount of data within the same network with minimal packet loss. However, telco operators prefer to transmit syslog data over TCP, because they need reliable, ordered data transmission between networks. For example, telco tracks user activities, and TCP provides retransmission in the event of network failure.

How Syslog over TCP works

To understand how syslog over TCP works, consider two hypothetical cases:

Sam, a network administrator, wants to log significant events on an external syslog server.

XYZ Telecom, an ISP, has to transmit and store a significant amount of data on syslog servers to comply with government regulations.

In both cases, the log messages must be transmitted over a reliable channel and stored safely on an external syslog server. Unlike UDP, TCP establishes a connection, transmits messages securely, and retransmits (from sender to receiver) any data that is corrupted or lost because of network failure.

The Citrix ADC appliance sends log messages over UDP to the local syslog daemon, and sends log messages over TCP or UDP to external syslog servers.

SNIP support for Syslog

When the audit-log module generates syslog messages, it uses a Citrix ADC IP (NSIP) address as the source address for sending the messages to an external syslog server. To configure a SNIP as the source address, you must make it part of the netProfile option and bind the netProfile to the syslog action.

> **Note**
>
> TCP uses SNIP for sending monitoring probes to check the connectivity and then sends the logs over NSIP. Hence the syslog server must be reachable via SNIP. Net profiles can be used to redirect all the TCP syslog traffic through SNIP entirely.

Use of a SNIP address is not supported in internal logging.

Fully qualified domain name Support for audit Log

Previously, the audit-log module was configured with the destination IP address of the external syslog server to which the log messages are sent. Now, the audit-log server uses a fully qualified domain name (FQDN) instead of the destination IP address. The FQDN configuration resolves the configured domain name of the syslog server to the corresponding destination IP address for sending the log messages from the audit-log module. The name server must be properly configured to resolve the domain name and avoid domain based service issues.

> **Note**
>
> When configuring an FQDN, server domain name configuration of the same Citrix ADC appliance in syslog action or nslog action is not supported.

Configuring Syslog over TCP by using the Command Line Interface

To configure a Citrix ADC appliance to send syslog messages over TCP by using the command line interface

At the command prompt, type:


```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>]) | -lbVserverName<string>))[-
          serverPort <port>] -logLevel <logLevel>[-dateFormat <dateFormat
          >] [-logFacility <logFacility>]   [-tcp ( NONE | ALL )] [-acl (
          ENABLED | DISABLED )][-timeZone ( GMT_TIME | LOCAL_TIME )][-
          userDefinedAuditlog ( YES | NO )][-appflowExport ( ENABLED |
          DISABLED )] [-lsn ( ENABLED | DISABLED )][-alg ( ENABLED |
          DISABLED )] [-subscriberLog ( ENABLED | DISABLED )][-transport (
           TCP | UDP )] [-tcpProfileName <string>][-maxLogDataSizeToHold <
          positive_integer>][-dns ( ENABLED | DISABLED )] [-netProfile <
          string>]
2 <!--NeedCopy-->
```

```
1     add audit syslogaction audit-action1 10.102.1.1 -loglevel
          INFORMATIONAL -dateformat MMDDYYYY -transport TCP
2 <!--NeedCopy-->
```

Adding SNIP IP address to net profile option by using the command line interface

To add a SNIP IP address to the net profile by using the command line interface

At the command prompt, type:

```
1     add netProfile <name> [-td <positive_integer>] [-srcIP <string>][-
          srcippersistency ( ENABLED | DISABLED )][-overrideLsn ( ENABLED
          | DISABLED )]add syslogaction <name> <serverIP>  - loglevel all
           - netprofile net1
2 <!--NeedCopy-->
```

```
1     add netprofile net1  - srcip 10.102.147.204`
2 <!--NeedCopy-->
```

Where, srcIP is the SNIP.

Adding net profile in a syslog action by using the command line interface

To add a netProfile option in a syslog action by using the command line interface

At the command prompt, type:

```
1      add audit syslogaction <name> (<serverIP> | -lbVserverName <string
           >)  -logLevel <logLevel>
2     -netProfile <string> …
```



```
3
4 <!--NeedCopy-->
```

```
1     add syslogaction sys_act1 10.102.147.36 -loglevel all -netprofile
          net1
2 <!--NeedCopy-->
```

Where, -netprofile specifies the name of the configured net profile. The SNIP address is configured as part of the netProfile and this netProfile option is bound to the syslog action.

> **Note**
>
> You must always bind the netProfile option to the SYSLOGUDP or SYSLOGTCP services bound to the SYSLOGUDP or SYSLOGTCP load balancing virtual server, when an LB virtual server name is configured in syslog action.

Configuring FQDN support by using the command line interface

To add a server domain name to a Syslog action by using the command line interface

At the command prompt, type:

```
1 add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
     domainResolveRetry <integer>]) | -lbVserverName <string>)) -logLevel
      <logLevel> ...
2     set audit syslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>]-
        serverDomainName <string>] [-lbVserverName <string>]-
        domainResolveRetry <integer>] [-domainResolveNow]
3 <!--NeedCopy-->
```

To add a server domain name to a Nslog action by using the command line interface.

At the command prompt, type:

```
1     add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
        domainResolveRetry <integer>]))  -logLevel <logLevel> ...
2     set audit nslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>][-
        serverDomainName <string>] [-domainResolveRetry <integer>][-
        domainResolveNow]
3 <!--NeedCopy-->
```

Where serverDomainName. Domain name of the log server. Is mutually exclusive with serverIP/ lbVserverName.

DomainResolveRetry integer. Time (in seconds) that the Citrix ADC appliance waits, after a DNS resolution fails, before sending the next DNS query to resolve the domain name.

DomainResolveNow. Included if the DNS query has to be sent immediately to resolve the server's domain name.

Configuring Syslog over TCP by using the GUI

To configure the Citrix ADC appliance to send Syslog messages over TCP by using the GUI

1. Navigate to **System** > **Auditing** > **Syslog** and select the **Servers** tab.
2. Click **Add** and select Transport Type as **TCP**.

Configuring a net profile for SNIP support by using the GUI

To configure net profile for SNIP support by using the GUI

1. Navigate to **System > Auditing > Syslog** and select the **Servers** tab.
2. Click **Add** and select a net profile from the list.

Configuring FQDN by using the GUI

To configure FQDN by using the GUI

1. Navigate to **System** > **Auditing** > **Syslog** and select the **Servers** tab.
2. Click **Add** and select a Server Type and Server Domain Name from the list.

Load balancing SYSLOG servers

September 14, 2021

The Citrix ADC appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the Citrix ADC appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

```
add service <name>(<IP> | <serverName>)<serviceType (SYSLOGTCP |
SYSLOGUDP)> <port>
```


2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGUDP, and load balancing method as AUDITLOGHASH.

   ```
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
   <AUDITLOGHASH>]
   ```

3. Bing the service to the load balancing virtual server.

   ```
   Bind lb vserver <name> <serviceName>
   ```

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   ```
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
   ```

5. Add a SYSLOG policy by specifying the rule and action.

   ```
   add syslogpolicy <name> <rule> <action>
   ```

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   ```
   bind system global <policyName>
   ```

Load balancing of SYSLOG servers using the GUI

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

   Navigate to **Traffic Management** > **Services**, click **Add** and select **SYLOGTCP** or **SYSLOGUDP** as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

   Navigate to **Traffic Management** > **Virtual Servers**, click **Add** and select **SYLOGTCP** or **SYSLOGUDP** as protocol.

3. Bing the service to the load balancing virtual server to the service.

   Bing the service to the load balancing virtual server.

   Navigate to **Traffic Management** > **Virtual Servers**, select a virtual server and then select **AUDITLOGHASH** in the **Load Balancing Method.**

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   Navigate to **System** > **Auditing**, click **Servers** and add a server by selecting **LB Vserver** option in **Servers**.

5. Add a SYSLOG policy by specifying the rule and action.

   Navigate to **System** > **Syslog**, click **Policies** and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   Navigate to **System** > **Syslog**, select a SYSLOG policy and click **Action**, and then click **Global Bindings** and bind the policy to system global.

Example:

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. AUDITLOGHASH method load balances the traffic based on input hash value from the audit agents. The agents are the modules which generate auditlog in a Citrix ADC appliance. For example, if an agent LSN wants to load balance auditlogs based on client IP address, LSN module generates the hash value based on clientIP and passes the hash value to auditlog module. The auditlog module sends the auditlog messages which have same hash value to the external syslog server.

The Citrix ADC appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
<!--NeedCopy-->
```

Limitations:

- The Citrix ADC appliance does not support an external load balancing virtual server load balancing the SYSLOG messages among the log servers.

Default settings for the log properties

September 14, 2021

The following is an example of the default filter with default settings for the log properties:

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
```



```
 logFilenameFormat  auditlog%`{
 `%y%m%d }
 t.log
 end default
<!--NeedCopy-->
```

Following are two examples of defining the default filters:

Example 1:

```
Filter f1 IP 192.168.10.1
<!--NeedCopy-->
```

This creates a log file for NSI 192.168.10.1 with the default values of the log in effect.

Example 2:

```
Filter f1 IP 192.168.10.1
begin f1
    logFilenameFormat logfiles.log
end f1
<!--NeedCopy-->
```

This creates a log file for NSIP 192.168.10.1. Since the log file name format is specified, the default values of the other log properties are in effect.

Sample configuration file (audit.conf)

September 14, 2021

Following is a sample configuration file:

```
##############################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##############################
MYIP <NSAuditserverIP>
MYPORT 3023
#   Filter filter_nsip  IP <Specify the Citrix ADC IP address to filter
    on > ON
#   begin filter_nsip
#       logInterval         Hourly
#       logFileSizeLimit    10
```



```
#       logDirectory          logdir\%A\
#       logFilenameFormat   nsip%\\{
 \\%d%m%Y }
 t.log
#   end filter_nsip
Filter default
begin default
    logInterval      Hourly
    logFileSizeLimit    10
    logFilenameFormat   auditlog%\{
 \%y%m%d }
 t.log
end default
<!--NeedCopy-->
```

Web server logging

September 14, 2021

You can use the Web server logging feature to send logs of HTTP and HTTPS requests to a client system for storage and retrieval. This feature has two components:

- The Web log server, which runs on the Citrix ADC.
- The Citrix ADC Web Logging (NSWL) client, which runs on the client system.

When you run the Citrix ADC Web Logging (NSWL) client:

1. It connects to the Citrix ADC.
2. The Citrix ADC buffers the HTTP and HTTPS request log entries before sending them to the client.
3. The client can filter the entries before storing them.

To configure Web server logging, you first enable the Web logging feature on the Citrix ADC and configure the size of the buffer for temporarily storing the log entries. Then, you install NSWL on the client system. You then add the Citrix ADC IP address (NSIP) to the NSWL configuration file. You are now ready to start the NSWL client to begin logging. You can customize Web server logging by making additional modifications to the NSWL configuration file (log.conf).

Configuring the Citrix ADC for web server logging

September 14, 2021

To configure the Citrix ADC for web server logging you are required to only enable the Web Server Logging feature. Optionally, you can perform the following configurations:

- Modify the size of the buffer (default size is 16 MB) that stores the logged information before it is sent to the Citrix ADC Web Logging (NSWL) client.
- Specify the custom HTTP headers that you want to export to the NSWL client. You can configure a maximum of two HTTP request and two HTTP response header names.

To configure web server logging by using the command line interface

At the command prompt, perform the following operations:

- Enable the web server logging feature.

  ```
  enable ns feature WL
  ```

- [Optional] Modify the buffer size for storing the logged information.

  ```
  set ns weblogparam -bufferSizeMB <size>
  ```

  > **Note**:
  >
  > To activate your modification, you must disable and then re-enable the Web server logging feature.

- [Optional] Specify the custom HTTP header names that you want to export.

  ```
  set ns weblogparam [-customReqHdrs <string> ...] [-customRspHdrs <
  string> ...]
  ```

```
1     > enable ns feature WL
2     Done
3     > set ns weblogparam -bufferSizeMB 60
4     Done
5     > show ns weblogparam
6         Web Logging parameters:
7         Log buffer size:  60MB
8         Custom HTTP request headers: (none)
9         Custom HTTP response headers: (none)
10    Done
11    > set ns weblogparam -customReqHdrs req1 req2 -customRspHdrs res1
          res2
12    Done
13    > show ns weblogparam
14        Web Logging parameters:
15        Log buffer size:  60MB
16        Custom HTTP request headers: req1, req2
17        Custom HTTP response headers: res1, res2
```



```
18       Done
19  <!--NeedCopy-->
```

To configure web server logging by using the GUI

1. Navigate to **System** > **Settings** and perform the following operations:
   a) To enable the web server logging feature, click **Change Advanced Features** and select **Web Logging**.
   b) To modify the buffer size, click **Change Global System Settings** and under **Web Logging**, enter the buffer size.
   c) To specify the custom HTTP headers to be exported, click **Change Global System Settings** and under **Web Logging**, specify the header values.

Installing the Citrix ADC web logging (NSWL) client

September 14, 2021

When you install NSWL, the client executable file (NSWL) is installed along with other files. The NSWL executable file provides a list of options that you can use. For details, see Configuring the NSWL Client.

> **Attention**
>
> The version of the NSWL client must be the same as Citrix ADC. For example, if the version of the Citrix ADC is 10.1 Build 125.9, the NSWL client must also be of the same version. Also, the web logging (NSWL) client works on both 32 bit and on 64 bit server machines. The download page has only a 32 bit weblog client. The 64 bit weblog client is available on request, and recommends you to contact Citrix support for more information.

The following table lists the operating systems on which the NSWL client can be installed.

Operating system	Version	Hardware requirements	Remarks
Windows	Windows Server 2016 or later	Processor - x86/amd64 CPU (1 GHz or higher), RAM - 4 GB (or higher)	
macOS	macOS 8.6 or later	Not supported on Citrix ADC 10.1 and later releases.	

Operating system	Version	Hardware requirements	Remarks
Linux	Ubuntu, SUSE Linux, CentOS, Red Hat Enterprise Linux released in 2016 or later	Processor - x86/amd64 CPU (1 GHz or higher), RAM - 4 GB (or higher)	
Solaris	Solaris Sun OS 5.6 or later	Processor - UltraSPARC-IIi 400 MHz, RAM - 512 MB, Controller - SCSI	Not supported on Citrix ADC 10.5 and later releases.
FreeBSD	FreeBSD 6.3 or later	Processor - x86/amd64 CPU (1 GHz or higher), RAM - 4 GB (or higher)	For Citrix ADC 10.5, use only FreeBSD 8.4.
AIX	AIX 6.1	-	Not supported on Citrix ADC 10.5 and later releases.

If the NSWL client system cannot process the log transaction because of a CPU limitation, the Web log buffer overruns and the logging process reinitiates.

> **Caution**
>
> Reinitiation of logging can result in loss of log transactions.

To temporarily solve an NSWL client system bottleneck caused by a CPU limitation, you can tune the Web server logging buffer size on the Citrix ADC appliance. To solve the problem, you need a client system that can handle the site's throughput.

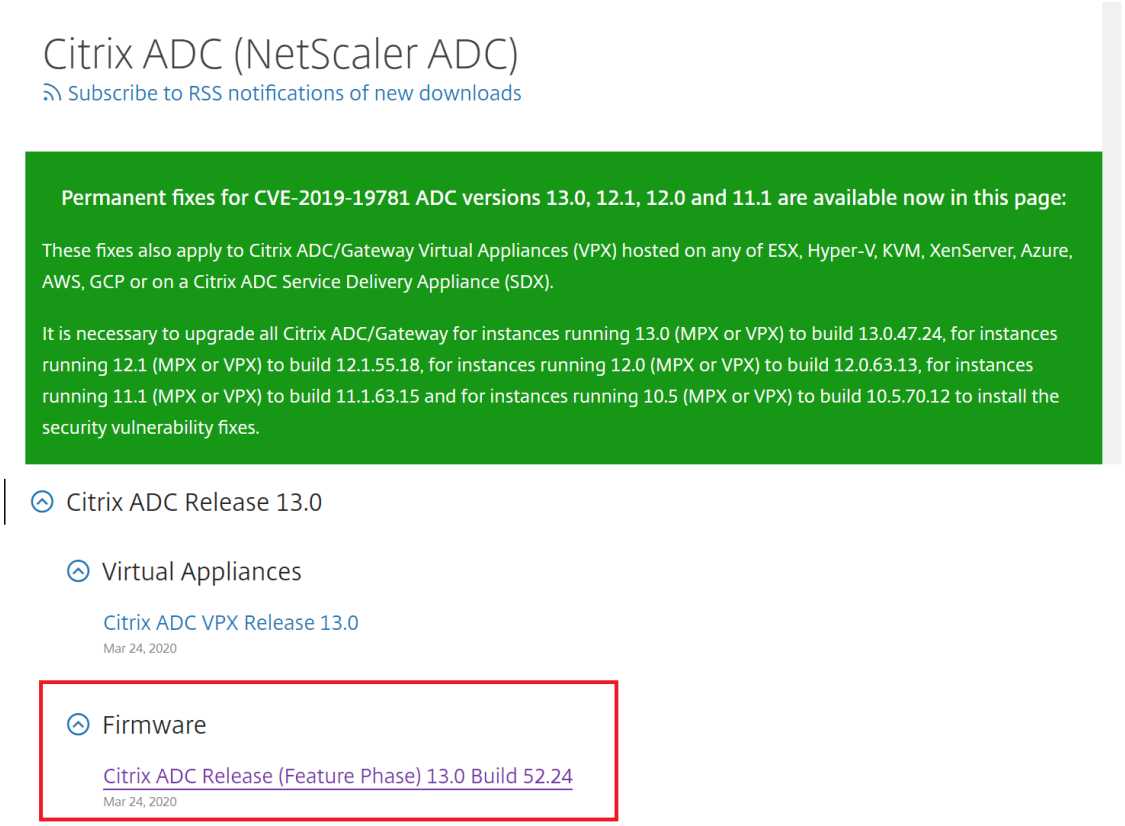
Download NSWL client

You can obtain the NSWL client package from either the Citrix ADC product CD or the Citrix downloads site. Within the package there are separate installation packages for each supported platform.

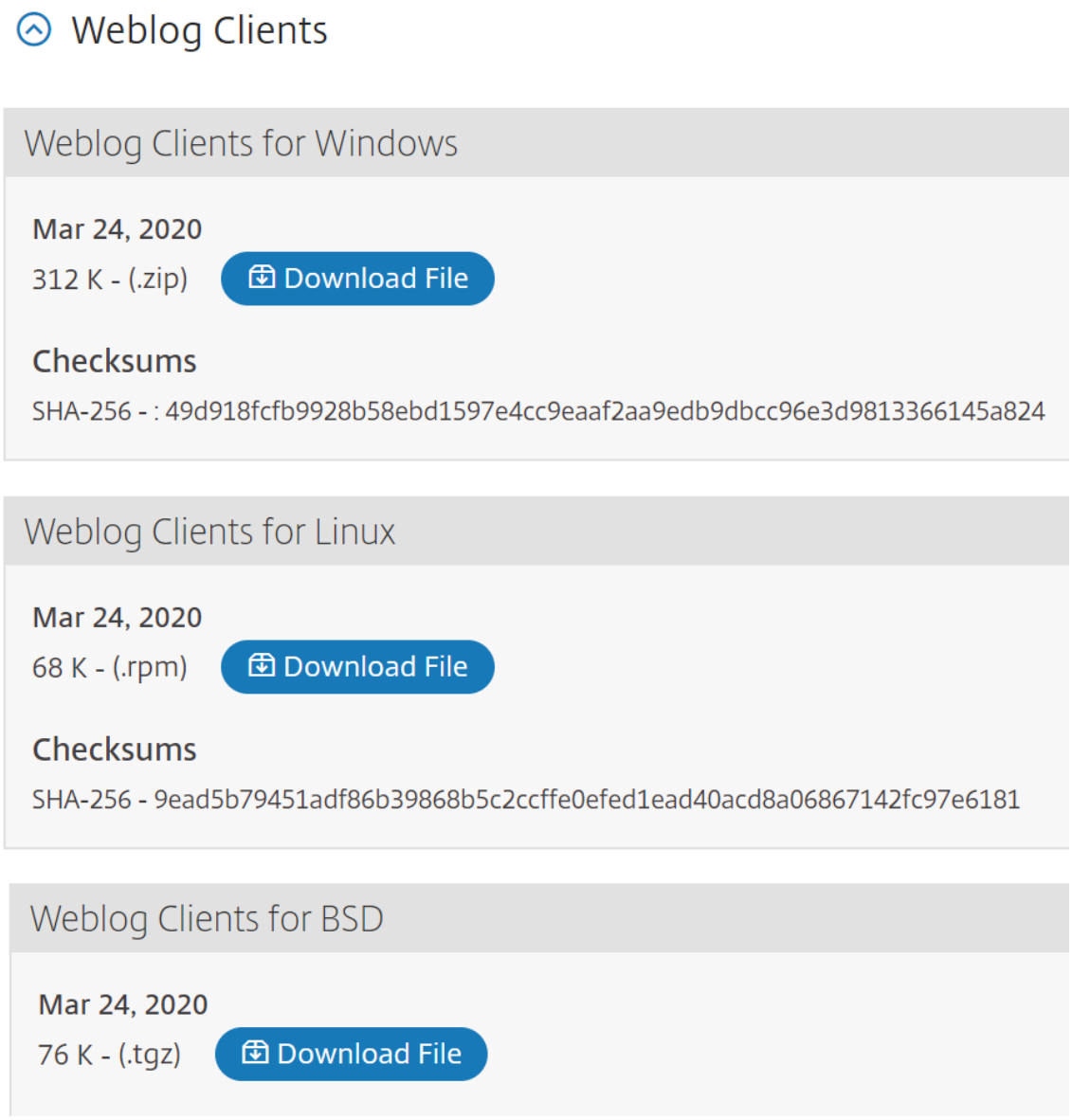
To download the NSWL client from the Citrix website

1. Log on to Citrix by accessing the URL `https://www.citrix.com/downloads/citrix-adc/`.

2. Navigate to a particular Citrix ADC release version and look for its Firmware.
3. Click **Firmware** (for example, Citrix ADC Release (Feature Phase) 13.0 Build 52.24).



4. In the **Citrix ADC Release (Feature Phase) Build** page, go to **Weblog Clients** section.
5. The section allows you to download Weblog clients for Windows, Linux, and BSD.



Install NSWL client on Solaris

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_solaris-<release number>-<build number>.tar file` from the package.
2. Copy the extracted file to a Solaris system on which you want to install the NSWL client.
3. Extract the files from the tar file with the following command:

   ```
   tar xvf nswl_solaris-9.3-51.5.tar
   ```


A directory Weblog is created in the temporary directory, and the files are extracted to the Weblog directory.

- Install the package with the following command:

  ```
  pkgadd -d
  ```

- The list of available packages appears. In the following example, one Weblog package is shown:

  ```
  1 NSweblog Citrix ADC Weblogging (SunOS,sparc)7.0
  ```

You are prompted to select the packages. Select the package number of the Weblog to be installed.

After you select the package number and press **Enter**, the files are extracted and installed in the following directories:

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. To check whether the NSWL package is installed, run the following command:

   ```
   pkginfo | grep NSweblog
   ```

2. To uninstall the NSWL package, run the following command:

   ```
   pkgrm NSweblog
   ```

Install NSWL client on Linux

> **Important**
>
> The installation of an NSWL client on Linux replaces the configuration file. You must take a backup before installing it.

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_linux-<release number>-<build number>.rpm` file from the package.
2. Copy the extracted file to a system, running Linux OS, on which you want to install the NSWL client.
3. To install the NSWL package, run the following command:

   ```
   rpm -i nswl_linux-9.3-51.5.rpm
   ```

This command extracts the files and installs them in the following directories.

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. To uninstall the NSWL package, run the following command:

   ```
   rpm -e NSweblog
   ```

2. To get more information about the Weblog RPM file, run the following command:

   ```
   rpm -qpi *.rpm
   ```

3. To view the installed Web server logging files, run the following command:

   ```
   rpm -qpl *.rpm
   ```

Install NSWL client on FreeBSD

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_bsd-<release number>-<build number>.tgz` file from the package.
2. Copy the extracted file to a system, running FreeBSD OS, on which you want to install the NSWL client.
3. To install the NSWL package, run the following command:

   ```
   pkg_add nswl_bsd-9.3-51.5.tgz
   ```

This command extracts the files and installs them in the following directories.

```
1 -  /usr/local/netscaler/etc
2 -  /usr/local/netscaler/bin
3 -  /usr/local/netscaler/samples
```

1. To uninstall the NSWL package, run the following command:

   ```
   pkg_delete NSweblog
   ```

2. To verify that the package is installed, run the following command:

   ```
   pkg_info | grep NSweblog
   ```

Install the NSWL client on Mac

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_macos-<release number>-<build number>.tgz` file from the package.
2. Copy the extracted file to a system, running macOS, on which you want to install the NSWL client.

3. To install the NSWL package, run the following command:

   ```
   pkg_add nswl_macos-9.3-51.5.tgz
   ```

This command extracts the files and installs them in the following directories:

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. To uninstall the NSWL package, run the following command:

   ```
   pkg_delete NSweblog
   ```

2. To verify that the package is installed, run the following command:

   ```
   pkg_info | grep NSweblog
   ```

Install NSWL client on Windows

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_win-<release number>-<build number>.zip file from the package.
2. Copy the extracted file to a Windows system on which you want to install the NSWL client.
3. On the Windows system, unzip the file in a directory (referred as <NSWL-HOME>). The following directories are extracted: /bin, and /etc and /samples.
4. At the command prompt, run the following command from the <NSWL-HOME>\bin directory:

   ```
   nswl -install -f <directorypath>\log.conf
   ```

Where,

Directory path refers to the path of the configuration file (log.conf). By default, the file is in the <NSWL-HOME> and /etc directory. You can copy the configuration file to any other directory.

> **Note**
>
> To uninstall the NSWL client, at the command prompt, run the following command from the <NSWL-HOME>\bin directory:

```
1 > nswl -remove
```


Install NSWL client on AIX system

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_aix-<release number>-<build number>.rpm` file from the package.
2. Copy the extracted file to a system, running AIX OS, on which you want to install the NSWL client.
3. To install the NSWL package, run the following command:

   ```
   rpm -i nswl_aix-9.3-51.5.rpm
   ```

This command extracts the files and installs them in the following directories.

- `/usr/local/netscaler/etc`
- `/usr/local/netscaler/`
- `usr/local/netscaler/samples`

1. To uninstall the NSWL package, run the following command:

   ```
   rpm -e NSweblog
   ```

2. To get more information about the Weblog RPM file, run the following command:

   ```
   rpm -qpi *.rpm
   ```

3. To view the installed Web server logging files, run the following command:

   ```
   rpm -qpl *.rpm
   ```

Configure NSWL client

September 14, 2021

After you install the NSWL client, you can configure the NSWL client using the nswl executable. These configurations are stored in the NSWL client configuration file (log.conf).

> **Note:**
>
> You can further customize logging on the NSWL client by making additional modifications to the NSWL configuration file (log.conf). For details, see Customizing Logging on the NSWL Client System.

The following table describes the commands that you can use to configure the NSWL client.

NSWL command	Specifies
nswl -help	The available NSWL help options.
nswl -addns -f <path-to-configuration-file>	The system that gathers the log transaction data. You are prompted to enter the IP address of the Citrix ADC appliance. Enter a valid user name and password.
nswl -verify -f <path-to-configuration-file>	Check for syntax or semantic errors in the configuration file.
nswl -start -f <path-to-configuration-file>	Start the NSWL client based on the settings in the configuration file. Note: For Solaris and Linux: To start Web server logging as a background process, type the ampersand sign (&) at the end of the command.
nswl -stop (Solaris and Linux only)	Stop the NSWL client if it was started as a background process; otherwise, use CTRL+C to stop Web server logging.
nswl -install -f <path-to-configuration-file> (Windows only)	Install the NSWL client as a service in Windows.
nswl -startservice (Windows only)	Start the NSWL client by using the settings in the configuration file specified in the nswl install option. You can also start NSWL client from Start > Control Panel > Services. Note: The NSWL log files will be created in C:\Windows\SysWOW64.
nswl -stopservice (Windows only)	Stops the NSWL client.
nswl -remove	Remove the NSWL client service from the registry.

Run the following commands from the directory in which the NSWL executable is located:

- Windows: \ns\bin
- Solaris and Linux: \usr\local\netscaler\bin

The Web server logging configuration files are located in the following directory path:

- Windows: \ns\etc
- Solaris and Linux: \usr\local\netscaler\etc

The NSWL executable is started as .\nswl in Linux and Solaris.

Add the IP addresses of the Citrix ADC appliance

In the NSWL client configuration file (log.conf), add the Citrix ADC IP address (NSIP) from which the NSWL client will start collecting logs.

To add the NSIP address of the Citrix ADC appliance

1. At the client system command prompt, type:

   ```
   nswl -addns -f < directorypath > \log.conf
   < directorypath >: Specifies the path to the configuration file (log.
   conf).
   ```

2. At the next prompt, enter the following information:

   - **NSIP**: Specify the IP address of the Citrix ADC appliance.
   - User name and Password: Specify the nsroot user credentials of the Citrix ADC appliance.

> **Note**:
>
> If you add multiple Citrix ADC IP addresses (NSIP), and later you do not want to log all of Citrix ADC system log details, you can delete the NSIPs manually by removing the NSIP statement at the end of the log.conf file. During a failover setup, you must add both primary and secondary Citrix ADC IP addresses to the log.conf by using the command. Before adding the IP address, make sure the user name and password exist on the Citrix ADC appliances.

Verify the NSWL configuration file

To make sure that logging works correctly, check the NSWL configuration file (log.conf) on the client system for syntax errors.

To verify the configuration in the NSWL configuration file

At the client system command prompt, type:

```
nswl -verify -f <directorypath>\log.conf
```

< directorypath>: Specifies the path to the configuration file (log.conf).

Execute NSWL Client

Start Web server logging

At the client system command prompt, type:

```
nswl -start -f <directorypath>\log.conf
```


<directorypath>: Specifies the path to the configuration file (log.conf).

Stop Web server logging started as a background process on the Solaris or Linux operating systems

At the command prompt, type:

```
nswl -stop
```

To stop Web server logging started as a service on the Windows operating system

At the command prompt, type:

```
nswl -stopservice
```

Customize logging on the NSWL client system

September 14, 2021

You can customize logging on the Citrix ADC Web Logging (NSWL) client system by making more modifications to the NSWL client configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the client system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information based on the host IP address, domain name, and host name of the Web servers.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

Sample configuration file

Following is a sample configuration file:

```
1  ##########
2  # This is the NSWL configuration file
3  # Only the default filter is active
4  # Remove leading # to activate other filters
5  ##########
6  ##########
7  # Default filter (default on)
8  # W3C Format logging, new file is created every hour or on reaching 10
       MB file size,
9  # and the file name is Exyymmdd.log
10 ##########
11 Filter default
12 begin default
```



```
13          logFormat               W3C
14          logInterval             Hourly
15          logFileSizeLimit        10
16          logFilenameFormat       Ex%`{
17   `%y%m%d }
18   t.log
19  end default
20  ##########
21  # Citrix ADC caches example
22  # CACHE_F filter covers all the transaction with HOST name www.
       netscaler.com and the listed server ip's
23  ##########
24  #Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95
        192.168.100.52 192.168.100.53 ON
25  ##########
26  # netscaler origin server example
27  # Not interested in Origin server to Cache traffic transaction logging
28  ##########
29  #Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66
       192.168.100.67 192.168.100.225 192.168.100.226 192.168.
30  100.227 192.168.100.228 OFF
31  ##########
32  # netscaler image server example
33  # all the image server logging.
34  ##########
35  #Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71
       192.168.100.72 192.168.100.169 192.168.100.170 192.168.10
36  0.171 ON
37  ##########
38  # NCSA Format logging, new file is created every day midnight or on
       reaching 20MB file size,
39  # and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.
       log.
40  # Exclude objects that ends with .png .jpg .jar.
41  ##########
42  #begin ORIGIN_SERVERS
43  #       logFormat               NCSA
44  #       logInterval             Daily
45  #       logFileSizeLimit        40
46  #       logFilenameFormat       /datadisk5/ORGIN/log/%v/NS%`{
47   `%m%d%y }
48   t.log
49  #       logExclude              .png .jpg .jar
50  #end ORIGIN_SERVERS
51
```



```
##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log with log record timestamp as GMT.
##########
#begin CACHE_F
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logtime                 GMT
#end CACHE_F

##########
# W3C Format logging, new file on reaching 20MB and the log file path name is
# atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record timestamp as LOCAL.
##########
#begin IMAGE_SERVER
#       logFormat               W3C
#       logInterval             Size
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk6/netscaler/log/%AEx%`{
 `%m%d%y }
 t
#       logtime                 LOCAL
#end IMAGE_SERVER

##########
# Virtual Host by Name firm, can filter out the logging based on the host name by,
##########

#Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
#begin VHOST_F
#       logFormat               W3C
#       logInterval             Daily
#       logFileSizeLimit        10
logFilenameFormat /ns/prod/vhost/%v/Ex%`{
 `%m%d%y }
 t
```

```
92 #end VHOST_F
93
94 ########## END FILTER CONFIGURATION ##########
95 <!--NeedCopy-->
```

Creating filters

You can use the default filter definition in the configuration file (log.conf), or you can modify the filter or create a filter. You can create more than one log filter.

> **Note**
>
> Consolidated logging, which logs transactions for which no filter is defined, uses the default filter if it is enabled. Consolidated logging of all servers can be done by defining only the default filter.

If the server hosts multiple websites and each website has its own domain name, and each domain is associated with a virtual server, you can configure Web server logging to create a separate log directory for each website. The following table displays the parameters for creating a filter.

Parameter	Specifies
filterName	Name of the filter. The filter name can include alphanumeric characters and cannot be longer than 59 characters. Filter names longer than 59 characters are truncated to 59 characters.
Host name	Host name of the server for which the transactions are being logged.
IP `ip`	IP address of the server for which transactions are to be logged (for example, if the server has multiple domains that have one IP address).
IP `ip 2...ip` n:	Multiple IP addresses (for example, if the server domain has multiple IP addresses).
ip6 IP	IPv6 address of the server for which transactions are to be logged.
IP ip NETMASK mask	IP addresses and netmask combination to be used on a subnet.
`ON | OFF`	Enable or disable the filter to log transactions. If no argument is selected, the filter is enabled (ON).

Table 1. Parameters for Creating a Filter

To create a filter

To create a filter, enter the following command in the log.conf file:

- `filter <filterName> <HOST name> | [IP<ip> ] | [IP<ip 2...ip n> ] | <IP ip NETMASK mask> [ON | OFF]`
- `filter <filterName> <HOST name> | [IP6 ip/<prefix length>] [ON | OFF]`

To create a filter for a virtual server

To create a filter for a virtual server, enter the following command in the log.conf file:

```
filter <filterName> <VirtualServer IP address>
```

Example

In the following example, you specify an IP address of 192.168.100.0 and netmask of 255.255.255.0. The filter applies to IP addresses 192.168.100.1 through 192.168.100.254.

```
1  Filter F1 HOST www.netscaler.com ON
2  Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
3  Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
4  Filter F4 IP 192.168.100.151
5  Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
6  Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com
      IP 192.168.100.200 ON
7  Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
8  Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
9  For creating filters for servers having IPv6 addresses.
10 Filter F9 2002::8/112 ON
11 Filter F10 HOST www.abcd.com IP6 2002::8 ON
12
13 <!--NeedCopy-->
```

Specifying log properties

Log properties are applied to all log entries associated with the filter. The log property definition begins with the keyword BEGIN and ends with END as illustrated in the following example:

```
1 BEGIN <filtername>
2  logFormat ...
3  logFilenameFormat ...
```



```
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
<!--NeedCopy-->
```

Entries in the definition can include the following:

- **LogFormat** specifies the Web server logging feature that supports NCSA, W3C Extended, and custom log file formats.

  By default, the `logformat` property is w3c. To override, enter custom or NCSA in the configuration file, for example:

```
     LogFormat NCSA
<!--NeedCopy-->
```

> **Note**
>
> For the NCSA and custom log formats, local time is used to time stamp transactions and for file rotation.

- **LogInterval** specifies the intervals at which new log files are created. Use one of the following values:
  - Hourly: A file is created every hour.
  - Daily: A file is created every day at midnight. Default value.
  - Weekly: A file is created every Sunday at midnight.
  - Monthly: A file is created on the first day of the month at midnight.
  - None: A file is created only once, when Web server logging starts.

Example:

```
LogInterval Daily
<!--NeedCopy-->
```

LogFileSizeLimit specifies the maximum size of the log file in MB. It can be used with any log interval (weekly, monthly, and so on.) A file is created when the maximum file size limit is reached or when the defined log interval time elapses.

To override this behavior, specify the size as the `loginterval` property so that a file is created only when the log file size limit is reached.

The default LogFileSizeLimit is 10 MB.

Example:


```
LogFileSizeLimit 35
<!--NeedCopy-->
```

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:
  - Static: Specifies a constant string that contains the absolute path and file name.

    Dynamic: Specifies an expression containing the following format:
    - Server IP address
    - Date (%{format}t)
    - URL suffix (%x)
    - Host name (%v)

Example:

```
LogFileNameFormat Ex%`{
 `%m%d%y }
 t.log
<!--NeedCopy-->
```

This command creates the first file name as Exmmddyy.log, then every hour creates a file with a file name: Exmmddyy.log.0, Exmmddyy.log.1,…, Exmmddyy.log.n.

Example:

```
      LogInterval size
      LogFileSize 100
      LogFileNameFormat Ex%`{
 `%m%d%y }
 t
<!--NeedCopy-->
```

> **Caution**
>
> The date format %t specified in the LogFilenameFormat command overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat.

- **LogExclude** prevents logging of transactions with the specified file name extensions.

Example:

```
      LogExclude .html
<!--NeedCopy-->
```


This command creates a log file that excludes log transactions for *.html files.

LogTime specifies log time as either GMT or LOCAL.

The defaults are:

- NCSA log file format: LOCAL
- W3C log file format: GMT.

Understanding NCSA and W3C log formats

The Citrix ADC supports the following standard log file formats:

- NCSA Common Log Format
- W3C Extended Log Format

NCSA Common Log Format

If the log file format is NCSA, the log file displays log information in the following format:

```
1 Client_IP_address -User_Name [Date:Time -TimeZone] "Method Object
     HTTP_version" HTTP_StatusCode BytesSent
2 <!--NeedCopy-->
```

To use the NCSA Common log format, enter NCSA in the LogFormat argument in the log.conf file.

The following table describes the NCSA Common log format.

Argument	Specifies
Client_IP_address	The IP address of the client computer.
User Name	The user name.
Date	The date of the transaction.
Time	The time when the transaction was completed.
Time Zone	The time zone (Greenwich Mean Time or local time).
Method	The request method (for example; GET, POST).
Object	The URL.
HTTP_version	The version of HTTP used by the client.
HTTP_StatusCode	The status code in the response.
Bytes Sent	The number of bytes sent from the server.

W3C extended log format

An extended log file contains a sequence of lines containing ASCII characters terminated by either a Line Feed (LF) or the sequence Carriage Return Line Feed (CRLF.) Log file generators must follow the line termination convention for the platform on which they are run.

Log analyzers must accept either LF or CRLF form. Each line might contain either a directive or an entry. If you want to use the W3C Extended log format, enter W3C as the Log-Format argument in the log.conf file.

By default, the standard W3C log format is defined internally as the custom log format, shown as follows:

```
%`{
 `%Y-%m-%d%H:%M:%S }
 t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{
 user-agent }
 i %+{
 cookie }
  i%+{
 referer }
 i
<!--NeedCopy-->
```

You can also change the order or remove some fields in this W3C log format. For example:

```
logFormat W3C %`{
 `%Y-%m-%d%H:%M:%S }
 t %m %U
<!--NeedCopy-->
```

W3C log entries are created with the following format:

```
#Version: 1.0 #Fields: date time cs-method cs-uri #Date: 12-Jun-2001
   12:34 2001-06-12 12:34:23 GET /sports/football.html 2001-06-12
   12:34:30 GET /sports/football.html
<!--NeedCopy-->
```

Entries

Entries consist of a sequence of fields relating to a single HTTP transaction. Fields are separated by white space. Citrix recommends the use of tab characters. If a field in a particular entry is not used, a dash (-) marks the omitted field.

Directives

See the Directives table for the information about the logging process. Lines beginning with the pound sign (#) contain directives.

Example:

The following sample log file shows the log entries in W3C Extended log format:

```
1 #Version: 1.0 #Fields: time cs-method cs-uri #Date: 12-Jan-1996
     00:00:00 00:34:23 GET /sports/football.html 12:21:16 GET /sports/
     football.html 12:45:52 GET /sports/football.html 12:57:34 GET /
     sports/football.html
2 <!--NeedCopy-->
```

Fields

The Fields directive lists a sequence of field identifiers that specify the information recorded in each entry. Field identifiers might have one of the following forms:

- **identifier:** Relates to the transaction as a whole.
- **prefix-identifier:** Relates to information transfer between parties defined by the value prefix.
- **prefix (header):** Specifies the value of the HTTP header field header for transfer between parties defined by the value prefix. Fields specified in this manner always have the type.

The following table describes defined prefixes.

Prefix	Specifies
c	Client
s	Server
r	Remote
cs	Client to server
sc	Server to client
sr	Server to remote server (prefix used by proxies)
rs	Remote server to server (prefix used by proxies)
x	Application-specific identifier

Examples:

The following examples are defined identifiers that use prefixes:

cs-method: The method in the request sent by the client to the server.

sc(Referer): The `Referer` field in the reply.

c-ip: The IP address of the client.

Identifiers

The following table describes the W3C Extended log format identifiers that do not require a prefix.

Identifier	Description
date	The date on which the transaction was done.
time	The time when the transaction is done.
time-taken	The time taken (in seconds) for the transaction to complete.
bytes	The number of bytes transferred.
cached	Records whether a cache hit has occurred. A zero indicates a cache miss.

Table 5. W3C Extended Log Format Identifiers (No Prefix Required)

The following table describes the W3C Extended log format identifiers that require a prefix.

Identifier	Description
IP	The IP address and the port number.
DNS	The DNS name.
status	The status code.
comment	The comment returned with a status code.
method	The method.
url	The URL.
url-stem	The stem portion of the URL.
url-query	The query portion of the URL.

Table 6. W3C Extended Log Format Identifiers (Requires a Prefix)

The W3C Extended Log file format allows you to choose log fields. These fields are shown in the following table.

Field	Description
Date	The date on which the transaction is done.
Time	The time when the transaction is done.
Client IP	The IP address of the client.
User Name	The user name.
Service Name	The service name, which is always HTTP.
Server IP	The server IP address.
Server Port	The server port number
Method	The request method (for example; GET, POST).
Url Stem	The URL stem.
Url Query	The query portion of the URL.
HTTP Status	The status code in the response.
Bytes Sent	The number of bytes sent to the server (request size, including HTTP headers).
Bytes Received	The number of bytes received from the server (response size, including HTTP headers).
Time Taken	The time taken for a transaction to complete, in seconds.
Protocol Version	The version number of HTTP being used by the client.
User Agent	The User-Agent field in the HTTP protocol.
Cookie	The Cookie field of the HTTP protocol.
`Referer`	The `Referer` field of the HTTP protocol.

Table 7. W3C Extended Log File Format (Allows Log Fields)

Creating a custom log format

You can customize the display format of the log file data manually or by using the NSWL library. By using the custom log format, you can derive most of the log formats that Apache currently supports.

Creating a custom log format by using the NSWL library

Use one of the following NSWL libraries depending on whether the NSWL executable has been installed on a Windows or Solaris host computer:

- **Windows:** The nswl.lib library in the \ns\bin directory on the system manager host computer.
- **Solaris:** The libnswl.a library in /usr/local/netscaler/bin.

To create the custom log format by using the NSWL library

1. Add the following two C functions defined by the system in a C source file:

   ns_userDefFieldName(): This function returns the string that must be added as a custom field name in the log record.

   ns_userDefFieldVal(): This function implements the custom field value, then returns it as a string that must be added at the end of the log record.

2. Compile the file into an object file.
3. Link the object file with the NSWL library (and optionally, with third party libraries) to form a new NSWL executable.
4. Add a %d string at the end of the logFormat string in the configuration file (log.conf).

Example:

```
########## # A new file is created every midnight or on reaching 20MB
   file size, # and the file name is /datadisk5/netscaler/log/NS<
   hostname>/Nsmmddyy.log and create digital #signature field for each
   record. BEGIN CACHE_F logFormat custom "%a - "%{
 user-agent }
 i" [%d/%B/%Y %T -%g] "%x" %s %b%{
 referrer }
 i "%{
 user-agent }
 i" "%{
 cookie }
 i" %d " logInterval Daily logFileSizeLimit 20 logFilenameFormat /
    datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log END CACHE_F
<!--NeedCopy-->
```


Creating a Custom log format manually

To customize the format in which log file data must appear, specify a character string as the argument of the LogFormat log property definition. The following is an example where character strings are used to create a log format:

```
LogFormat Custom ""%a - "%{
 user-agent }
 i" %[%d/%m/%Y]t %U %s %b %T"
<!--NeedCopy-->
```

- The string can contain the “c” type control characters \n and \t to represent new lines and tabs.
- Use the Esc key with literal quotes and backslashes.

The characteristics of the request are logged by placing % directives in the format string, which are replaced in the log file by the values.

If the %v (Host name) or %x (URL suffix) format specifier is present in a log file name format string, the following characters in the file name are replaced by an underscore symbol in the log configuration file name:

```
” *. /: < > ? \
```

Characters whose ASCII values lie in the range of 0-31 are replaced by the following:

%<ASCII value of character in hexadecimal>.

For example, the character with ASCII value 22 is replaced by %16.

> **Caution**
>
> If the %v format specifier is present in a log file name format string, a separate file is opened for each virtual host. To ensure continuous logging, the maximum number of files that a process can have open must be sufficiently large. See your operating system documentation for a procedure to change the number of files that can be opened.

Creating Apache log formats

You can derive from the custom logs most of the log formats that Apache currently supports. The custom log formats that match Apache log formats are:

NCSA/combined: LogFormat custom %h %l %u [%t] “%r” %s %B “%{referer}i” “%{user-agent}i”

NCSA/Common: LogFormat custom %h %l %u [%t] “%r” %s %B

Referer Log: LogFormat custom “%{referer}i” -> %U

User agent: LogFormat custom %{user-agent}i

Similarly, you can derive the other server log formats from the custom formats.

Arguments for defining a custom log format

See the Custom log format PDF table for more information about defining a custom log format.

> **Note**
>
> For instructions on how to export custom HTTP headers, see Configuring the Citrix ADC for Web Server Logging

For example, if you define the log format as %+{user-agent}i, and if the user agent value is Citrix ADC system Web Client, then the information is logged as Citrix ADC system+Web+Client. An alternative is to use double quotation marks. For example, "%{user-agent}i" logs it as "Citrix ADC system Web Client." Do not use the <Esc> key on strings from %.. .r, %. . .i and, %. . .o. It complies with the requirements of the Common Log Format. Clients can insert control characters into the log. Therefore, you must take care when working with raw log files.

Time format definition

See the Time format definition table to know about the format part of the %{format}t string described in the Custom Log Format table. Values within brackets ([ ]) show the range of values that appear. For example, [1,31] in the %d description in the following table shows %d ranges from 1 to 31.

> **Note**
>
> If you specify a conversion that does not correspond to any of the ones described in the preceding table, or to any of the modified conversion specifications listed in the next paragraph, the behavior is undefined and returns 0.

The difference between %U and %W (and also between modified conversions %OU and %OW) is the day considered to be the first day of the week. Week number 1 is the first week in January (starting with a Sunday for %U, or a Monday for %W). Week number 0 contains the days before the first Sunday or Monday in January for %U and %W.

Displaying server logs

You can configure an NSWL feature to display server logs on the console or redirect server logs to a directory on the Citrix ADC appliance.

There are two ways to display logs on the console (standard output):
Option 1: Display all logs on the console.
Option 2: Display only selected logs on the console with filters with `logfilenameformat` as STDOUT.

Call Home

September 14, 2021

Appliances might sometimes fail to perform well because of software or hardware issues. In such cases, Citrix needs to collect data and perform issue resolution before a potential impact can occur at the customer site. By enabling Call Home on your Citrix ADC appliance, you can automate the error notification process. Not only you avoid calling Citrix support, raising a service request, and uploading system data before the support team can troubleshoot the issue, but support can identify and address an issue before it occurs. Call Home periodically monitors the appliance and automatically uploads data to the Citrix technical support server. In addition, the incoming Call Home data provide insights about Citrix ADC usage. Multiple teams within Citrix can use this data to better design, support, and implement Citrix ADC.

By default, Call Home is enabled on all platforms and all flavors of Citrix ADC (MPX, VPX, SDX). By having this feature enabled, you allow Citrix to collect Citrix ADC deployment and telemetry data for better implementation, and support service.

> **Note**
>
> You can also see the Call Home FAQ page for information pertaining to Call Home.

Benefits

Call Home provides the following benefits.

- Monitor hardware and software error conditions. For more information, see Monitor critical error conditions section.
- Notify critical events that impact your network.
- Send performance data and system usage details to Citrix to:
  - Analyze and improve product quality.
  - Provide real-time troubleshooting information for proactive issue identification, and faster issue resolution.

Platform support

Call Home feature is supported on all Citrix ADC platforms and all appliance models (MPX, VPX, and SDX).

- Citrix ADC MPX: All MPX models.
- Citrix ADC VPX: All VPX models, including VPX appliances that obtain their license from external or central licensing pools.

- Citrix ADC SDX: Monitors the disk drive and assigned SSL chips for any errors or failures. The VPX instances, however, do not have access to the Power Supply Unit (PSU) and therefore their status is not monitored. In an SDX platform, you can configure Call Home either directly on an individual instance or through the SVM.

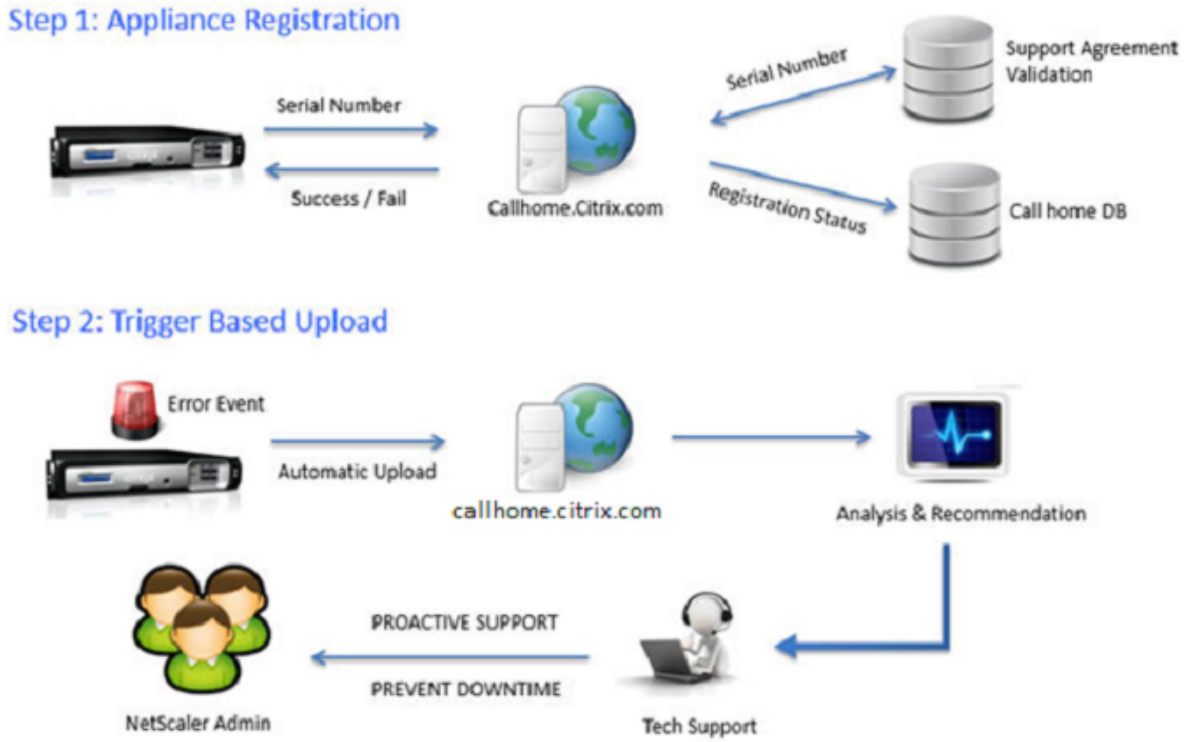
Prerequisites

To use Call Home, the Citrix ADC appliance must have the following:

- **Internet connection.** Call Home requires an internet connection for the Citrix ADC to connect to the Citrix support server for uploading a data archive.

How Call Home works

The following figure shows a basic workflow of Call Home in a Citrix ADC appliance deployed at a customer site.



The following are the workflow of a Call Home:

1. Set up Internet Connectivity. For Call Home to upload system data, your appliance must have an internet connectivity. If it does not, you can configure a proxy server configuration to provide internet connectivity. For more information, see Configuring Call Home section.

2. Enable Call Home. When upgrading your appliance to the latest software through either Citrix ADC command interface or GUI, Call Home is enabled by default and the system delays the registration process by 24 hours. During this period, you can choose to manually disable the feature, but Citrix recommends you to have it enabled.

> **Note**
>
> If you are upgrading your appliance from an older version that explicitly has Call Home disabled, the system still enables the feature by default and displays a notification message on your first login.

In addition, if you are doing any configuration changes for an internet connectivity, you must disable and enable Call Home. It enables Call Home to register with the Citrix Insight Services (CIS) server without any failure errors.

3. Register the Citrix ADC appliance on the Citrix Support Server. When Call Home registers the appliance with the Citrix Support server, the server checks the database for the validity of the appliance serial number. If the serial number is valid, the server registers the appliance for Call Home service and sends a successful registration response. Otherwise, the server sends back a registration failure message. The basic system information is sent as a separate message. The data includes memory and CPU usage details along with the throughput numbers. The data is sent periodically as part of the heartbeat message every 7 days, by default. However, a value of less than 5 days is not recommended, because frequent uploads are not useful.

4. Monitor critical error conditions. Once registered, Call Home starts monitoring the appliance. The following table lists the conditions Call Home can monitor on the appliance.

Critical Error Condition	Description	Call Home Monitoring Interval	Corresponding SNMP Alarm Name
Compact flash drive errors	The compact flash drive on the appliance encountered read or write failures.	24 hours	COMPACT-FLASH-ERRORS
Hard disk drive errors	The hard drives on the appliance encountered read or write failures.	24 hours	HARD-DISK-DRIVE-ERRORS
Power supply unit failure	One of the power supply units on the Citrix ADC appliance has failed.	7 seconds	POWER-SUPPLY-FAILURE

Critical Error Condition	Description	Call Home Monitoring Interval	Corresponding SNMP Alarm Name
SSL card failure	One of the SSL cards on the Citrix ADC appliance has failed.	7 seconds	SSL-CARD-FAILED
Warm restart	The appliance has warm restarted due to a failure of a system process.	After every restart of the Citrix ADC appliance.	WARM-RESTART-EVENT
Memory anomaly error	Memory utilization progressively increases above its normal limit and exceeds the threshold.	1 day	No SNMP alarm
Rate limit packet drop	The throughput limits or packets-per-second (pps) limits is reached.	7 seconds	PF-RL-PPS-PKTS-DROPPED, PF-RL-RATE-PKTS-DROPPED

5. Upload Call Home data. If any one of the previous critical conditions is identified on the appliance, the Call Home feature automatically notifies the Citrix support. The support archives are uploaded to the Citrix support server. Also, you can configure the CALLHOME-UPLOAD-EVENT SNMP alarm to generate an SNMP alert whenever Call Home upload happens. The SNMP alert notifies the local administrator about the critical event.

> **Note**
>
> Call Home creates the Call Home tar file and uploads it to the Citrix tech support server for only the first occurrence of a particular error condition since the last reboot. If you want the appliance to send alerts each time a particular error condition occurs, configure the corresponding SNMP alarm for the error condition.

6. Create Service Request. Call Home automatically creates a service request for all the critical hardware related events. The events are classified as; power supply failure, SSL card failure, hard disk drive errors, and compact flash errors. For other errors, after you review the System Logs, you can contact the Citrix support team to raise a service request for investigation.

Configuring Call Home

To configure Call Home, verify the internet connectivity on the appliance and ensure a DNS name-server is configured. If there is no internet connection, configure a proxy server or service. Then, enable Call Home on the appliance and verify the registration status of the appliance with the Citrix support server. Once registered, Call Home can monitor and upload data. In addition, you can configure SNMP alarms to notify the administrator at the customer site.

To configure Call Home, you can use either the Citrix ADC command interface or the GUI to do the following tasks:

- Enable Call Home.
- Configure Call Home for optional proxy server parameters.
- Verify Call Home Registration Status.
- View errors and timestamp details.
- Configure SNMP Alarms.

To configure Call Home by using the Citrix ADC command interface

The Citrix ADC command interface enables you to do the following:

`Enabling Call Home`

At the command prompt, type:

```
enable ns feature callhome
```

Configuring Call Home for optional proxy server parameters

Call Home enables you to configure the optional proxy server for internet connectivity. You can either configure a proxy server with IP address and port or configure a proxy authentication service with one-way or two-way authentication.

`To configure optional proxy server with IP address and port`

At the command prompt, type:

```
set callhome -proxyMode ( YES | NO )[-IPAddress <ip_addr|ipv6_addr|*>] [-port <port |*>]
```

```
1 set callhome -proxyMode YES -IPAddress 10.102.167.33 -port 80
2 <!--NeedCopy-->
```

> **Note**
>
> Call Home uses the proxy server only when you set the proxy-mode parameter to YES. If you set

> it to NO, the proxy functionality does not work, even if the IP address and port are configured. The port number must be for an HTTP service, not for an HTTPS service.

To configure optional proxy authentication service

This mode provides two types of security authentication: one-way and two-way. To set up either type, you must configure an SSL service. For more information, see Configuring an SSL Service topic.

In one-way authentication, only the Citrix ADC appliance authenticates the proxy server. In two-way authentication, the Citrix ADC appliance authenticates the proxy server and the proxy server, in turn, authenticates the appliance.

To configure proxy authentication service

At the command prompt, type:

```
set callhome -proxyMode ( YES | NO )[-proxyAuthService <string>]
```

```
1 set callhome  -proxyMode YES  -proxyAuthService callhome_proxy
2 <!--NeedCopy-->
```

To configure one-way proxy server authentication

Do the following tasks to configure one-way proxy server authentication.

1. Create an SSL service.
2. Bind a CA certificate to the service.
3. Bind an HTTPS Monitor to the service.
4. Configure Call Home to use the SSL service.

To configure two-way proxy server authentication

Do the following tasks to configure two-way proxy server authentication.

1. Create an SSL service
2. Bind a CA certificate to the service.
3. Bind a Client Certificate.
4. Bind an HTTPS Monitor to the service.
5. Configure Call Home to use the SSL service.

Verifying Call Home Registration status

At the command prompt, type:

```
1 show callhome
2
3     show callhome
4
5      Registration with Citrix upload server SUCCESSFUL
6
```



```
7       Mode: Default
8
9       Contact email address: exampleadmin@example.com
10
11      Heartbeat Custom Interval (days): 7
12
13      Proxy Mode: Yes
14
15         Proxy IP Address:10.102.29.200
16
17         Proxy Authentication Service:
18
19         Proxy Port: 80
20
21     Trigger event                   State   First occurrence
                      Latest occurrence
22
23      -------------                   -----   ----------------
                       -----------------
24
25      1) Warm boot                    Enabled N/A
                                        ..
26
27      2) Compact flash errors         Enabled ..
                                          ..
28
29      3) Hard disk drive errors       Enabled ..
                                          ..
30
31      4) SSL card failure             N/A     N/A
                                        N/A
32
33      5) Power supply unit failure    N/A     N/A
                                        N/A
34
35      6) Rate limit packet drops      Enabled ..
                                          ..
36
37      7) Memory anomaly               Enabled ..
                                          ..
38
39      Done
40 <!--NeedCopy-->
```


> **Note**
>
> If the Call Home fails to register with CIS, the appliance displays an error message.

Enabling SNMP Alarms

The Citrix ADC appliance provides a set of error condition entities called *SNMP alarms*. When an error condition in an SNMP alarm is met, the appliance generates SNMP trap messages that are sent to the configured trap listeners. For example, when the SSL-CARD-FAILED alarm is enabled, a trap message is generated and sent to the trap listener. The trap message is sent whenever there is an SSL card failure on the appliance. For more information, see SNMP.

At the command prompt, type:

```
enable snmp alarm <trapName>
```

```
show snmp alarm <trapName>
```

To configure Call Home by using GUI

To verify if the Call Home feature is enabled by default in the GUI

1. Navigate to **Configuration** > **System** > **Settings**.
2. In the **details** pane, click **Configure Advanced features** link.
3. In the **Configure Advanced Features** page, **Call Home** option must display as enabled.

To enable Call Home by using GUI

1. Navigate to **Configuration** > **System** > **Settings**.
2. In the **details** pane, click **Configure Advanced features** link and select **Callhome** option.

To configure Call Home for optional proxy mode authentication by using GUI

1. You can use any of the two ways to access the Call Home page:
   a) Navigate to **System** > **System Information**.
   b) Navigate to **System** > **Diagnostics**.
      i. In the details pane, under **Technical Support Tools**, select **Call Home**.
2. On the **Configure Call Home** page, set the following parameters.
   a) **Mode**. Call Home mode of operation. Possible types: Default, Citrix Service Provider (CSP) deployment.

      > **Note**
      >
      > This option is not user configurable. The mode is automatically determined and set based on the type of Citrix ADC deployment.

   b) **Email address**. Email address of the contact administrator at the customer site.
   c) **CallHome Heartbeats Interval (days).** Monitoring interval (in days) between Call Home heartbeats. Minimum value=1 and Maximum value=7.

d) **Enable Call Home**. Enable or disable the Call Home feature to view the status of appliance registration on the Citrix support server.
e) **Proxy Mode**. If you do not have an internet connectivity, enable proxy mode and set the optional proxy parameters.
f) **Proxy Server**. If you set the proxy mode by using a proxy server, specify the server IP address.
    i. **Proxy Service**. If you set the proxy mode by using a proxy service, specify the service name.
    ii. **IP Address**. IP address of the proxy server.
    iii. **Port**. Port number of the proxy server.
    iv. **Proxy Authentication SSL Service**. The name of the proxy service that provides proxy-mode authentication.

3. Click **OK** and **Done**.

To configure SSL service for proxy server authentication by using GUI

For information about configuring the SSL service by using the GUI, see Configuring an SSL Service topic.

To verify Call Home registration status by using the GUI

1. You can use any of the two ways to access the **Call Home** page:
    a) Navigate to **System** > **System Information**.
    b) Navigate to **System** > **Diagnostics**.
        i. In the details pane, under **Technical Support Tools**, select **Call Home**.
2. In the **Configure Call Home** page, the **Registration with Citrix upload server** field shows the registration status.

To configure an SNMP alarm

1. Navigate to **System** > **SNMP** > **Alarms**.
2. In the details pane, select an alarm and configure its parameters.
3. Click **OK** and **Close**.

Citrix Service Provider (CSP) deployment support

In a Citrix Service Provider (CSP) environment where Citrix ADC services are deployed on VPX instances, Call Home can monitor and track the license-specific information and securely send the information to Citrix Insight Services (CIS). CIS in turn sends the information to the License Usage Insights (LUI) portal for accounting purposes and for CSP customers to review their license usage. Currently, CSP environments support Citrix ADC services on VPX instances only, not on MPX or SDX appliances. The VPX instances can be deployed in either standalone or high availability mode.

Reporting tool

September 14, 2021

Use the Citrix® Citrix ADC® Reporting tool to view Citrix ADC performance statistics data as reports. Statistics data are collected by the `nscollect` utility and are stored in a database. When you want to view certain performance data over a period, the Reporting tool pulls out specified data from the database and displays them in charts.

Reports are a collection of charts. The Reporting tool provides built-in reports and the option to create custom reports. In a report, you can modify the charts and add new charts. You can also modify the operation of the data collection utility, `nscollect`, and stop or start its operation.

Using the reporting tool

The Reporting tool is a web-based interface accessed from the Citrix® Citrix ADC® appliance. Use the Reporting tool to display the performance statistics data as reports containing graphs. In addition to using the built-in reports, you can create custom reports, which you can modify at any time. Reports can have between one and four charts. You can create up to 256 custom reports. You can create a custom report for any number of entities.

To invoke the reporting tool

1. Use the Web browser of your choice to connect to the IP address of the Citrix ADC (for example, `http://10.102.29.170/`). The Web Logon screen appears.
2. In the User Name text box, type the user name assigned to the Citrix ADC.
3. In the Password text box, type the password.
4. In the Start in drop-down list box, select Reporting.Click Login.

The following screenshots show the report toolbar and the chart toolbar, which are frequently referenced in this documentation.

Figure 1. *Report Toolbar*

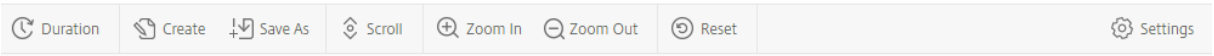
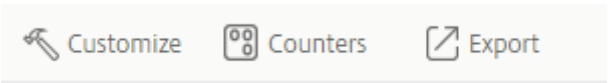


Figure 2. *Chart Toolbar*



Working with reports

You can plot and monitor statistics for the various functional groups configured on the Citrix ADC over a specified time interval. Reports enable you to troubleshoot or analyze the behavior of your appliance. There are two types of reports: built-in reports and custom reports. Report content for built-in or custom reports can be viewed in a graphical format or a tabular format. The graphical view consists of line, area, and bar charts that can display up to 32 sets of data (counters). The tabular view displays the data in columns and rows. This view is useful for debugging error counters.

The default report that is displayed in the Reporting tool is CPU vs. Memory Usage and HTTP Requests Rate. You can change the default report view by displaying the report you want as your default view, and then clicking Default Report.

Reports can be generated for the last hour, last day, last week, last month, last year, or you can customize the duration.

You can do the following with reports:

- Toggle between a tabular view of data and a graphical view of data.
- Change the graphical display type, such as bar chart or line chart.
- Customize charts in a report.
- Export the chart as an Excel comma-separated value (CSV) file.
- View the charts in detail by zooming in, zooming out, or using a drag operation (scrolling).
- Set a report as the default report for viewing whenever you log on.
- Add or remove counters.
- Print reports.
- Refresh reports to view the latest performance data.

Using built-in reports

The Reporting tool provides built-in reports for frequently viewed data. Built-in reports are available for the following functional groups: System, Network, SSL, Compression, Integrated Cache, Citrix ADC Gateway, and Citrix ADC Application Firewall. By default, the built-in reports are displayed for the last day. However, you can view the reports for the last hour, last week, last month, or last year.

> **Note**:
>
> You cannot save changes to built-in reports, but you can save a modified built-in report as a custom report.

To display a built-in report

1. In the left pane of the Reporting tool, under Built-in Reports, expand a group (for example, SSL).
2. Click a report (for example, **SSL > All Backend Ciphers)**.

Creating and deleting reports

You can create your own custom reports and save them with user-defined names for reuse. You can plot different counters for different groups based on your requirements. You can create up to 256 custom reports.

You can either create a new report or save a built-in report as a custom report. By default, a newly created custom report contains one chart named System Overview, which displays the CPU Usage counter plotted for the last day. You can customize the interval and set the data source and time zone from the report toolbar.

To create a custom report

1. In the **Reporting** tool, on the report toolbar, click **Create**, or if you want to create a new custom report based on an existing report, open the existing report, and then click **Save As**.
2. In **Report Name** box, type a name for the custom report.
3. Do one of the following:
   - To add the report to an existing folder, in Create in or Save in, click the down arrow to choose an existing folder, and then click **OK**.
   - To create a new folder to store the report, click the Click to add folder icon, in Folder Name, type the name of the folder, and in Create in, specify where you want the new folder to reside in the hierarchy, and then click **OK**.

> **Note**:
> You can create up to 128 folders.

To delete a custom report

1. In the left pane of the Reporting tool, next to Custom Reports, click the Click to manage the custom reports icon.
2. Select the check box that corresponds with the report you want to delete, and then click Delete.

   > **Note**:
   >
   > When you delete a folder, all the contents of that folder are deleted.

Modifying the time interval

By default, built-in reports display data for the last day. However, if you want to change the time interval for a built-in report, you can save the report as a custom report. The new interval applies to all charts in the report. The following table describes the time-interval options.

To modify the time interval

1. In the left pane of the Reporting tool, click a report.
2. On the report toolbar, click **Duration**, and then click a time interval.

Setting the data Source and time zone

You can retrieve data from different data sources to display them in the reports. You can also define the time zone for the reports and apply the currently displayed report's time selection to all the reports, including the built-in reports.

To set the data source and time zone

1. In the **Reporting tool**, on the report toolbar, click **Settings**.
2. In the **Settings** dialog box, in Data Source, select the data source from which you want to retrieve the counter information.
3. Do one or both of the following:
   - If you want the tool to remember the time period for which a chart is plotted, select the **Remember time selection for charts** check box.
   - If you want the reports to use the time settings of your Citrix ADC appliance, select the **Use Appliance's time zone** check box.

Exporting and importing custom reports

You can share reports with other Citrix ADC administrators by exporting reports. You can also import reports.

To export or import custom reports

1. In the left pane of the Reporting tool, next to Custom Reports, click the **Click to manage custom reports** icon.
2. Select the check box that corresponds with the report you want to export or import, and then click **Export** or **Import**.

   > **Note**:
   >
   > When you export the file, it is exported in a .gz file format.

Working with charts

Use charts to plot and monitor counters or groups of counters. You can include up to four charts in one report. In each chart, you can plot up to 32 counters. The charts can use different graphical formats

(for example, area and bar). You can move the charts up or down within the report, customize the colors and visual display for each counter in a chart, and delete a chart when you do not want to monitor it.

In all report charts, the horizontal axis represents time and the vertical axis represents the value of the counter.

Adding a chart

When you add a chart to a report, the System Overview chart appears with the CPU Usage counter plotted for the last one day.

> **Note**:
>
> If you add charts to a built-in report, and you want to retain the report, you must save the report as a custom report.

Use the following procedure to add a chart to a report.

To add a chart to a report

1. In the left pane of the Reporting tool, click a report.
2. Under the chart where you want to add the new chart, click the Add icon.

Modifying a chart

You can modify a chart by changing the functional group for which the statistics are displayed and by selecting different counters.

To modify a chart

1. In the left pane of the Reporting tool, click a report.
2. Under the chart that you want to modify, click Counters.
3. In the dialog box that appears, in the Title box, type a name for the chart.
4. Next to the Plot chart for, do one of the following:
   - To plot counters for global counters, such as Integrated Cache and Compression, click System global statistics.
   - To plot entity counters for entity types, such as Load Balancing and GSLB, click System entities statistics.
5. In the Select group, click the desired entity.
6. Under Counters, in Available, click one or more counter names that you want to plot, and then click the > button.

7. If you selected System entities statistics in step 4, on the Entities tab, under Available, click one or more entity instance names you want to plot, and then click the > button.
8. Click OK.

Viewing a chart

You can specify the graphical formats of the plotted counters in a chart. Charts can be viewed as line charts, spline charts, step-line charts, scatter charts, area charts, bar charts, stacked area charts, and stacked bar charts. You can also zoom in, zoom out, or scroll inside the plot area of a chart. You can zoom in or out for all data sources for 1 hour, 1 day, 1 week, 1 month, 1 year, and 3 years.

Other options for customizing the view of a chart include customizing the axes of the charts, changing the background and edge color of the plot area, customizing the color and size of the grids, and customizing the display of each data set (counter) in a chart.

Data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.

Whenever you modify a built-in report, you need to save the report as a custom report to retain your changes.

To change the graph type of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart you want to view, on the chart toolbar, click **Customize**.
3. On the **Chart** tab, under **Category**, click **Plot type**, and then click the graph type you want to display for the chart. If you want to display the graph is 3D, select the Use 3D check box.

To refocus a chart with detailed data

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Zoom In**, and do one or both of the following:
   - To refocus the chart to display data for a specific time window, drag the cursor from the start time to the end time. For example, you can view data for a one-hour period on a certain day.
   - To refocus the chart to display data for a data point, simply click once on the chart where you want to zoom in and get more detailed information.
3. Once you have the desired range of time for which you want to view detailed data, on the report toolbar, click Tabular View. Tabular view displays the data in numeric form in rows and columns.

To view numeric data for a graph

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click Tabular View. To return to the graphical view, click **Graphical View**.

> **Note**: You can also view the numeric data in the graphical view by hovering your cursor over the notches in the gridlines.

To scroll through time in a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Scroll**, and then click inside the chart and drag the cursor in the direction for which you want to see data for a new time period. For example, if you want to view data in the past, drag to the left.

To change the background color and text color of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click **Customize**.
3. On the **Chart** tab, under **Category**, click one or more of the following:
   - To change the background color, click **Background Color**, and then select the options for color, transparency, and effects.
   - To change the text color, click Text **Color**, and then select the options for color, transparency, and effects.

To customize the axes of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click **Customize**.
3. On the **Chart** tab, under Category, click one or more of the following:
   - To change the scale of the left y-axis, click **Left Y-Axis**, and then select the scale you want.
   - To change the scale of the right y-axis, click Right Y-Axis, in the data set to plot, select the date set, and then select the scale you want.

     > Note:
     > The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.

   - To plot each data set in its own hidden y-axis, click Multiple Axes, and then click Enable.

To change the background color, edge color, and gridlines for a plot area of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the plot area, click **Customize**.
3. On the **Plot Area** tab, under Category, click one or more of the following:
   - To change the background color and edge color of the chart, click **Background Color and Edge Color**, and then select the options for color, transparency, and effects.
   - To change the horizontal or vertical grids of the chart, click **Horizontal Grids** or **Vertical Grids**, and then select the options for displaying the grids, grid width, grid color, transparency, and effects.

To change the color and graph type of a data set

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the display of the data set (counters), click **Customize**.
3. On the **Data Set** tab, in Select Data Set, select the data set (counter) for which you want to customize the graphical display.
   Note: The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if
   CPU usage and
   Memory usage are displayed in first and second order at the bottom of the chart,
   CPU usage is equal to
   Data Set 1 and
   Memory usage is equal to
   Data Set 2.
4. Under Category, do one of more of the following:
   - To change the background color, click **Color**, and then select the options for color, transparency, and effects.
   - To change the graph type, click **Plot type**, and then select the graph type you want to display for the data set. If you want to display the graph as 3D, select the Use 3D check box.

Exporting chart data to excel

For further data analysis, you can export charts to Excel in a comma-separated value (CSV) format.

To export chart data to excel

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart with the data you want to export to Excel, click **Export**.

Deleting a chart

If you do not want to use a chart, you can remove it from the report. You can permanently remove charts from custom reports only. If you delete a chart from a built-in report and want to retain the changes, you need to save the report as a custom report.

To delete a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart that you want to delete, click the **Delete** icon.

Examples

To display the trend report for CPU usage and memory usage for the last week

1. In the left pane of the Reporting tool, under Built-in Reports, expand System.
2. Click the report CPU vs. Memory Usage and HTTP Requests Rate.
3. In the right pane, on the report toolbar, click **Duration**, and then click **Last Week**.

To compare the bytes received rate and the bytes transmitted rate between the two interfaces for the last week

1. In the right pane, on the report toolbar, click Create.
2. In the **Report Name** box, type a name for the custom report (for example, Custom_Interfaces), and then click **OK**. The report is created with the default System Overview chart, which displays the CPU Usage counter plotted for the last hour.
3. Under System Overview, on the chart toolbar, click Counters.
4. In the counter selection pane, in Title, type a name for the chart (for example, Interfaces bytes data).
5. In Plot chart for, click System entities statistics, and then in Select Group, select Interface.
6. On the **Entities** tab, click one or more interface names you want to plot (for example, 1/1 and 1/2), and then click the > button.
7. On the Counters tab, click Bytes received (Rate) and Bytes transmitted (Rate) and then click the > button.
8. Click **OK**.
9. On the report toolbar, click **Duration**, and then click **Last Week**.

Stopping and starting the data collection utility

The data collection utility, `nscollect`, runs automatically when you start the Citrix ADC. This utility retrieves the application performance data and stores it in the form of data sources on the ADC. You

can create up to 32 data sources. The default data source is `/var/log/db/default`.

The data collection utility creates databases for global counters and entity-specific counters, and uses this data to generate reports. Global-counter databases are created at `/var/log/db/<DataSourceName>`. The entity-specific databases are created based on the entities configured on the Citrix ADC, and a separate folder is created for each entity type in `/var/log/db/<DataSourceName/EntityNameDB>`.

The `nscollect` retrieves data once every 5 minutes. It retains data in 5-minute granularity for one day, hourly for the last 30 days, and daily for three years.

You might have to stop and restart the data collection utility if data is not updated accurately or the reports display corrupted data.

To stop `nscollect`

At the command prompt, type:

```
/netscaler/nscollect stop
```

To start nscollect on the current SSH session to the Citrix ADC:

At the command prompt, type:

```
/netscaler/nscollect start
```

To start nscollect on the local system:

At the command prompt, type:

```
/netscaler/nscollect start &
```

CloudBridge Connector

September 14, 2021

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

The CloudBridge Connector feature of the Citrix ADC appliance connects enterprise datacenters to external clouds and hosting environments, making the cloud a secure extension of your enterprise network. Cloud-hosted applications appear as though they are running on one contiguous enterprise network. With Citrix CloudBridge Connector, you can augment your datacenters with the capacity and efficiency available from cloud providers.

The CloudBridge Connector enables you to move your applications to the cloud to reduce costs and increase reliability.

In addition to using CloudBridge Connector between a datacenter and a cloud, you can use it to connect two datacenters for a high-capacity secure and accelerated link.

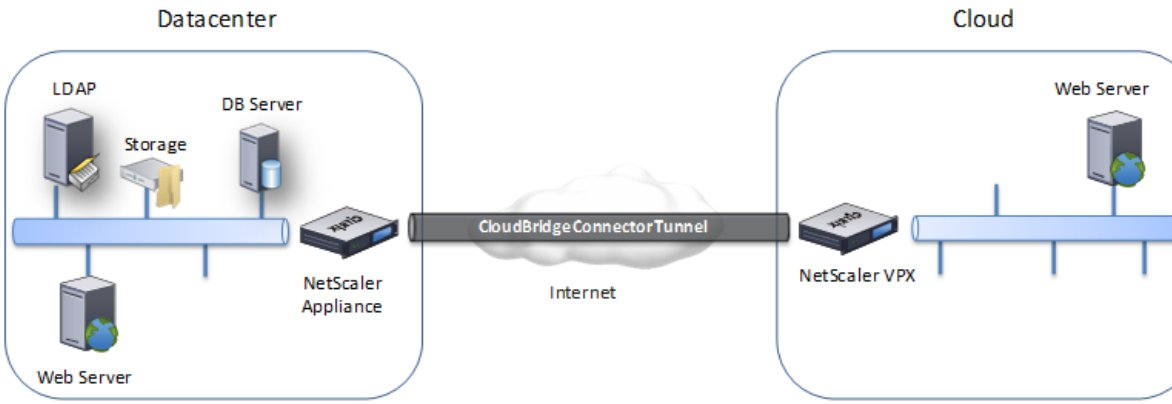
Understanding CloudBridge Connector

To implement the Citrix CloudBridge Connector solution, you connect a datacenter to another datacenter or an external cloud by setting up a tunnel called the CloudBridge Connector tunnel.

To connect a datacenter to another datacenter, you set up a CloudBridge Connector tunnel between two Citrix ADC appliances, one in each datacenter.

To connect a datacenter to an external cloud (for example, Amazon AWS cloud), you set up a CloudBridge Connector tunnel between a Citrix ADC appliance in the datacenter and a virtual appliance (VPX) that resides in the Cloud. The remote end point can be a CloudBridge Connector or a Citrix ADC VPX with Premium license.

The following illustration shows a CloudBridge Connector tunnel set up between a datacenter and an external cloud.



The appliances between which a CloudBridge Connector tunnel is set up are called the *end points* or *peers* of the CloudBridge Connector tunnel.

A CloudBridge Connector tunnel uses the following protocols:

- Generic Routing Encapsulation (GRE) protocol
- Open-standard IPSec Protocol suite, in transport mode

The GRE protocol provides a mechanism for encapsulating packets, from a wide variety of network protocols, to be forwarded over another protocol. GRE is used to:

- Connect networks running non-IP and non-routable protocols.
- Bridge across a wide area network (WAN).
- Create a transport tunnel for any type of traffic that needs to be sent unchanged across a different network.

The GRE protocol encapsulates packets by adding a GRE header and a GRE IP header to the packets.

The Internet Protocol security (IPSec) protocol suite secures communication between peers in the CloudBridge Connector tunnel.

In a CloudBridge Connector tunnel, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

IPSec uses the transport mode in which the GRE encapsulated packet is encrypted. The encryption is done by the Encapsulating Security Payload (ESP) protocol. The ESP protocol ensures the integrity of the packet by using a HMAC hash function, and ensures confidentiality by using an encryption algorithm. After the packet is encrypted and the HMAC is calculated, an ESP header is generated. The ESP header is inserted after the GRE IP header and, an ESP trailer is inserted at the end of the encrypted payload.

Peers in the CloudBridge Connector tunnel use the Internet Key Exchange version (IKE) protocol (part of the IPSec protocol suite) to negotiate secure communication, as follows:

- The two peers mutually authenticate with each other, using one of the following authentication methods:
  - **Pre-shared key authentication**. A text string called a pre-shared key is manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.
  - **Digital certificates authentication**. The initiator (sender) peer signs message interchange data by using its private key, and the other receiver peer uses the sender's public key to verify the signature. Typically, the public key is exchanged in messages containing an X.509v3 certificate. This certificate provides a level of assurance that a peer's identity as represented in the certificate is associated with a particular public key.
- The peers then negotiate to reach agreement on:
  - An encryption algorithm.
  - Cryptographic keys for encrypting data in one peer and decrypting the data in the other.

This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one-way (simplex). For example, when two peers, CB1 and CB2, are communicating through a Connector tunnel, CB1 has two Security Associations. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets.

SAs expire after a specified length of time, which is called the *lifetime*. The two peers use the Internet Key Exchange (IKE) protocol (part of the IPSec protocol suite) to negotiate new cryptographic keys and establish new SAs. The purpose of the limited lifetime is to prevent attackers from cracking a key.

The following table lists some IPSec propeties supported by a Citrix ADC appliance:

IPSec Properties	Types Supported
IKE Versions	V1, V2
IKE DH group	A Citrix ADC appliance supports only DH group 2 (1024 bits MODP algorithm) for both IKEv1 and IKEv2.
IKE Authentication Methods	Pre-shared key authentication, Digital certificates authentication
Encryption Algorithm	AES (128 bits), AES 256 (256 bits), 3DES
Hash Algorithm	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5

Monitoring CloudBridge Connector tunnels

September 14, 2021

You can display the statistics for monitoring the performance of a CloudBridge Connector tunnel. To display CloudBridge Connector tunnel statistics on a Citrix ADC appliance, use the GUI or the Citrix ADC command line.

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels on a Citrix ADC appliance.

Statistical counter	**Specifies**
Bytes Received	Total number of bytes received by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Sent	Total number of bytes sent by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.

Statistical counter	Specifies
Packets Received	Total number of packets received by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Sent	Total number of packets sent by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Received Rate	Number of bytes per second received by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels.
Bytes Sent Rate	Number of bytes per second sent by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels
Packets Received Rate	Number of bytes per second received by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels
Packets Sent Rate	Number of bytes per second received by the Citrix ADC appliance through all the configured CloudBridge Connector tunnels

All these counters are reset to 0 when the Citrix ADC appliance is restarted. They do not increment during the following phases:

- Internet Key Exchange (IKE) authentication (pre-shared key) phase on any configured CloudBridge Connector tunnel.
- IKE Security Association (SA) establishment phase on any configured CloudBridge Connector tunnel.

To display CloudBridge Connector tunnel statistics by using the Citrix ADC command line

At the command prompt, type:

- **stat ipsec counters**

To display CloudBridge Connector tunnel statistics by using the GUI

1. Access the GUI by using a web browser to connect to the IP address of the Citrix ADC appliance.
2. On the **Configuation** tab, navigate to **System** > **CloudBridge Connector**.

3. On the CloudBridge Connector page, click **Create/Monitor CloudBridge Connector**. The **IPSec Bytes** and **IPSec Packets** charts display the bytes received rate, bytes sent rate, packets received rate, and packets sent rate of all the CloudBridge Connector tunnels configured on the Citrix ADC appliance.

```
    > stat ipsec counters
    Secure tunnel(s) summary
                        Rate (/s)        Total
    Bytes Received    0    2811248
    Bytes Sent            0    157460630
    Packets Received    0    56787
    Packets Sent        0    200910
    Done
    >
<!--NeedCopy-->
```

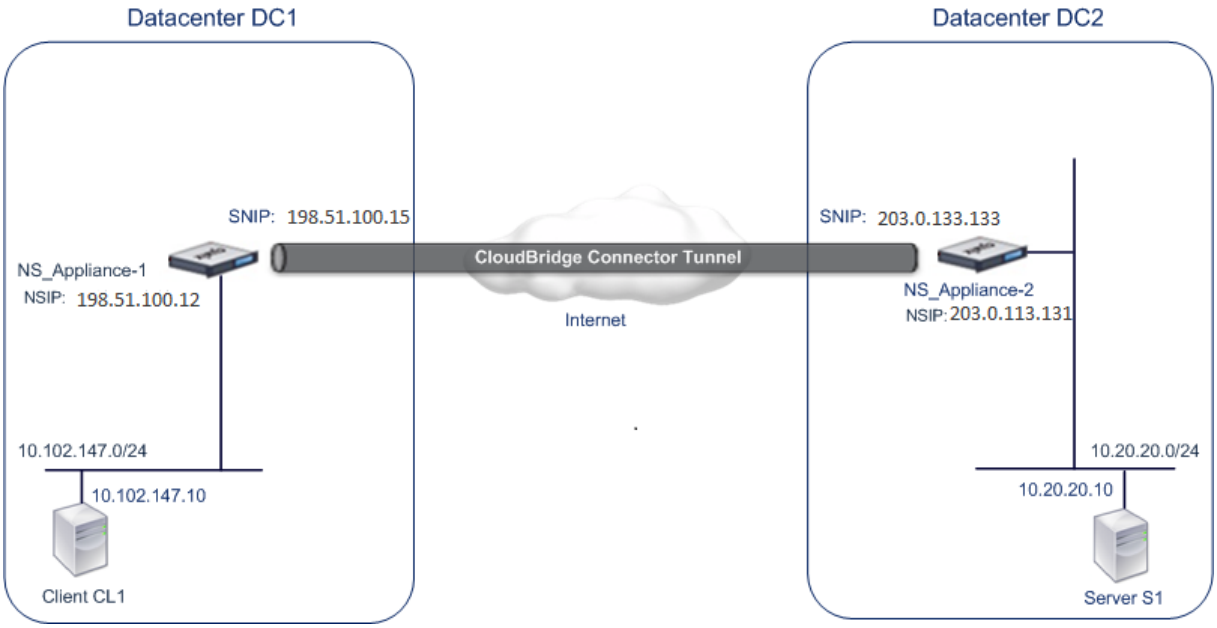
Configuring a CloudBridge Connector tunnel between two datacenters

September 14, 2021

You can configure a CloudBridge Connector tunnel between two different datacenters to extend your network without reconfiguring it, and leverage the capabilities of the two datacenters. A CloudBridge Connector tunnel between the two geographically separated datacenters enables you to implement redundancy and safeguard your setup from failure. The CloudBridge Connector tunnel helps achieve optimal utilization of infrastructure and resources across datacenters. The applications available across the two datacenters appear as local to the user.

To connect a datacenter to another datacenter, you set up a CloudBridge Connector tunnel between a Citrix ADC appliance in one datacenter and a Citrix ADC appliance in the other datacenter.

As an illustration of CloudBridge Connector tunnel between datacenters, consider an example in which a CloudBridge Connector tunnel is set up between Citrix ADC appliance NS_Appliance-1 in datacenter DC1 and Citrix ADC appliance NS_Appliance-2 in datacenter DC2.



Both NS_Appliance-1 and NS_Appliance-2 function in L2 and L3 mode. They enable communication between private networks in datacenters DC1 and DC2. In L3 mode, NS_Appliance-1 and NS_Appliance-2 enable communication between client CL1 in datacenter DC1 and server S1 in the datacenter DC2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

Because client CL1 and server S1 are on different private networks, L3 mode is enabled on NS_Appliance-1 and NS_Appliance-2, and routes are updated as follows:

- CL1 has a route to NS_Appliance-1 for reaching S1.
- NS_Appliance-1 has a route to NS_Appliance-2 for reaching S1.
- S1 has a route to NS_Appliance-2 for reaching CL1.
- NS_Appliance-2 has a route to NS_Appliance-1 for reaching CL1.

The following table lists the settings on Citrix ADC appliance NS_Appliance-1 in datacenter DC1.

The following table lists the settings on Citrix ADC appliance NS_Appliance-2 in datacenter DC2.

Entity	Name	Details
The NSIP address		198.51.100.12
SNIP address		198.51.100.15

Entity	Name	Details
CloudBridge Connector tunnel	Cloud_Connector_DC1-DC2	1. Local endpoint IP address of the CloudBridge Connector tunnel: 198.51.100.15, 2. Remote endpoint IP address of the CloudBridge Connector tunnel: 203.0.113.133. GRE Tunnel Details Name = Cloud_Connector_DC1-DC2, IPSec Profile Details Name = Cloud_Connector_DC1-DC2, Encryption algorithm = AES, Hash algorithm = HMAC SHA1

Points to consider for configuring CloudBridge Connector tunnel

Before setting up a CloudBridge Connector tunnel, verify that the following tasks have been completed:

1. Deploy and set up a Citrix ADC appliance in each of the two datacenters.
2. Make sure that the CloudBridge Connector tunnel end-point IP addresses are accessible to each other.

Configuration procedure

To set up a CloudBridge Connector tunnel between a Citrix ADC appliance that resides in one datacenter and another Citrix ADC appliance that resides in the other datacenter, use the GUI or the command line interface of one of the Citrix ADC appliances.

When you use the GUI, the CloudBridge Connector tunnel configuration created on the first Citrix ADC appliance is automatically pushed to the other endpoint (the other Citrix ADC appliance) of the CloudBridge Connector tunnel. Therefore, you do not have to access the GUI of the other Citrix ADC appliance to create the corresponding CloudBridge Connector tunnel configuration on it.

The CloudBridge Connector tunnel configuration on each of the Citrix ADC appliances consists of the following entities:

- **IPSec profile**—An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in the CloudBridge Connector tunnel.

- **GRE tunnel**—An IP tunnel specifies the local IP address (a public SNIP address configured on the local Citrix ADC appliance), remote IP address (a public SNIP address configured on the remote Citrix ADC appliance), protocol (GRE) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity.
- **Create a PBR rule and associate the IP tunnel with it**—A PBR entity specifies a set of conditions and an IP tunnel entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range and the destination IP address range to specify the subnet whose traffic is to traverse the CloudBridge Connector tunnel. For example, consider a request packet that originates from a client on the subnet in the first datacenter and is destined to a server on the subnet in the second datacenter. If this packet matches the source and destination IP address range of the PBR entity on the Citrix ADC appliance in the first datacenter, it is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the command line interface

At the command prompt, type:

- `add ipsec profile <name> [-ikeVersion (V1 | V2)] [-encAlgo (AES | 3 DES)...] [-hashAlgo <hashAlgo\> ...] [-lifetime <positive_integer>] (-psk | (-publickey<string> -privatekey <string>-peerPublicKey <string >))[-livenessCheckInterval <positive_intege>][-replayWindowSize \< positive_integer>] [-ikeRetryInterval <positive_integer>] [-retransmissiontime <positive_integer>]`
- `show ipsec profile <name>`

To create an IP tunnel and bind the IPSEC profile to it by using the command line interface

At the command prompt, type:

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol < protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the command line interface

At the command prompt, type:

- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = < remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

Example

```
1  add ipsec profile Cloud_Connector_DC1-DC2  -encAlgo AES -hashAlgo
       HMAC_SHA1
```

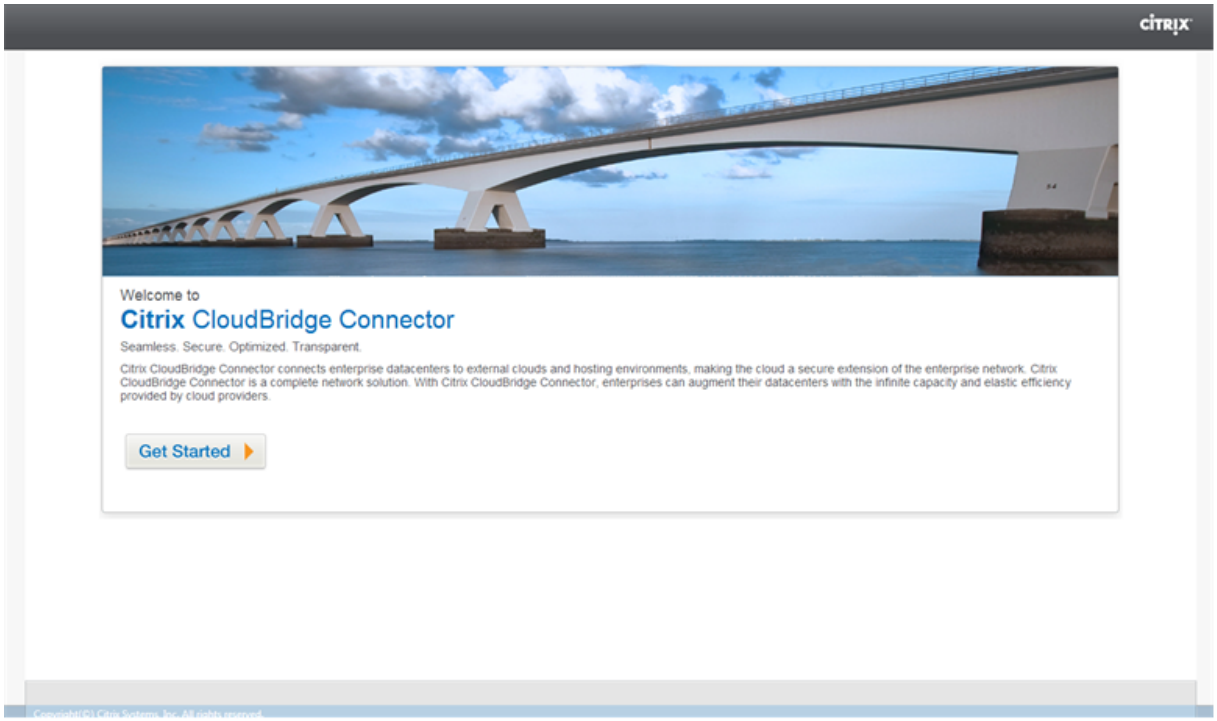


```
2      Done
3      > add ipTunnel Cloud_Connector_DC1-DC2 203.0.113.133
           255.255.255.255 198.51.100.15 -protocol GRE -ipsecProfileName
           Cloud_Connector_DC1-DC2
4
5      Done
6      > add ns pbr PBR-DC1-DC2 ALLOW -srcIP 198.51.100.15 -destIP
           203.0.113.133 ipTunnel Cloud_Connector_DC1-DC2
7
8      Done
9      > apply ns pbrs
10
11     Done
12 <!--NeedCopy-->
```

To configure a CloudBridge Connector tunnel in a Citrix ADC appliance by using the GUI

1. Type the NSIP address of a Citrix ADC appliance in the address line of a web browser.
2. Log on to the GUI of the Citrix ADC appliance by using your account credentials for the appliance.
3. Navigate to **System** > **CloudBridge Connector**.
4. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.

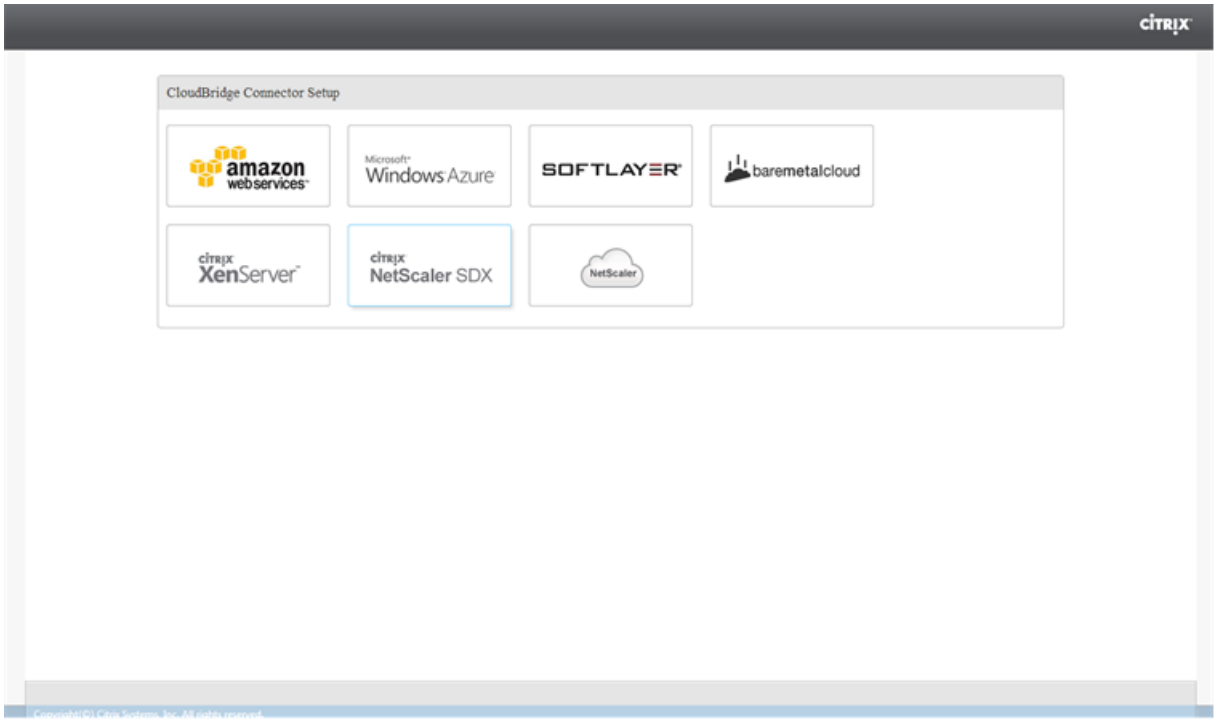
   The first time you configure a CloudBridge Connector tunnel on the appliance, a **Welcome** screen appears.
5. On the **Welcome** screen click **Get Started**.



Note:

If you already have a CloudBridge Connector tunnel configured on the Citrix ADC appliance, the Welcome screen does not appear, so you do not click Get Started.

1. In the **CloudBridge Connector Setup** pane, click **Citrix ADC**.



1. In the Citrix ADC pane, provide your account credentials for the remote Citrix ADC appliance. Click **Continue**.
2. In the **CloudBridge Connector Setting** pane, set the following parameter:
   - **CloudBridge Connector Name**—Name for the CloudBridge Connector configuration on the local appliance. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CloudBridge Connector configuration is created.
3. Under **Local Setting**, set the following parameter:
   - **Subnet IP**—IP address of the local endpoint of the CloudBridge Connector tunnel.
4. Under **Remote Setting**, set the following parameter:
   - **Subnet IP**—IP address of the peer endpoint of the CloudBridge Connector tunnel.
5. Under **PBR** Setting, set the following parameters:
   - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
   - **Source IP Low** —Lowest source IP address to match against the source IP address of an outgoing IPv4 packet.
   - **Source IP High**—Highest source IP address to match against the source IP address of an outgoing IPv4 packet.
   - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
   - Destination IP Low*—Lowest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
   - **Destination IP High**—Highest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
6. (Optional) Under **Security Settings**, set the following IPSec protocol parameters for the CloudBridge Connector tunnel:
   - **Encryption Algorithm**—Encryption algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
   - **Hash Algorithm**—Hash algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
   - **Key**—Select one of the following IPSec authentication methods to be used by the two peers to mutually authenticate.
     - **Auto Generate Key**—Authentication based on a text string, called a pre-shared key (PSK), generated automatically by the local appliance. The PSKs keys of the peers are matched against each other for authentication.
     - **Specific Key**—Authentication based on a manually entered PSK. The PSKs of the peers are matched against each other for authentication.

        * Pre Shared Security Key—The text string entered for pre-shared key based authentication.
    - **Upload Certificates**—Authentication based on digital certificates.
        * **Public Key**—A local digital certificate to be used to authenticate the local Citrix ADC appliance to the peer before establishing IPSec security associations. The same certificate should be present and set for the Peer Public Key parameter in the peer.
        * **Private Key**—Private key of the local digital certificate.
        * **Peer Public Key**—Digital certificate of the peer. Used to authenticate the peer to the local end point before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the peer.

7. Click **Done**.

The new CloudBridge Connector tunnel configuration on both the Citrix ADC appliances appears on the Home tab of the respective GUI. The current status of the CloudBridge connector tunnel is indicated in the Configured CloudBridge Connectors pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector Tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

Configuring CloudBridge Connector between datacenter and AWS cloud

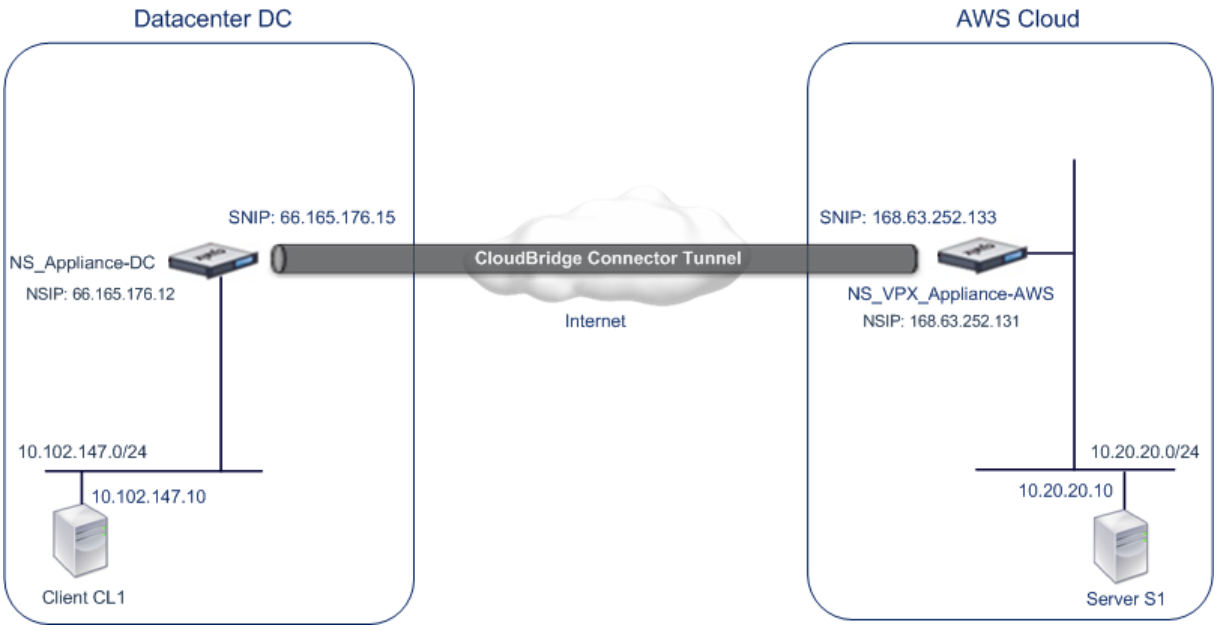
September 14, 2021

You can configure a CloudBridge Connector tunnel between a datacenter and AWS cloud to leverage the infrastructure and computing capabilities of the data center and the AWS cloud. With AWS, you can extend your network without initial capital investment or the cost of maintaining the extended network infrastructure. You can scale your infrastructure up or down, as required. For example, you can lease more server capabilities when the demand increases.

To connect a datacenter to AWS cloud, you set up a CloudBridge Connector tunnel between a Citrix ADC appliance that resides in the datacenter and a Citrix ADC virtual appliance (VPX) that resides in AWS cloud.

As an illustration of a CloudBridge Connector tunnel between a datacenter and Amazon AWS cloud,

consider an example in which a CloudBridge Connector tunnel is set up between Citrix ADC appliance NS_Appliance-DC, in datacenter DC, and Citrix ADC virtual appliance (VPX) NS_VPX_Appliance-AWS.



Both NS_Appliance-DC and NS_VPX_Appliance-AWS function in L3 mode. They enable communication between private networks in datacenter DC and the AWS cloud. NS_Appliance-DC and NS_VPX_Appliance-AWS enable communication between client CL1 in datacenter DC and server S1 in the AWS cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

> **Note**:
>
> AWS does not support L2 mode, hence it is necessary to have only L3 mode enabled on both the endpoints.

For proper communication between CL1 and S1, L3 mode is enabled on NS_Appliance-DC and NS_VPX_Appliance-AWS and routes are updated as such:

- CL1 have a route to NS_Appliance-DC for reaching S1.
- NS_Appliance-DC have a route to NS_VPX_Appliance-AWS for reaching S1.
- S1 should have a route to NS_VPX_Appliance-AWS for reaching CL1.
- NS_VPX_Appliance-AWS have a route to NS_Appliance-DC for reaching CL1.

The following table lists the settings on Citrix ADC appliance NS_Appliance-DC in datacenter DC.

Entity	Name	Details
The NSIP address		66.165.176.12
SNIP address		66.165.176.15

Entity	Name	Details
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	Local endpoint IP address of the CloudBridge Connector tunnel: 66.165.176.15, Remote endpoint IP address of the CloudBridge Connector tunnel: 168.63.252.133, GRE Tunnel Details - Name= CC_Tunnel_DC-AWS

The following table lists the settings on Citrix ADC VPX NS_VPX_Appliance-AWS on AWS cloud.

Entity	Name	Details
NSIP address		10.102.25.30
Public EIP address mapped to the NSIP address		168.63.252.131
SNIP address		10.102.29.30
Public EIP address mapped to the SNIP address		168.63.252.133
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	Local endpoint IP address of the CloudBridge Connector tunnel:168.63.252.133, Remote endpoint IP address of the CloudBridge Connector tunnel: 66.165.176.15; **GRE Tunnel Details** Name= CC_Tunnel_DC-AWS, IPSec Profile Details, Name= CC_Tunnel_DC-AWS, Encryption algorithm= AES, Hash algorithm= HMAC SHA1

Prerequisites

Before setting up a CloudBridge Connector tunnel, verify that the following tasks have been completed:

1. Install, configure, and launch an instance of Citrix ADC Virtual appliance (VPX) on AWS cloud. For instructions on installing Citrix ADC VPX on AWS, see Deploy a Citrix ADC VPX instance on AWS.
2. Deploy and configure a Citrix ADC physical appliance, or provisioning and configuring a Citrix ADC virtual appliance (VPX) on a virtualization platform in the datacenter.
3. Make sure that the CloudBridge Connector tunnel end-point IP addresses are accessible to each other.

Citrix ADC VPX license

After the initial instance launch, Citrix ADC VPX for AWS requires a license. If you are bringing your own license (BYOL), see the VPX Licensing Guide at: http://support.citrix.com/article/CTX122426.

You have to:

1. Use the licensing portal within Citrix website to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance will activate automatically.

Configuration steps

To set up a CloudBridge Connector tunnel between a Citrix ADC appliance that resides in a datacenter and a Citrix ADC virtual appliance (VPX) that resides on the AWS cloud, use the GUI of the Citrix ADC appliance.

When you use the GUI, the CloudBridge Connector tunnel configuration created on the Citrix ADC appliance, is automatically pushed to the other endpoint or peer (the Citrix ADC VPX on AWS) of the CloudBridge Connector tunnel. Therefore, you do not have to access the GUI (GUI) of the Citrix ADC VPX on AWS to create the corresponding CloudBridge Connector tunnel configuration on it.

The CloudBridge Connector tunnel configuration on both peers (the Citrix ADC appliance that resides in the datacenter and the Citrix ADC virtual appliance (VPX) that resides on the AWS cloud) consists of the following entities:

- **IPSec profile**—An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in both the peers of the CloudBridge Connector tunnel.
- **GRE tunnel**—An IP tunnel specifies a local IP address (a public SNIP address configured on the local peer), remote IP address (a public SNIP address configured on the remote peer), protocol (GRE) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity.

- **Create a PBR rule and associate the IP tunnel with it**—A PBR entity specifies a set of conditions and an IP tunnel entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range and the destination IP address range to specify the subnet whose traffic is to traverse the CloudBridge Connector tunnel. For example, consider a request packet that originates from a client on the subnet in the datacenter and is destined to a server on the subnet in the AWS cloud. If this packet matches the source and destination IP address range of the PBR entity on the Citrix ADC appliance in the datacenter, it is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the command line interface

At the command prompt, type:

- add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES)...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>))[-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]
- **show ipsec profile** <name>

To create an IP tunnel and bind the IPSEC profile to it by using the command line interface

At the command prompt, type:

- add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]
- show ipTunnel <name>

To create a PBR rule and bind the IPSEC tunnel to it by using the command line interface

At the command prompt, type:

- add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>
- apply ns pbrs
- show ns pbr <pbr_name>

Example

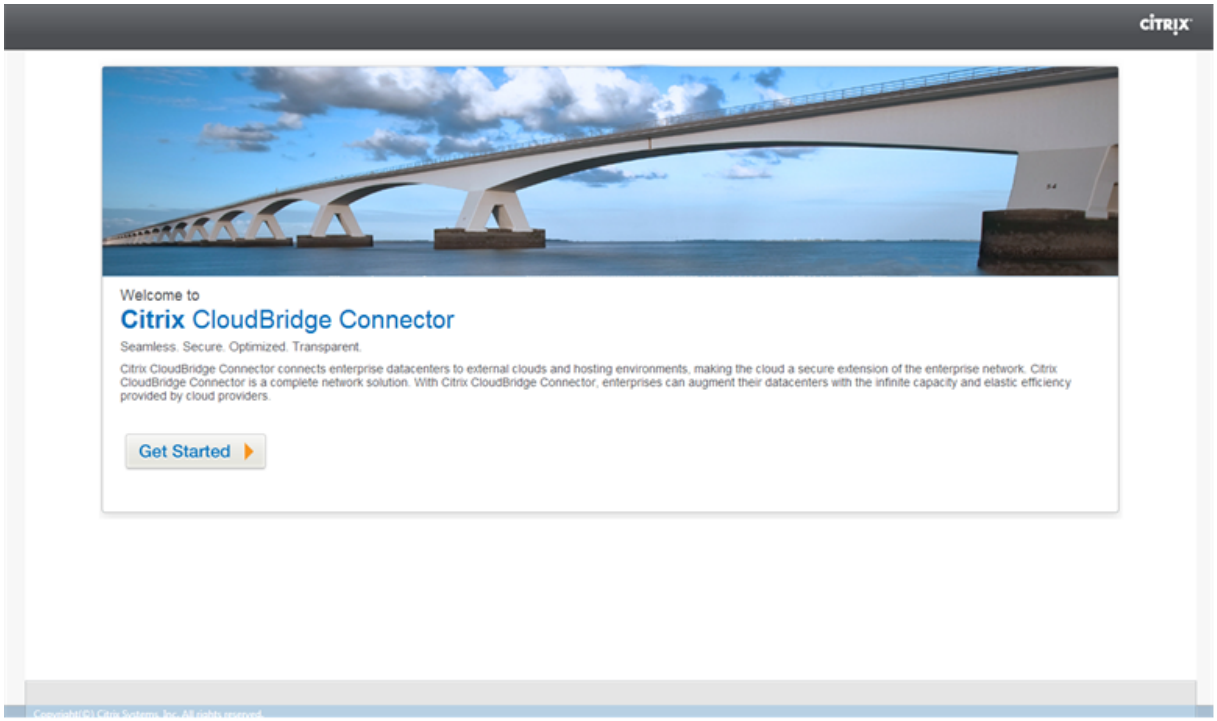
```
1     > add ipsec profile CC_Tunnel_DC-AWS -encAlgo AES -hashAlgo
          HMAC_SHA1
2
3     Done
4     > add ipTunnel CC_Tunnel_DC-AWS 168.63.252.133 255.255.255.0
          66.165.176.15  - protocol GRE -ipsecProfileName CC_Tunnel_DC-AWS
```



```
5
6       Done
7       > add ns pbr PBR-DC-AWS ALLOW  -srcIP 66.165.176.15  -destIP
            168.63.252.133 ipTunnel CC_Tunnel_DC-AWS
8
9       Done
10      > apply ns pbrs
11
12      Done
13  <!--NeedCopy-->
```

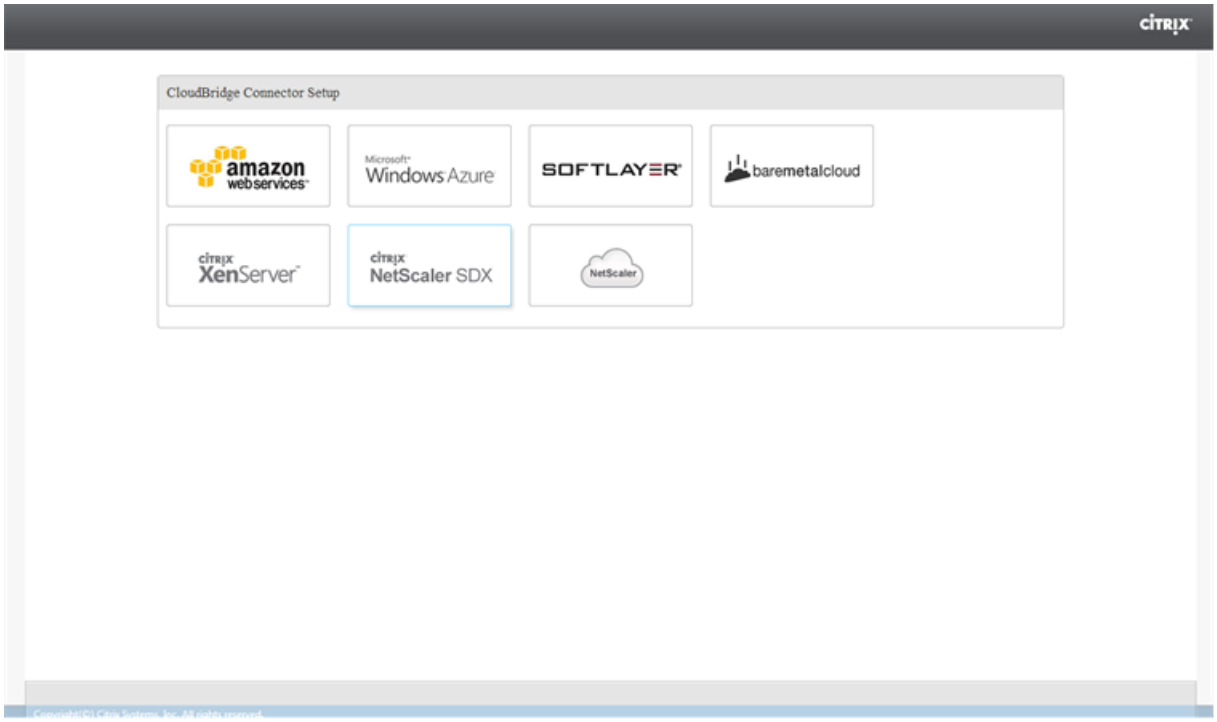
To configure a CloudBridge Connector tunnel in a Citrix ADC appliance by using the GUI

1. Type the NSIP address of a Citrix ADC appliance in the address line of a web browser.
2. Log on to the GUI of the Citrix ADC appliance by using your account credentials for the appliance.
3. Navigate to **System** > **CloudBridge Connector**.
4. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.
5. The first time you configure a CloudBridge Connector tunnel on the appliance, a **Welcome** screen appears.
6. On the **Welcome** screen click **Get Started**.



> **Note**:
>
> If you already have a CloudBridge Connector tunnel configured on the Citrix ADC appliance, the Welcome screen does not appear, so you do not click Get Started.

1. In the **CloudBridge Connector Setup** pane, click **amazon web services**



1. In the **Amazon** pane, provide your AWS account credentials: AWS Access Key ID and AWS Secret Access Key. You can obtain these access keys from the AWS GUI console. Click **Continue**.

> **Note**
>
> Earlier, the Setup wizard always connects to the same AWS region even when another region is selected. As a result, configuring CloudBridge Connector tunnel to a Citrix ADC VPX running on the selected AWS region used to fail. This issue has been fixed now.

1. In the **Citrix ADC** pane, select the NSIP address of the Citrix ADC virtual appliance running on AWS. Then, provide your account credentials for the Citrix ADC virtual appliance. Click **Continue**.
2. In the **CloudBridge Connector Setting** pane, set the following parameter:
   - **CloudBridge Connector Name**—Name for the CloudBridge Connector configuration on the local appliance. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CloudBridge Connector configuration is created.

3. Under **Local Setting**, set the following parameter:

    - **Subnet IP**—IP address of the local endpoint of the CloudBridge Connector tunnel. Must be a public IP address of type SNIP.

4. Under **Remote Setting**, set the following parameter:

    - **Subnet IP**—IP address of the CloudBridge Connector tunnel end point on the AWS side. Must be an IP address of type SNIP on the Citrix ADC VPX instance on AWS.
    - **NAT**—Public IP address (EIP) in AWS that is mapped to the SNIP configured on the Citrix ADC VPX instance on AWS.

5. Under **PBR Setting**, set the following parameters:

    - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
    - **Source IP Low**—Lowest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Source IP High**—Highest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
    - **Destination IP Low**—Lowest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
    - **Destination IP High**—Highest destination IP address to match against the destination IP address of an outgoing IPv4 packet.

6. (Optional) Under **Security Settings**, set the following IPSec protocol parameters for the CloudBridge Connector tunnel:

    - **Encryption Algorithm**—Encryption algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Hash Algorithm**—Hash algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Key**— Select one of the following IPSec authentication methods to be used by the two peers to mutually authenticate.
        - **Auto Generate Key**— Authentication based on a text string, called a pre-shared key (PSK), generated automatically by the local appliance. The PSKs keys of the peers are matched against each other for authentication.
        - **Specific Key**—Authentication based on a manually entered PSK. The PSKs of the peers are matched against each other for authentication.
            * **Pre Shared Security Key**—The text string entered for pre-shared key based authentication.
        - **Upload Certificates**—Authentication based on digital certificates.
            * **Public Key**—A local digital certificate to be used to authenticate the local peer to the remote peer before establishing IPSec security associations. The same certifi-

cate should be present and set for the Peer Public Key parameter in the peer.

* **Private Key**—Private key of the local digital certificate.
* **Peer Public Key**—Digital certificate of the peer. Used to authenticate the peer to the local end point before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the peer.

7. Click **Done**.

The new CloudBridge Connector tunnel configuration on the Citrix ADC appliance in the datacenter appears on the Home tab of the GUI. The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC VPX appliance in the AWS cloud appears on the GUI. The current status of the CloudBridge connector tunnel is indicated in the Configured CloudBridge pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

Configuring a CloudBridge Connector tunnel between a Citrix ADC appliance and virtual private gateway on AWS

September 14, 2021

To connect a datacenter to Amazon Web Services (AWS), you can configure a CloudBridge Connector tunnel between a Citrix ADC appliance in the datacenter and a virtual private gateway on AWS. The Citrix ADC appliance and the virtual private gateway form the endpoints of the CloudBridge Connector tunnel and are called peers.

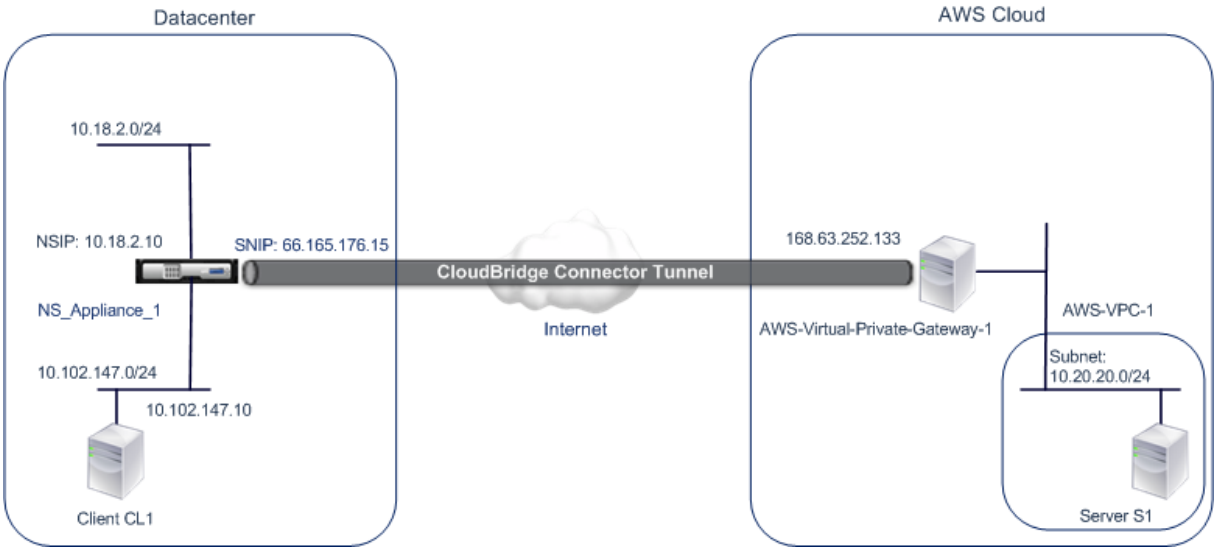
> **Note:**
>
> You can also set up a CloudBridge Connector tunnel between a Citrix ADC appliance in a datacenter and a Citrix ADC VPX instance (instead of a virtual private gateway) on AWS. For more information, see Configuring CloudBridge Connector between Datacenter and AWS Cloud.

Virtual private gateways on AWS support the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings when you configure the Citrix ADC appliance for the CloudBridge Connector tunnel.

IPSec Properties	Setting
IPSec mode	Tunnel mode
IKE version	Version 1
IKE Authentication method	Pre-Shared Key
Encryption algorithm	AES
Hash algorithm	HMAC SHA1

Example of CloudBridge Connector tunnel configuration and data flow

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between Citrix ADC appliance NS_Appliance-1 in a datacenter and virtual private gateway gateway AWS-Virtual-Private-Gateway-1 on AWS cloud.



NS_Appliance-1 also functions as an L3 router, which enables a private network in the datacenter to reach a private network in the AWS cloud through the CloudBridge Connector tunnel. As a router, NS_Appliance-1 enables communication between client CL1 in the datacenter and server S1 in the AWS cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes an IPSec profile entity named NS_AWS_IPSec_Profile, a CloudBridge Connector tunnel entity named NS_AWS_Tunnel, and a policy based routing (PBR) entity named NS_AWS_Pbr.

The IPSec profile entity NS_AWS_IPSec_Profile specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, and hash algorithm, to be used by the IPSec protocol in the CloudBridge Connector tunnel. NS_AWS_IPSec_Profile is bound to IP tunnel entity NS_AWS_Tunnel.

CloudBridge Connector tunnel entity NS_AWS_Tunnel specifies the local IP address (a public IP—SNIP—address configured on the Citrix ADC appliance), the remote IP address (the IP address of the AWS-Virtual-Private-Gateway-1), and the protocol (IPSec) used to set up the CloudBridge Connector tunnel. NS_AWS_Tunnel is bound to policy based routing (PBR) entity NS_AWS_Pbr.

The PBR entity NS_AWS_Pbr specifies a set of conditions and a CloudBridge Connector tunnel entity (NS_AWS_Tunnel). The source IP address range and the destination IP address range are the conditions for NS_AWS_Pbr. The source IP address range and the destination IP address range are specified as a subnet in the datacenter and a subnet in the AWS cloud, respectively. Any request packet originating from a client in the subnet in the datacenter and destined to a server in the subnet on the AWS cloud matches the conditions in NS_AWS_Pbr. This packet is then considered for CloudBridge Connector processing and is sent across the CloudBridge Connector tunnel (NS_AWS_Tunnel) bound to the PBR entity.

The following table lists the settings used in this example.

IP address of the CloudBridge Connector tunnel end point (NS_Appliance-1) in the datacenter side	66.165.176.15
IP address of the CloudBridge Connector tunnel end point (AWS-Virtual-Private-Gateway-1) in the AWS	168.63.252.133
Datacenter Subnet, the traffic of which is to traverse the CloudBridge Connector tunnel	10.102.147.0/24
AWS Subnet, the traffic of which is to traverse the CloudBridge Connector tunnel	10.20.20.0/24

Settings on Amazon AWS

Customer Gateway	AWS-Customer-Gateway-1	Routing = Static, IP Address = Internet-routable CloudBridge Connector tunnel endpoint IP address on the Citrix ADC side = 66.165.176.15
Virtual Private Gateway	AWS-Virtual-Private-Gateway-1	Associated VPC = AWS-VPC-1

Customer Gateway	AWS-Customer-Gateway-1	Routing = Static, IP Address = Internet-routable CloudBridge Connector tunnel endpoint IP address on the Citrix ADC side = 66.165.176.15
VPN Connection	AWS-VPN-Connection-1	Customer Gateway = AWS-Customer-Gateway-1, Virtual Private Gateway= Virtual-Private-Gateway-1, Routing Options: Type = Static, Static IP Prefixes = Subnets on the Citrix ADC side = 10.102.147.0/24

Settings on Citrix ADC appliance NS_Appliance-1 in Datacenter-1:

Appliance	Settings	
SNIP1(for reference purposes only)	66.165.176.15	
IPSec profile	NS_AWS_IPSec_Profile	IKE version = v1, Encryption algorithm = AES, Hash algorithm = HMAC SHA1
CloudBridge Connector tunnel	NS_AWS_Tunnel	Remote IP = 168.63.252.133, Local IP= 66.165.176.15, Tunnel protocol = IPSec, IPSec profile= NS_AWS_IPSec_Profile
Policy based route	NS_AWS_Pbr	Source IP range = Subnet in the datacenter =10.102.147.0-10.102.147.255, Destination IP range =Subnet in AWS =10.20.20.0-10.20.20.255, IP Tunnel = NS_AWS_Tunnel

Points to consider for a CloudBridge Connector tunnel configuration

Before configuring a CloudBridge Connector tunnel between a Citrix ADC appliance and AWS gateway, consider the following points:

1. AWS supports the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings when you configure the Citrix ADC appliance for the CloudBridge Connector tunnel.
    - IKE version = v1
    - Encryption algorithm = AES
    - Hash algorithm = HMAC SHA1
2. You must configure the firewall at the Citrix ADC end to allow the following.
    - Any UDP packets for port 500
    - Any UDP packets for port 4500
    - Any ESP (IP protocol number 50) packets
3. You must configure Amazon AWS before specifying the tunnel configuration on the Citrix ADC, because the public IP address of the AWS end (gateway) of the tunnel and the PSK are automatically generated when you set up the tunnel configuration in AWS. You need this information for specifying the tunnel configuration on the Citrix ADC appliance.
4. AWS gateway supports static routes and the BGP protocol for route updates. The Citrix ADC appliance does not support the BGP protocol in a CloudBridge Connector tunnel to AWS gateway. Therefore, appropriate static routes must be used on both sides of the CloudBridge Connector tunnel for proper routing of traffic through the tunnel.

Configuring Amazon AWS for the CloudBridge Connector tunnel

To create a CloudBridge Connector tunnel configuration on Amazon AWS, use the Amazon AWS Management Console, which is a web based graphical interface for creating and managing resources on Amazon AWS.

Before you begin the CloudBridge Connector tunnel configuration on AWS cloud, make sure that:

- You have a user account for Amazon AWS cloud.
- You have a virtual private cloud whose networks you want to connect to the networks at the Citrix ADC side through the CloudBridge Connector tunnel.
- You are familiar with the Amazon AWS Management Console.

> **Note**:
>
> The procedures for configuring Amazon AWS for a CloudBridge Connector tunnel might change over time, depending on the Amazon AWS release cycle. Citrix recommends you refer Amazon

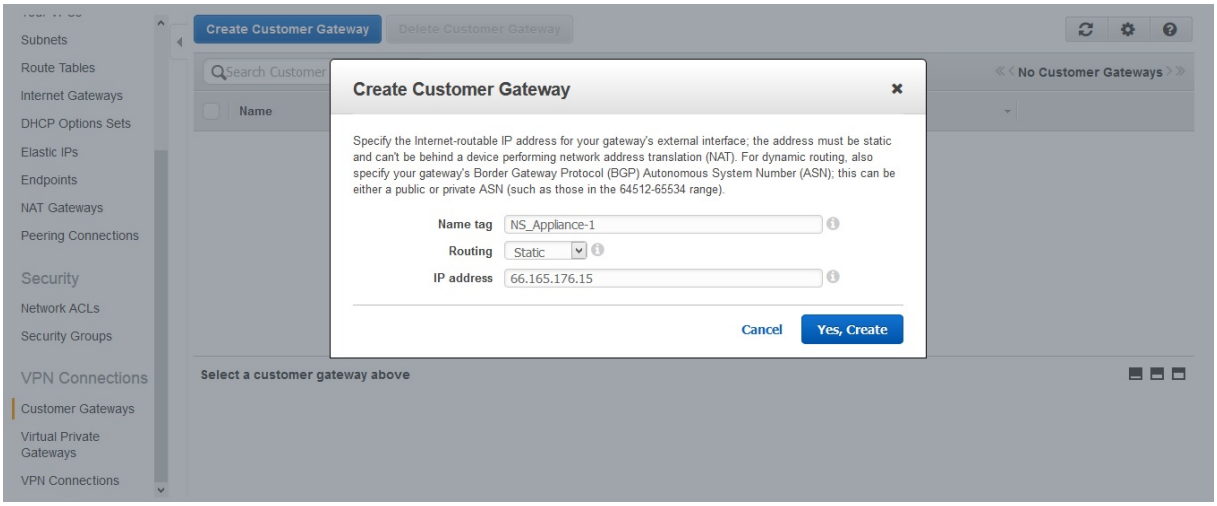
> AWS documentation for the latest procedures.

To configure a CloudBridge connector tunnel between a Citrix ADC and AWS gateway perform the following tasks on the AWS Management Console:

- **Create a Customer Gateway**. A customer gateway is an AWS entity that represents a CloudBridge Connector tunnel endpoint. For a CloudBridge Connector tunnel between a Citrix ADC appliance and AWS gateway, the customer gateway represents the Citrix ADC appliance on AWS. The customer gateway specifies a name, the type of routing (static or BGP) used in the tunnel, and the CloudBridge Connector tunnel endpoint IP address on the Citrix ADC side. The IP address can be an Internet-routable Citrix ADC owned subnet IP (SNIP) address or, if the Citrix ADC appliance is behind a NAT device, an Internet-routable NAT IP address that represents the SNIP address.
- **Create a Virtual Private Gateway and attach it to a VPC**. A virtual private gateway is a CloudBridge Connector tunnel endpoint at the AWS side. When you create a virtual private gateway, you assigned it a name or allow AWS to assign the name. You then associate the virtual private gateway with a VPC. This association enables the subnets of the VPC to connect to the subnets at the Citrix ADC side through the CloudBridge Connector tunnel.
- **Create a VPN Connection**. A VPN connection specifies a customer gateway and a virtual private gateway between which a CloudBridge Connector tunnel is to be created. It also specifies an IP prefix for the networks at the Citrix ADC side. Only IP prefixes that are known to the virtual private gateway (through static route entry) can receive traffic from the VPC through the tunnel. Also, the virtual private gateway does not route any traffic not destined to the specified IP prefixes through the tunnel. After configuring a VPN connection, you might have to wait few minutes for it to be created.
- **Configure Routing Options**. For the VPC's network to reach the networks at the Citrix ADC side through the CloudBridge Connector tunnel, you must configure the VPC's routing table to include routes for the networks at the Citrix ADC side and point those routes to the virtual private gateway. You can include routes in a VPC's routing table in one of the following ways:
  - **Enable Route Propagation**. You can enable route propagation for your routing table, so that routes are automatically propagated to the table. The static IP prefixes that you specify for VPN configuration are propagated to the routing table after you've created the VPN connection.
  - **Enter Static Routes Manually**. If you do not enable route propagation, you must manually enter the static routes for the networks at the Citrix ADC side.
- **Download Configuration**. After the CloudBridge Connector tunnel (VPN connection) configuration is created on AWS, download the configuration file of the VPN connection to your local system. You might need the information in the configuration file for configuring the CloudBridge Connector tunnel on the Citrix ADC appliance.

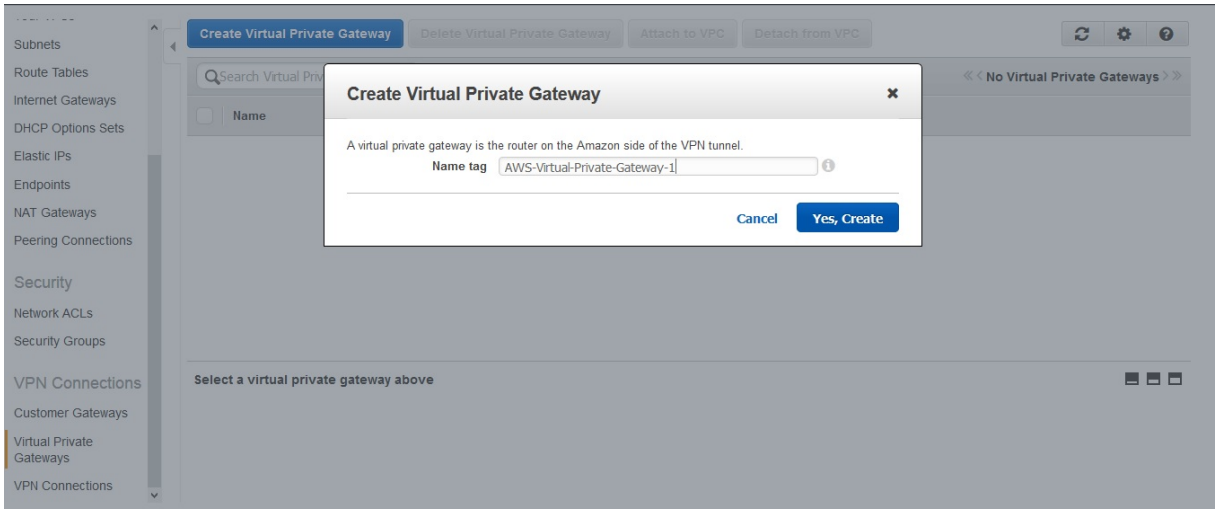
To create a customer gateway

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Navigate to **VPN Connections** > **Customer Gateways** and click on **Create Customer Gateway**.
3. In the **Create Customer Gateway** dialog box, set the following parameters and then click **Yes, Create**:
    - **Name tag**. A name for the customer gateway.
    - **Routing list**. Type of routing between Citrix ADC appliance and AWS virtual private gateway for advertising routes to each other through the CloudBridge Connector tunnel. Select **Static Routing** from the **Routing** list. **Note**: The Citrix ADC appliance does not support the BGP protocol in a CloudBridge Connector tunnel to AWS gateway. Therefore, appropriate static routes must be used on both sides of the CloudBridge Connector tunnel for proper routing of traffic through the tunnel.
    - **IP Address**. Internet-routable CloudBridge Connector tunnel endpoint IP address on the Citrix ADC side. The IP address can be an Internet-routable Citrix ADC owned subnet IP (SNIP) address or, if the Citrix ADC appliance is behind a NAT device, an Internet-routable NAT IP address that represents the SNIP address.

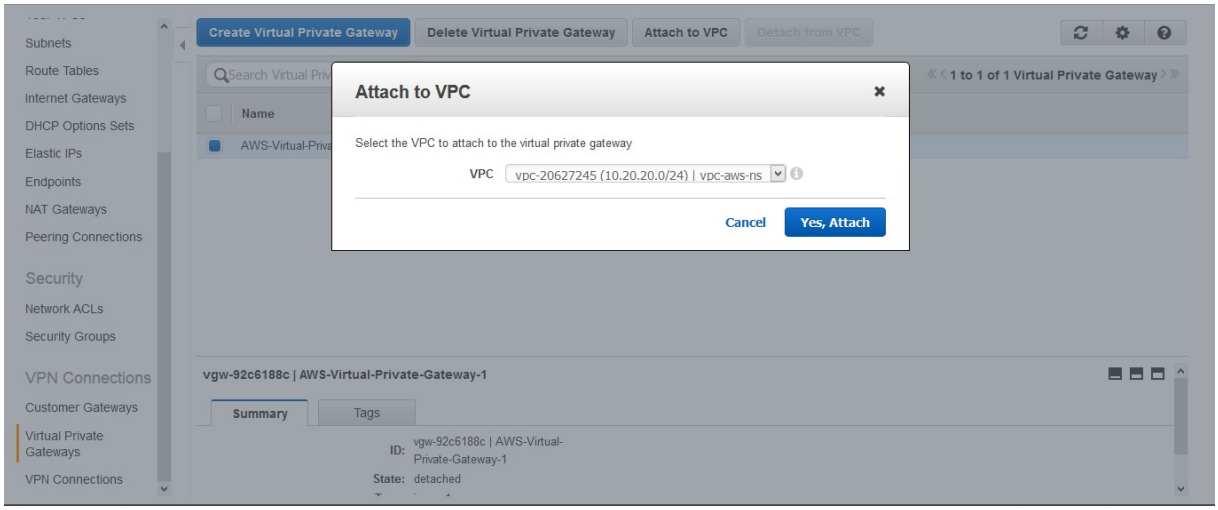


To create a virtual private gateway and attach it to a VPC

1. Navigate to **VPN Connections** > **Virtual Private Gateways**, and then click Create Virtual Private Gateway.
2. Enter a name for the virtual private gateway, and then click Yes, Create.

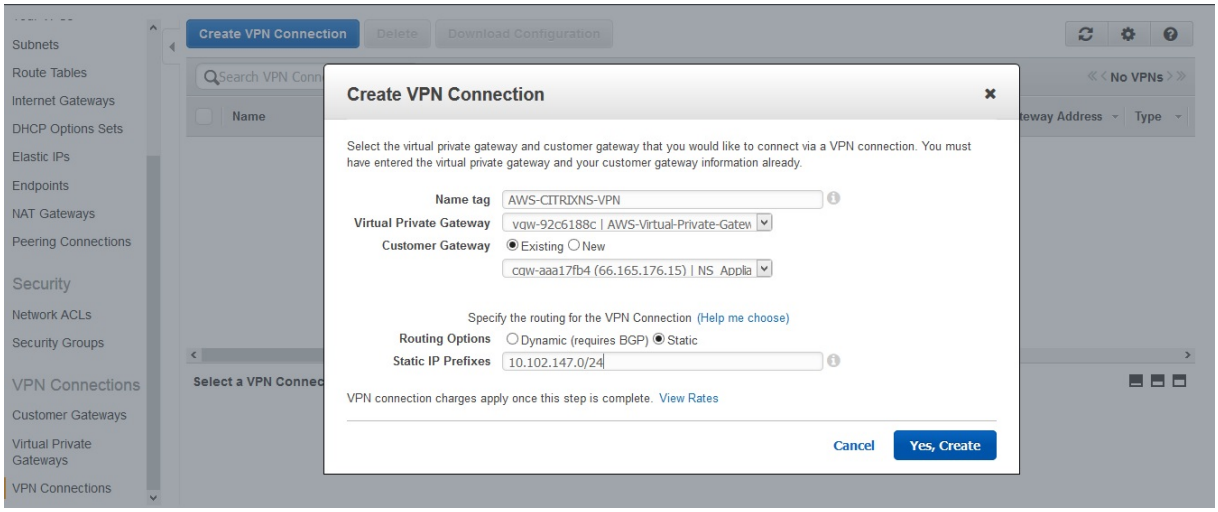


1. Select the virtual private gateway that you created, and then click Attach to VPC.
2. In the Attach to VPC dialog box, select your VPC from the list, and then choose Yes, Attach.



To create a VPN connection:

1. Navigate to VPN Connections > VPN Connections and then click Create VPN Connection.
2. In the Create VPN Connection dialog box set the following parameters and then choose Yes, Create:
   - **Name tag**. A name for the VPN connection.
   - **Virtual Private Gateway**. Select the virtual private gateway that you created earlier.
   - **Customer Gateway**. Select Existing. Then, from the drop down list, select the customer gateway that you created earlier.
   - **Routing Options**. Type of routing between the virtual private gateway and customer gateway (Citrix ADC appliance). Select Static. In the Static IP Prefixes field, specify the IP prefixes for the subnet on the Citrix ADC side, separated by commas.



To enable route propagation:

1. Navigate to **Route Tables** and select the routing table that's associated with the subnet whose traffic is to traverse the CloudBridge Connector tunnel.

> **Note**
>
> By default, this is the main routing table for the VPC.

1. On the **Route Propagation** tab in the details pane, choose **Edit**, select the virtual private gateway, and then choose **Save**.

To manually enter static routes:

1. Navigate to **Route Tables** and select your routing table.
2. On the **Routes** tab, click **Edit**.
3. In the **Destination** field, enter the static route used by your CloudBridge Connector tunnel (VPN connection).
4. Select the virtual private gateway ID from the **Target** list, and then click **Save**.

To download the configuration file:

1. Navigate to **VPN Connection**, select a VPN connection, and then click **Download Configuration**.
2. In the **Download Configuration** dialog box, set the following parameters, and then click **Yes, Download**.
   - **Vendor**. Select **Generic**.
   - **Platform**. Select **Generic**.
   - **Software**. Select **Vendor Agnostic**.

Configuring the Citrix ADC appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a virtual private gateway on AWS cloud, perform the following tasks on the Citrix ADC appliance.
You can use either the Citrix ADC command line or the GUI.

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm and PSK to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (a SNIP address configured on the Citrix ADC appliance), remote IP address (the public IP address of the virtual private gateway in AWS), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the Citrix ADC-side subnet whose traffic is to traverse the tunnel, and set the destination IP address range to specify the AWS VPC subnet whose traffic is to traverse the CloudBridge Connector tunnel. Any request packet that originates from a client in the subnet on the Citrix ADC side and is destined to a server in the AWS cloud subnet, and matches the source and destination IP range of the PBR entity, is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the Citrix ADC command line

At the Command prompt, type:

- `add ipsec profile <name> -psk <string> -**ikeVersion** v1`
- `show ipsec profile** <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line

At the Command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line

At the Command prompt, type:

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP** <subnet-range> -*ipTunnel <tunnelName>`
- `apply pbrs`

- `show pbr <pbrName>`

The following commands create all settings of Citrix ADC appliance NS_Appliance-1 used in "Example of CloudBridge Connector Configuration and Data Flow."

```
1      > add ipsec profile NS_AWS_IPSec_Profile -psk
          DkiMgMdcbqvYREEuIvxsbKkW0Foyabcd -ikeVersion v1  - lifetime
          31536000
2      Done
3      > add iptunnel NS_AWS_Tunnel 168.63.252.133 255.255.255.255
          66.165.176.15  - protocol IPSEC  - ipsecProfileName
          NS_AWS_IPSec_Profile
4
5      Done
6      > add pbr NS_AWS_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
          10.20.0.0-10.20.255.255  - ipTunnel NS_AWS_Tunnel
7      Done
8
9      > apply pbrs
10
11     Done
12 <!--NeedCopy-->
```

To create an IPSEC profile by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** dialog box, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version (select V1)
4. Select the **Pre-shared Key Authentication** method and set the **Pre-Shared Key Exists** parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click Add.
3. In the **Add IP Tunnel** dialog box, set the following parameters:
   - Name

- Remote IP
- Remote Mask
- Local IP Type (In the Local IP Type drop down list, select Subnet IP).
- Local IP (All the configured IPs of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
- Protocol
- IPSec Profile

4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** dialog box, set the following parameters:

   - Name
   - Action
   - Next Hop Type (Select IP Tunnel)
   - IP Tunnel Name
   - Source IP Low
   - Source IP High
   - Destination IP Low
   - Destination IP High

4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC appliance appears in the GUI.

The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters.
For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

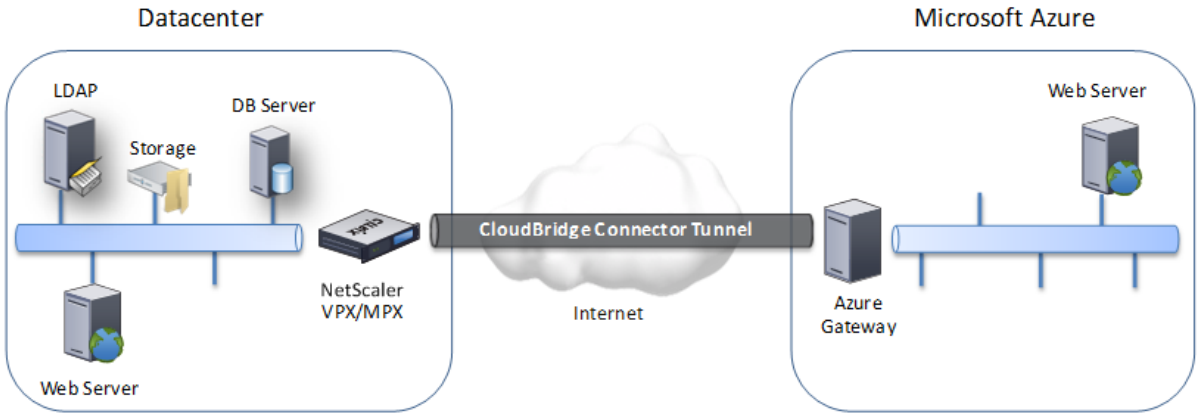
Configuring a CloudBridge Connector tunnel between a datacenter and Azure cloud

September 14, 2021

The Citrix ADC appliance provides connectivity between your enterprise datacenters and the Microsoft cloud hosting provider, Azure, making Azure a seamless extension of the enterprise network. Citrix ADC encrypts the connection between the enterprise datacenter and Azure cloud so that all data transferred between the two is secure.

How CloudBridge Connector tunnel works

To connect a datacenter to Azure cloud, you set up a CloudBridge Connector tunnel between a Citrix ADC appliance that resides in the datacenter and a gateway that resides in the Azure cloud. The Citrix ADC appliance in the datacenter and the gateway in Azure cloud are the end points of the CloudBridge Connector tunnel and are called peers of the CloudBridge Connector tunnel.



A CloudBridge Connector tunnel between a datacenter and Azure cloud uses the open-standard Internet Protocol security (IPSec) protocol suite, in tunnel mode, to secure communications between peers in the CloudBridge Connector tunnel. In a CloudBridge Connector tunnel, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

IPSec uses the tunnel mode in which the complete IP packet is encrypted and then encapsulated. The encryption uses the Encapsulating Security Payload (ESP) protocol, which ensures the integrity of the packet by using a HMAC hash function and ensures confidentiality by using an encryption algorithm. The ESP protocol, after encrypting the payload and calculating the HMAC, generates an ESP header

and inserts it before the encrypted IP packet. The ESP protocol also generates an ESP trailer and inserts it at the end of the packet.

The IPSec protocol then encapsulates the resulting packet by adding an IP header before the ESP header. In the IP header, the destination IP address is set to the IP address of the CloudBridge Connecter peer.

Peers in the CloudBridge Connector tunnel use the Internet Key Exchange version 1 (IKEv1) protocol (part of the IPSec protocol suite) to negotiate secure communication, as follows:

1. The two peers mutually authenticate with each other, using pre-shared key authentication, in which the peers exchange a text string called a pre-shared key (PSK). The pre-shared keys are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.
2. The peers then negotiate to reach agreement on:
   - An encryption algorithm
   - Cryptographic keys for encrypting data on one peer and decrypting it on the other.

   This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one-way (simplex). For example, when a CloudBridge Connector tunnel is set up between a Citrix ADC appliance in a datacenter and a gateway in an Azure cloud, both the datacenter appliance and the Azure gateway have two SAs. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets. SAs expire after a specified interval of time, which is called the lifetime.

Example of CloudBridge Connector tunnel configuration and data flow

As an illustration of CloudBridge Connector Tunnel, consider an example in which a CloudBridge Connector tunnel is set up between Citrix ADC appliance CB_Appliance-1 in a datacenter and gateway Azure_Gateway-1 in Azure cloud.

CB_Appliance-1 also functions as an L3 router, which enables a private network in the datacenter to reach a private network in the Azure cloud through the CloudBridge Connector tunnel. As a router, CB_Appliance-1 enables communication between client CL1 in the datacenter and server S1 in the Azure cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On CB_Appliance-1, the CloudBridge Connector tunnel configuration includes an IPSec profile entity named CB_Azure_IPSec_Profile, a CloudBridge Connector tunnel entity named CB_Azure_Tunnel, and a policy based routing (PBR) entity named CB_Azure_Pbr.

The IPSec profile entity CB_Azure_IPSec_Profile specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, and hash algorithm, to be used by the IPSec protocol in the CloudBridge Connector tunnel. CB_Azure_IPSec_Profile is bound to IP tunnel entity CB_Azure_Tunnel.

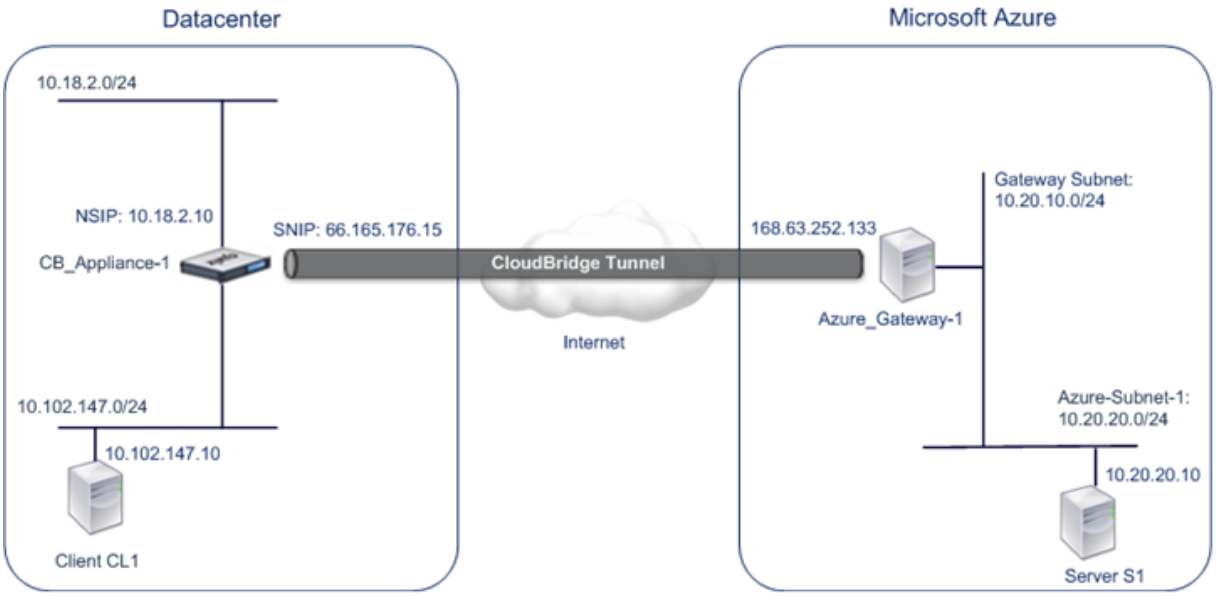
CloudBridge Connector tunnel entity CB_Azure_Tunnel specifies the local IP address (a public IP (SNIP) address configured on the Citrix ADC appliance), the remote IP address (the IP address of the Azure_Gateway-1), and the protocol (IPSec) used to set up the CloudBridge Connector tunnel. CB_Azure_Tunnel is bound to the PBR entity CB_Azure_Pbr.

The PBR entity CB_Azure_Pbr specifies a set of conditions and a CloudBridge Connector tunnel entity (CB_Azure_Tunnel). The source IP address range and the destination IP address range are the conditions for CB_Azure_Pbr. The source IP address range and the destination IP address range are specified as a subnet in the datacenter and a subnet in the Azure cloud, respectively. Any request packet originating from a client in the subnet in the datacenter and destined to a server in the subnet on the Azure cloud matches the conditions in CB_Azure_Pbr. This packet is then considered for CloudBridge processing and is sent across the CloudBridge Connector tunnel (CB_Azure_Tunnel) bound to the PBR entity.

On Microsoft Azure, the CloudBridge Connector tunnel configuration includes a local network entity named My-Datacenter-Network, a virtual network entity named Azure-Network-for-CloudBridge-Tunnel, and a gateway named Azure_Gateway-1.

The local (local to Azure) network entity My-Datacenter-Network specifies the IP address of the Citrix ADC appliance on the datacenter side, and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel. The virtual network entity Azure-Network-for-CloudBridge-Tunnel defines a private subnet named Azure-Subnet-1 in Azure. The traffic of the subnet traverses the CloudBridge Connector tunnel. The server S1 is provisioned in this subnet.

The local network entity My-Datacenter-Network is associated with the virtual network entity Azure-Network-for-CloudBridge-Tunnel. This association defines the remote and local network details of the CloudBridge Connector tunnel configuration in Azure. Gateway Azure_Gateway-1 was created for this association to become the CloudBridge end point at the Azure end of the CloudBridge Connector tunnel.



For more information about the settings, refer to the CloudBridge Connector Tunnel Settings pdf.

Points to consider for a CloudBridge Connector tunnel configuration

Before configuring a CloudBridge Connector tunnel between a Citrix ADC appliance in datacenter and Microsoft Azure, consider the following points:

1. The Citrix ADC appliance must have a public facing IPv4 address (type SNIP) to use as a tunnel end-point address for the CloudBridge Connector tunnel. Also, the Citrix ADC appliance should not be behind a NAT device.
2. Azure supports the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings while configuring the Citrix ADC for the CloudBridge Connector tunnel.
   - IKE version = v1
   - Encryption algorithm = AES
   - Hash algorithm = HMAC SHA1
3. You must configure the firewall in the datacenter edge to allow the following.
   - Any UDP packets for port 500
   - Any UDP packets for port 4500
   - Any ESP (IP protocol number 50) packets
4. IKE re-keying, which is renegotiation of new cryptographic keys between the CloudBridge Connector tunnel end points to establish new SAs, is not supported. When the Security Associations (SAs) expire, the tunnel goes into the DOWN state. Therefore, you must set a very large value for the lifetimes of SAs.
5. You must configure Microsoft Azure before specifying the tunnel configuration on the Citrix ADC, because the public IP address of the Azure end (gateway) of the tunnel, and the PSK, are auto-

matically generated when you set up the tunnel configuration in Azure. You need this information for specifying the tunnel configuration on the Citrix ADC.

Configuring the CloudBridge Connector tunnel

For setting up a CloudBridge Connector tunnel between your datacenter and Azure, you must install CloudBridge VPX/MPX in your datacenter, configure Microsoft Azure for the CloudBridge Connector tunnel, and then configure the Citrix ADC appliance in the data center for the CloudBridge Connector tunnel.

Configuring a CloudBridge Connector tunnel between a Citrix ADC appliance in datacenter and Microsoft Azure consists of the following tasks:

1. **Setting up the Citrix ADC appliance in the datacenter**. This task involves deploying and configuring a Citrix ADC physical appliance (MPX), or provisioning and configuring a Citrix ADC virtual appliance (VPX) on a virtualization platform in the datacenter.
2. **Configuring Microsoft Azure for the CloudBridge Connector tunnel**. This task involves creating local network, virtual network, and gateway entities in Azure. The local network entity specifies the IP address of the CloudBridge Connector tunnel end point (the Citrix ADC appliance) on the datacenter side, and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel. The virtual network defines a network on Azure. Creating the virtual network includes defining a subnet whose traffic is to traverse the CloudBridge Connector tunnel to be formed. You then associate the local network with the virtual network. Finally, you create a gateway that becomes the end point at the Azure end of the CloudBridge Connector tunnel.
3. **Configuring the Citrix ADC appliance in the datacenter for the CloudBridge Connector tunnel**. This task involves creating an IPSec profile, an IP tunnel entity, and a PBR entity in the Citrix ADC appliance in datacenter. The IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used in the CloudBridge Connector tunnel. The IP tunnel specifies the IP address of both the CloudBridge Connector tunnel end points (the Citrix ADC appliance in datacenter and the gateway in Azure) and the protocol to be used in the CloudBridge Connector tunnel. You then associate the IPSec profile entity with the IP tunnel entity. The PBR entity specifies the two subnets, in the datacenter and in the Azure cloud, that are to communicate with each other through the CloudBridge Connector tunnel. You then associate the IP tunnel entity with the PBR entity.

Configuring Microsoft Azure for the CloudBridge Connector tunnel

To create a CloudBridge Connector tunnel configuration on Microsoft Azure, use the Microsoft Windows Azure Management Portal, which is a web based graphical interface for creating and managing resources on Microsoft Azure.

Before you begin the CloudBridge Connector tunnel configuration on Azure cloud, make sure that:

- You have a user account for Microsoft Azure.
- You have a conceptual understanding of Microsoft Azure.
- You are familiar with the Microsoft Windows Azure Management Portal.

To configure a CloudBridge Connector tunnel between a datacenter and an Azure cloud, perform the following tasks on Microsoft Azure by using the Microsoft Windows Azure Management Portal:

- **Create a local network entity**. Create a local network entity in Windows Azure for specifying the network details of the datacenter. A local network entity specifies the IP address of the CloudBridge Connector tunnel end point (the Citrix ADC) on the datacenter side and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel.
- **Create a Virtual Network**. Create virtual network entity that defines a network on Azure. This task includes defining a private address space, where you provide a range of private addresses and subnets belonging to the range specified in the address space. The traffic of the subnets will traverse the CloudBridge Connector tunnel. You then associate a local network entity with the virtual network entity. This association lets Azure create a configuration for a CloudBridge Connector tunnel between the virtual network and the data center network. A gateway (to be created) in Azure for this virtual network will be the CloudBridge end point at the Azure end of the CloudBridge Connector tunnel. You then define a private subnet for the gateway to be created. This subnet belongs to the range specified in the address space in the virtual network entity.
- **Create a gateway in Windows Azure**. Create a gateway that becomes the end point at the Azure end of the CloudBridge Connector tunnel. Azure, from its pool of public IP addresses, assigns an IP address to the gateway created.
- **Gather the public IP address of the gateway and the pre-shared key.** For a CloudBridge Connector tunnel configuration on Azure, the public IP address of the gateway and the pre-shared Key (PSK) are automatically generated by Azure. Make a note of this information. You will need it for configuring the CloudBridge Connector tunnel on the Citrix ADC in datacenter.

> **Note:**
>
> The procedures for configuring Microsoft Azure for a CloudBridge Connector tunnel might change over time, depending on the Microsoft Azure release cycle. For the latest procedures, see the Microsoft Azure documentation.

Configuring the Citrix ADC Appliance in the datacenter for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a datacenter and an Azure cloud, perform the following tasks on the Citrix ADC in the datacenter. You can use either the Citrix ADC command line or the GUI:

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol

in the CloudBridge Connector tunnel.

- **Create an IP tunnel with IPSec protocol and associate the IPSec profile to it**. An IP tunnel specifies the local IP address (a public SNIP address configured on the Citrix ADC appliance), remote IP address (the public IP address of the gateway in Azure), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate the IP tunnel to it**. A PBR entity specifies a set of conditions and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range to specify the datacenter subnet whose traffic is to traverse the tunnel, and the destination IP address range to specify the Azure subnet whose traffic is to traverse the CloudBridge Connector tunnel. Any request packet originated from a client in the subnet on the datacenter and destined to a server in the subnet on the Azure cloud matches the source and destination IP range of the PBR entity. This packet is then considered for CloudBridge Connector tunnel processing and is sent across sent across the CloudBridge Connector tunnel associated with the PBR entity.

The GUI combines all these tasks in a single wizard called the CloudBridge Connector wizard.
To create an IPSEC profile by using the Citrix ADC command line:

At the Command prompt, type:

```
add ipsec profile <name> -psk <string> -ikeVersion v1
```

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line:

At the Command prompt, type:

```
add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -
ipsecProfileName <string>
```

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line

```
add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> ipTunnel
 <tunnelName> apply pbrs
```

Sample Configuration

The following commands create all settings of Citrix ADC appliance CB_Appliance-1 used in “Example of CloudBridge Connector Configuration and Data Flow”.

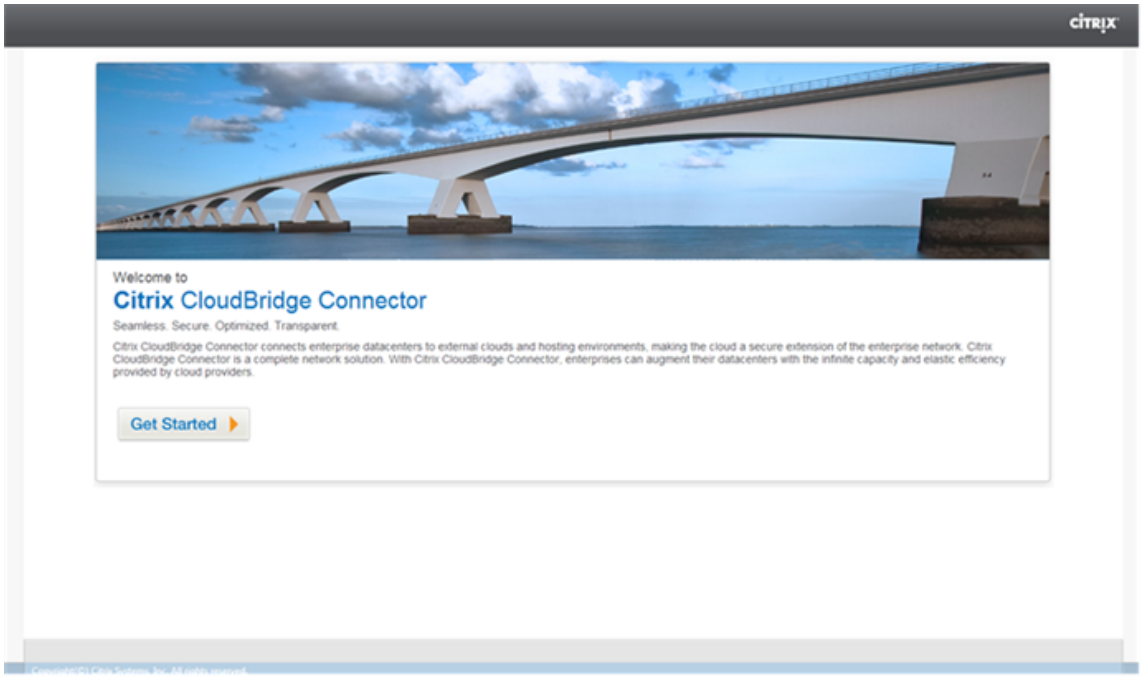
```
1 > add ipsec profile CB_Azure_IPSec_Profile -psk
     DkiMgMdcbqvYREEuIvxsbKkW0FOyDiLM  -ikeVersion v1  - lifetime 31536000
2 Done
3
4 >  add iptunnel CB_Azure_Tunnel 168.63.252.133 255.255.255.255
     66.165.176.15  - protocol IPSEC  - ipsecProfileName
     CB_Azure_IPSec_Profile
```



```
Done

> add pbr CB_Azure_Pbr -srcIP 10.102.147.0-10.102.147.255 -destIP
    10.20.0.0-10.20.255.255 -ipTunnelCB_Azure_Tunnel
Done

> apply pbrs
Done
<!--NeedCopy-->
```

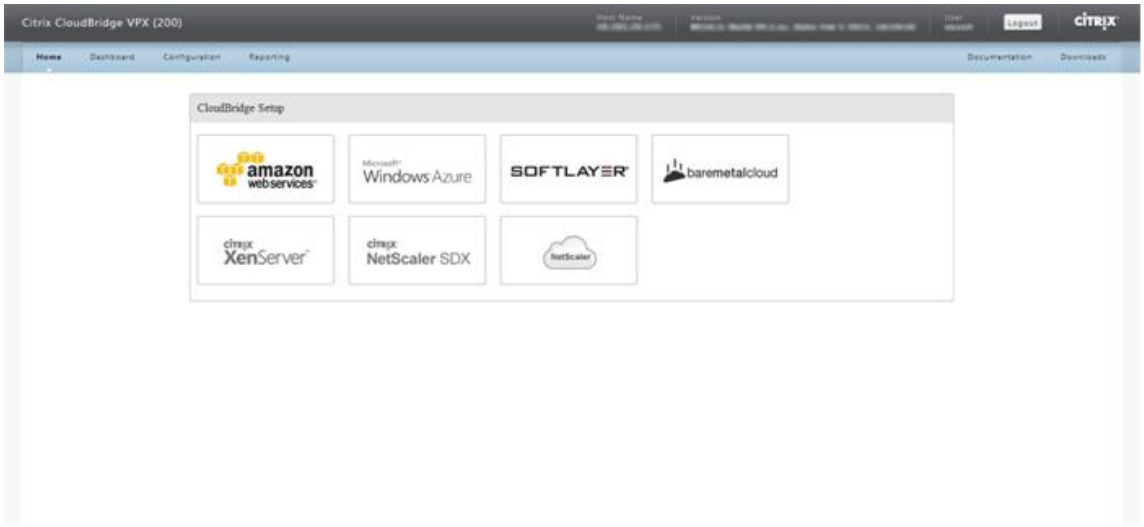
To configure a CloudBridge Connector tunnel in a Citrix ADC appliance by using the GUI

1. Access the GUI by using a web browser to connect to the IP address of the Citrix ADC appliance in the datacenter.
2. Navigate to **System** > **CloudBridge Connector**.
3. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.
4. Click **Get Started**.

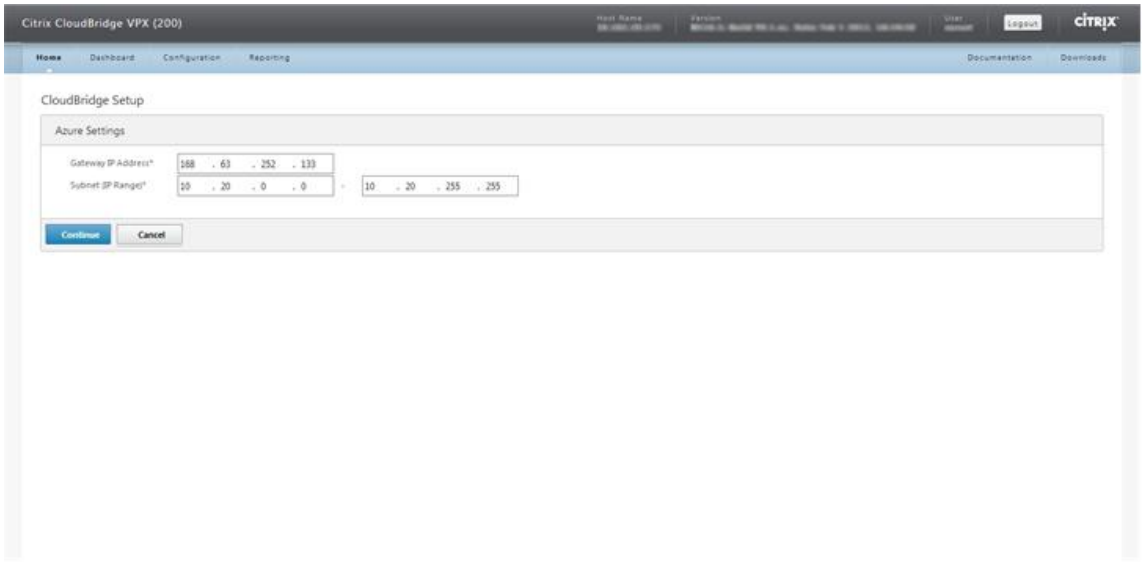


> **Note**: If you already have any CloudBridge Connector tunnel configured on the Citrix ADC appliance, this screen does not appear, and you are taken to the CloudBridge Connector Setup pane.

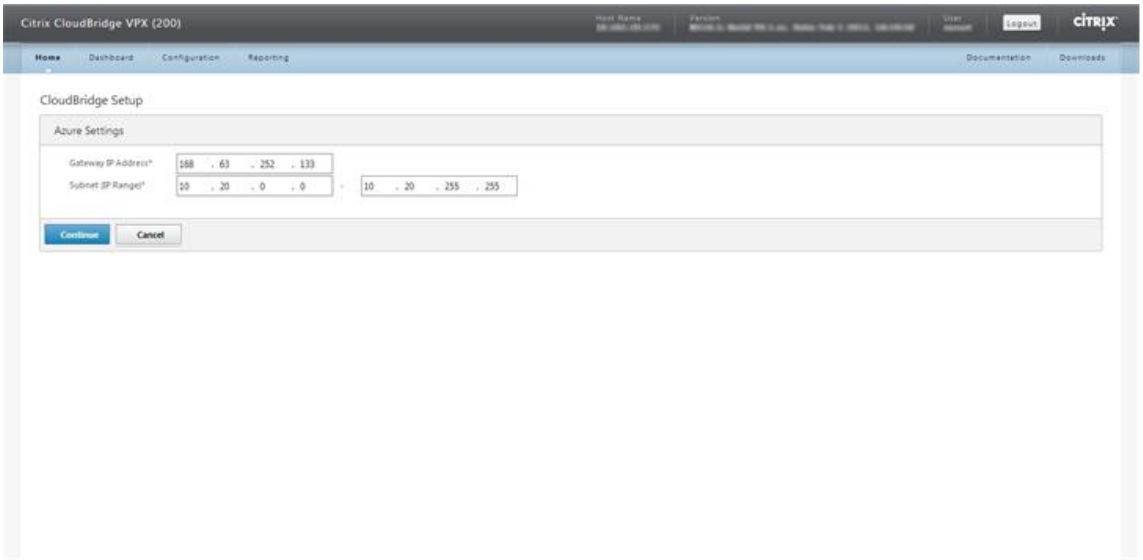
5. In the CloudBridge Setup pane, click **Microsoft Windows Azure**.



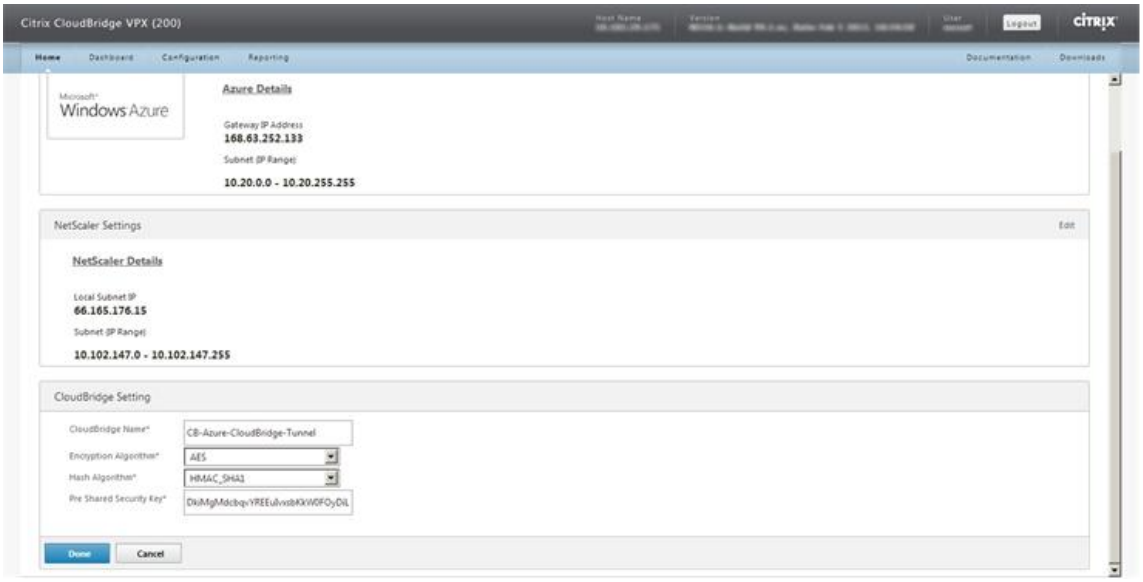
6. In the Azure Settings pane, in the **Gateway IP Address** field, type the IP address of the Azure gateway. The CloudBridge Connector tunnel is then set up between the Citrix ADC appliance and the gateway. In the **Subnet (IP Range)** text boxes, specify a subnet range (in Azure cloud), the traffic of which is to traverse the CloudBridge Connector tunnel. Click **Continue**.



7. In the Citrix ADC Settings pane, from the **Local Subnet IP** drop-down list, select a publicly accessible SNIP address configured on the Citrix ADC appliance. In **Subnet (IP Range)** text boxes, specify a local subnet range, the traffic of which is to traverse the CloudBridge Connector tunnel. Click **Continue**.



8. In the **CloudBridge Setting** pane, in the CloudBridge Name text box, type a name for the CloudBridge that you want to create.



9. From the Encryption Algorithm and Hash Algorithm drop-down lists, select the AES and HMAC_SHA1 algorithms, respectively. In the Pre Shared Security Key text box, type the security key.
10. Click **Done**.

Monitoring the CloudBridge Connector tunnel

You can view statistics for monitoring the performance of a CloudBridge Connector tunnel between the Citrix ADC appliance in the datacenter and Microsoft Azure. To view CloudBridge Connector tunnel statistics on the Citrix ADC appliance, use GUI or Citrix ADC command line. To view CloudBridge

Connector tunnel statistics in Microsoft Azure, use the Microsoft Windows Azure Management Portal.

Displaying CloudBridge Connector tunnel Statistics in the Citrix ADC appliance

For information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

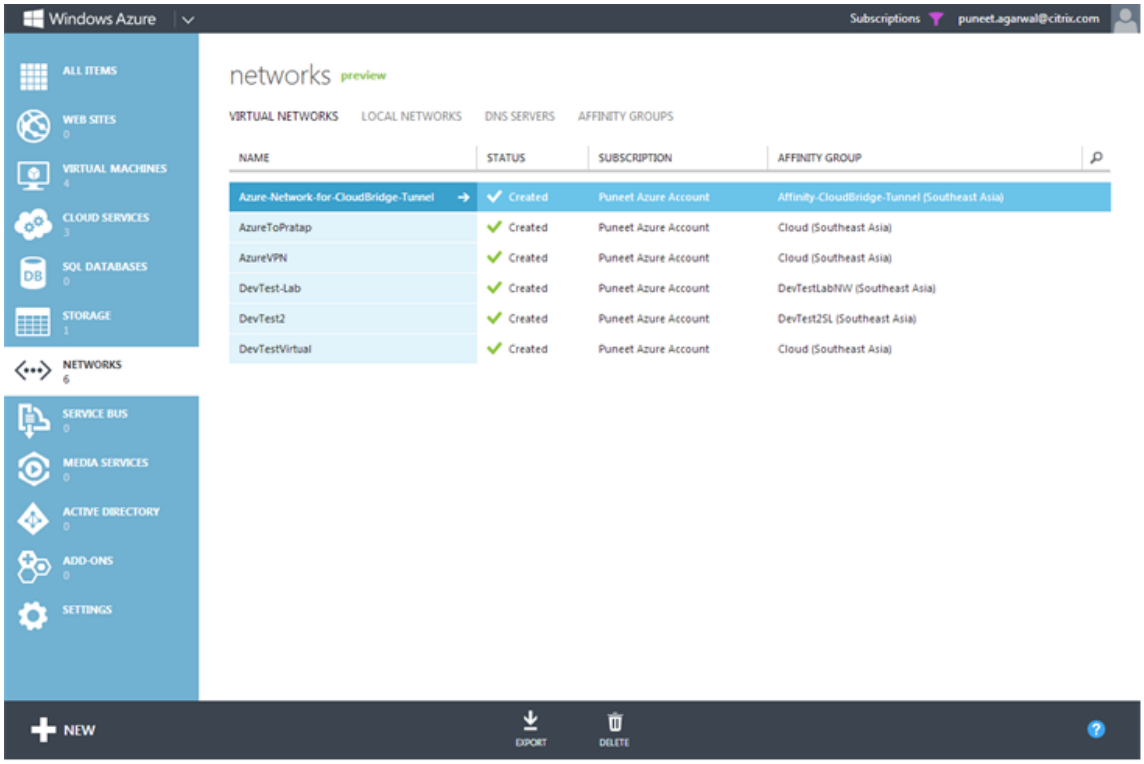
Displaying CloudBridge Connector tunnel Statistics in Microsoft Azure

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels in Microsoft Azure.

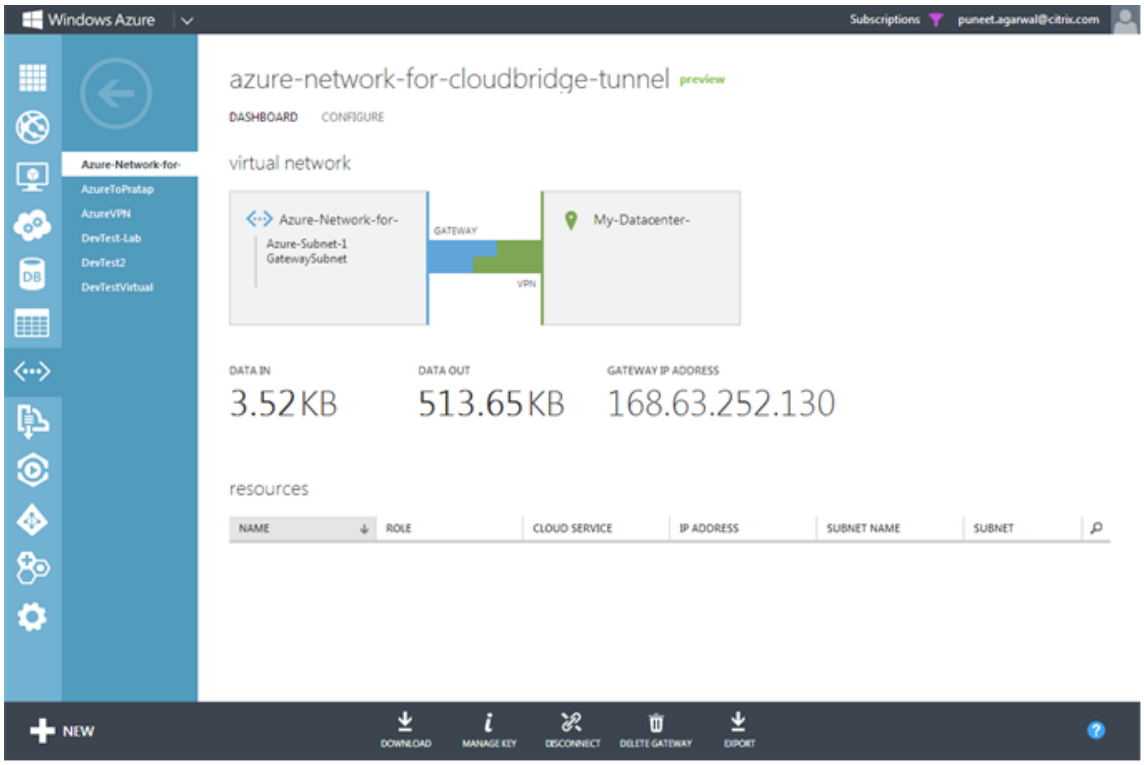
Statistical counter	Specifies
DATA IN	Total number of kilobytes received by the Azure gateway through the CloudBridge Connector tunnel since the gateway was created.
DATA OUT	Total number of kilobytes sent by the Azure gateway through the CloudBridge Connector tunnel since the gateway was created.

To display CloudBridge Connector tunnel statistics by using the Microsoft Windows Azure Management Portal

1. Log on to the Windows Azure Management Portal by using your Microsoft Azure account credentials.
2. In the left pane, click **NETWORKS**.
3. On the **Virtual Network** tab, in the Name column, select the virtual network entity associated with a CloudBridge Connector tunnel whose statistics you want to display.



4. On the **DASHBOARD** page of the virtual network, view the DATA IN and DATA OUT counters for the CloudBridge Connector tunnel.



Configuring CloudBridge Connector tunnel between datacenter and softlayer enterprise cloud

September 14, 2021

The GUI includes a wizard that helps you to easily configure a CloudBridge Connector tunnel between a Citrix ADC appliance in a datacenter and Citrix ADC VPX instances on the SoftLayer enterprise cloud.

When you use the wizard of the Citrix ADC appliance in the datacenter, the CloudBridge Connector tunnel configuration created on the Citrix ADC appliance, is automatically pushed to the other endpoint or peer (the Citrix ADC VPX on SoftLayer) of the CloudBridge Connector tunnel.

Using the wizard of the Citrix ADC appliance in the datacenter, you perform the following steps to configure a CloudBridge Connector tunnel.

1. Connect to the Softlayer enterprise cloud by providing the user log on credentials.
2. Select the Citrix XenServer that is running the Citrix ADC VPX appliance.
3. Select the Citrix ADC VPX appliance.
4. Provide CloudBridge Connector tunnel parameters to:
    - Configure a GRE Tunnel.
    - Configure IPsec on the GRE tunnel.
    - Create a netbridge, which is a logical representation of the CloudBridge connector, by specifying a name.
    - Bind the GRE Tunnel to the netbridge.

To configure a CloudBridge Connector tunnel by using the GUI

1. Log on to the GUI of the Citrix ADC appliance in the datacenter by using your account credentials for the appliance.
2. Navigate to **System** > **CloudBridge Connector** .
3. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge Connector**.
4. Click **Get Started**.

> **Note**:
>
> If you already have any CloudBridge Connector tunnel configured on the Citrix ADC appliance, this screen does not appear, and you are taken to the CloudBridge Connector Setup pane.

1. In the CloudBridge Connector Setup pane, click Softlayer, and then follow the instructions in the wizard.

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

Configuring a CloudBridge Connector tunnel between a Citrix ADC appliance and Cisco IOS device

September 14, 2021

You can configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco device to connect two datacenters or extend your network to a Cloud provider. The Citrix ADC appliance and the Cisco IOS device form the end points of the CloudBridge Connector tunnel and are called peers.

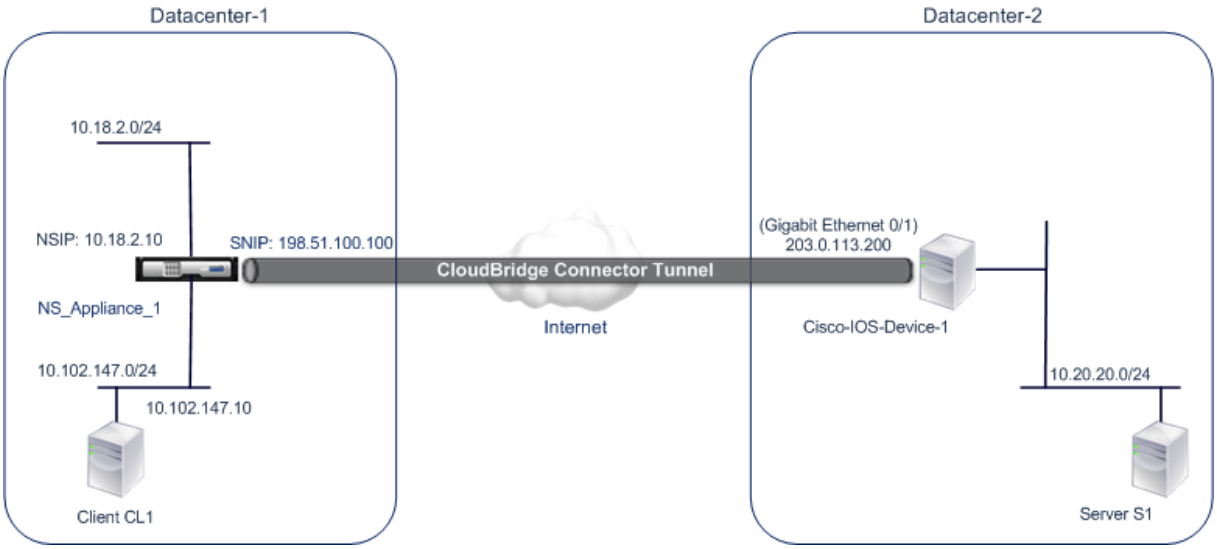
Example of CloudBridge Connector tunnel configuration and data flow

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- Citrix ADC appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- Cisco IOS device Cisco-IOS-Device-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and Cisco-IOS-Device-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and Cisco-IOS-Device-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_Cisco_IPSec_Profile, CloudBridge Connector tunnel entity NS_Cisco_Tunnel, and policy based routing (PBR) entity NS_Cisco_Pbr.



For more information, refer to the CloudBridge Connector tunnel between a Citrix ADC appliance and Cisco IOS device settings pdf.

Points to Consider for a CloudBridge Connector tunnel configuration

Before configuring a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco IOS device, consider the following points:

- The following IPSec settings are supported for a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco IOS device.

IPSec Properties	Setting
IPSec mode	Tunnel mode
IKE version	Version 1
IKE DH group	DH group 2 (1024 bits MODP algorithm)
IKE authentication method	Pre-Shared Key
IKE encryption algorithm	AES, 3DES
IKE hash algorithm	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5
ESP encryption algorithm	AES, 3DES
ESP hash algorithm	HMAC SHA1, HMAC SHA256, HMAC SHA256, HMAC SHA256, HMAC MD5

- You must specify the same IPSec settings on the Citrix ADC appliance and the Cisco IOS device

at the two ends of the CloudBridge Connector.

- Citrix ADC provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another, and a common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore on the Cisco device, you must specify the same hash algorithm and same encryption algorithm for IKE (while creating IKE policy) and ESP (while creating IPSec transform set).
- You must configure the firewall at the Citrix ADC end and Cisco device end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets

Configuring the Cisco IOS device for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel on a Cisco IOS device, use the Cisco IOS command line interface, which is the primary user interface for configuring, monitoring, and maintaining Cisco devices.

Before you begin the CloudBridge Connector tunnel configuration on a Cisco IOS device, make sure that:

- You have a user account with administrator credentials on the Cisco IOS device.
- You are familiar with the Cisco IOS command line interface.
- The Cisco IOS device is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.

> **Note**:
>
> The procedures for configuring CloudBridge Connector tunnel on a Cisco IOS device might change over time, depending on the Cisco release cycle. Citrix recommends that you follow the official Cisco product documention for more information, see Configuring IPSec VPN tunnels topic.

To configure a CloudBridge connector tunnel between a Citrix ADC appliance and a Cisco IOS device, perform the following tasks on the Cisco device's IOS command line:

- Create an IKE Policy.
- Configure a Pre-shared key for IKE authentication.
- Define a transform set and configure IPSec in tunnel mode.
- Create a crypto access List
- Create a crypto map
- Apply the crypto Map to an interface

The examples in the following procedures create settings in `Cisco IOS device Cisco-IOS-Device-1` mentioned in section "Example of CloudBridge Connector Configuration and Data Flow."

To create an IKE policy, refer to the IKE policy pdf.

To configure a pre-shared key by using the Cisco IOS command line:

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto isakmp identity address	Cisco-ios-device-1(config)# crypto isakmp identity address	Specify the ISAKMP identity (address) for the Cisco IOS device to use when communicating with the peer (Citrix ADC appliance) during IKE negotiations. This example specifies the address keyword, which uses IP address 203.0.113.200 (Gigabit Ethernet interface 0/1 of Cisco-IOS-Device-1) as the identity for the device.
crypto isakmp key keystringaddress peer-address	Cisco-ios-device-1 (config)# crypto isakmp key examplepresharedkey address 198.51.100.100	Specify a pre-shared key for the IKE authentication. This example configures shared key examplepresharedkey to be used with the Citrix ADC appliance NS_Appliance-1 (198.51.100.100). The same pre-shared key must be configured on the Citrix ADC appliance for IKE authentication to be successful between the Cisco IOS device and the Citrix ADC appliance.

To create a crypto access list by using the Cisco IOS command line:

At the Cisco IOS device's command prompt, type the following command in global configuration mode, in the order shown:

Command	Example	Command Description
access-listaccess-list-number permit IPsource source-wildcard destination destination-wildcard	Cisco-ios-device-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	Specify conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example configures access list 111 to protect traffic from subnets 10.20.20.0/24 (at the Cisco-IOS-Device-1 side) and 10.102.147.0/24 (at the NS_Appliance-1 side).

To define a transform and configure IPSec tunnel mode by using the Cisco IOS command line:

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:
Command	Example	Command Description
–	–	–
crypto ipsec transform-setname ESP_Authentication_Transform ESP_Encryption_Transform Note: ESP_Authentication_Transform can take the following values: esp-sha-hmac, esp-sha256-hmac, esp-sha384-hmac, esp-sha512-hmac, esp-md5-hmac. ESP_Encryption_Transform can take the following values: esp-aes or esp-3des	Cisco-ios-device-1(config)# crypto ipsec transform-set NS-CISCO-TS esp-sha256-hmac esp-3des	Define a transform set and specify the ESP hash algorithm (for authentication) and the ESP encryption algorithm to be used during exchange of data between the CloudBridge Connector tunnel peers. This example defines transform set NS-CISCO-TS and specifies ESP authentication algorithm as esp-sha256-hmac, and ESP encryption algorithm as esp-3des.
mode tunnel	Cisco-ios-device-1 (config-crypto-trans)# mode tunnel	Set IPSec in tunnel mode.
exit	Cisco-ios-device-1 (config-crypto-trans)# exit, Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

To create a crypto map by using the Cisco IOS command line:

At the Cisco IOS device's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto mapmap-name seq-num ipsec-isakmp	Cisco-ios-device-1 (config)# crypto map NS-CISCO-CM 2 ipsec-isakmp	Enter crypto map configuration mode, specify a sequence number for the crypto map, and configure the crypto map to use IKE to establish security associations (SAs). This example configures sequence number 2 and IKE for crypto map NS-CISCO-CM.
set peer ip-address	Cisco-ios-device-1 (config-crypto-map)# set peer 172.23.2.7	Specify the peer (Citrix ADC appliance) by its IP address. This example specifies 198.51.100.100, which is the CloudBridge Connector endpoint IP address on the Citrix ADC appliance.
match addressaccess-list-id	Cisco-ios-device-1 (config-crypto-map)# match address 111	Specify an extended access list. This access list specifies conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example specifies access list 111.
set transform-set transform-set-name	Cisco-ios-device-1 (config-crypto-map)# set transform-set NS-CISCO-TS	Specify which transform sets are allowed for this crypto map entry. This example specifies transform set NS-CISCO-TS.
exit	Cisco-ios-device-1 (config-crypto-map)# exit	
Cisco-ios-device-1 (config)#	Exit back to global configuration mode.	

To apply a crypto map to an interface by using the Cisco IOS command line:

At the Cisco IOS device's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
interfaceinterface-ID	Cisco-ios-device-1(config)# interface GigabitEthernet 0/1	Specify a physical interface to which to apply the crypto map and enter interface configuration mode. This example specifies Gigabit Ethernet interface 0/1 of the Cisco device Cisco-IOS-Device-1. IP address 203.0.113.200 is already set to this interface.
crypto mapmap-name	Cisco-ios-device-1 (config-if)# crypto map NS-CISCO-CM	Apply the crypto map to the physical interface. This example applies crypto map NS-CISCO-CM.
exit	Cisco-ios-device-1 (config-if)# exit, Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

Configuring the Citrix ADC appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco IOS device, perform the following tasks on the Citrix ADC appliance. You can use either the Citrix ADC command line or the Citrix ADC graphical user interface (GUI):

- Create an IPSec profile.
- Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it.
- Create a PBR rule and associate it with the IP tunnel.

To create an IPSEC profile by using the Citrix ADC command line:

At the Command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1`
- `show ipsec profile <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line:

At the Command prompt, type:

- add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>
- add ipTunnel <name>

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line:

At the Command prompt, type:

- add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>
- apply pbrs
- show pbrs <pbrName>

The following commands create settings in Citrix ADC appliance NS_Appliance-1 mentioned in section **Example of CloudBridge Connector Configuration and Data Flow**.

```
1      >  add ipsec profile NS_Cisco_IPSec_Profile -psk
           examplepresharedkey -ikeVersion v1  - lifetime 315360  - encAlgo 3
           DES
2      Done
3      >  add iptunnel NS_Cisco_Tunnel 203.0.113.200 255.255.255.255
           198.51.100.100  - protocol IPSEC  - ipsecProfileName
           NS_Cisco_IPSec_Profile
4
5      Done
6      > add pbr NS_Cisco_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
           10.20.0.0-10.20.255.255  - ipTunnel NS_Cisco_Tunnel
7
8      Done
9      > apply pbrs
10
11     Done
12 <!--NeedCopy-->
```

To create an IPSEC profile by using the GUI:

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** dialog box, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version
4. Configure the **IPSec authentication** method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the **Pre-shared key authentication** method and set

the **Pre-Shared Key Exists** parameter.

5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI:

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the **Add IP Tunnel** dialog box, set the following parameters:
   - Name
   - Remote IP
   - Remote Mask
   - Local IP Type (In the Local IP Type drop down list, select Subnet IP).
   - Local IP (All the configured IPs of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
   - Protocol
   - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** dialog box, set the following parameters:
   - Name
   - Action
   - Next Hop Type (Select IP Tunnel)
   - IP Tunnel Name
   - Source IP Low
   - Source IP High
   - Destination IP Low
   - Destination IP High
4. Click **Create**, and then click **Close**.

To apply a PBR by using the GUI:

1. Navigate to **System** > **Network** > **PBRs**.
2. On the **PBRs** tab, select the **PBR**, in the **Action list**, select **Apply**.

The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC appliance appears in the GUI. The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector Tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

Configuring a CloudBridge Connector tunnel between a Citrix ADC appliance and fortinet fortiGate appliance

September 14, 2021

You can configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a Fortinet FortiGate appliance to connect two datacenters or extend your network to a cloud provider. The Citrix ADC appliance and the FortiGate appliance form the end points of the CloudBridge Connector tunnel and are called peers.

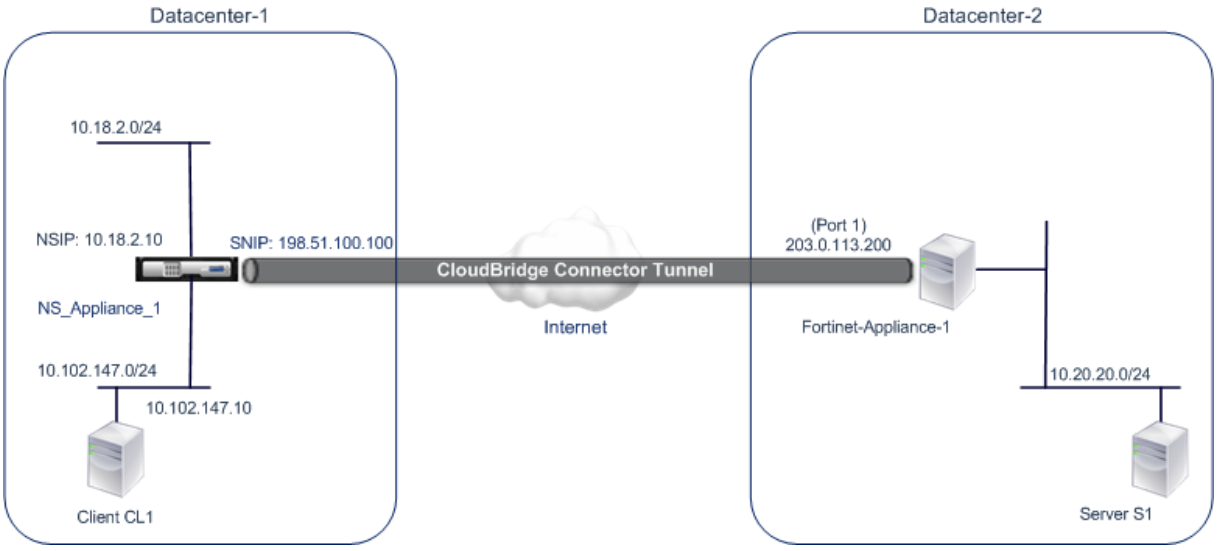
Example of a CloudBridge Connector Tunnel Configuration

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- Citrix ADC appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- FortiGate appliance FortiGate-Appliance-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and FortiGate-Appliance-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and FortiGate-Appliance-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_Fortinet_IPSec_Profile, CloudBridge Connector tunnel entity NS_Fortinet_Tunnel, and policy based routing (PBR) entity NS_Fortinet_Pbr.



For more information, see CloudBridge Connector tunnel configuration table pdf.

For information about Settings on Fortinet FortiGate-Appliance-1 in Datacenter-2, see table.

Points to consider for a CloudBridge Connector tunnel configuration

Before configuring a CloudBridge Connector tunnel between a Citrix ADC appliance and a FortiGate appliance, consider the following points:

- The following IPSec settings are supported for a CloudBridge Connector tunnel between a Citrix ADC appliance and a FortiGate appliance.

IPSec Properties	Settings
IPSec mode	Tunnel mode
IKE version	Version 1
IKE DH group	DH group 2 (1024 bits MODP algorithm)
IKE authentication method	Pre-Shared Key
IKE encryption algorithm	AES
IKE hash algorithm	HMAC SHA1
ESP encryption algorithm	AES
ESP hash algorithm	HMAC SHA1

- You must specify the same IPSec settings on the Citrix ADC appliance and the FortiGate appliance at the two ends of the CloudBridge Connector.

- Citrix ADC provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore in the FortiGate appliance, you must specify the same hash algorithm and same encryption algorithm in IKE (phase 1 configuration) and ESP (phase 2 configuration).
- You must configure the firewall at the Citrix ADC end and FortiGate end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets
- FortiGate appliance supports two types of VPN tunnels: Policy-based and Route-based. Only policy-based VPN tunnel is supported between a FortiGate appliance and a Citrix ADC appliance.

Configuring FortiGate appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel on a FortiGate appliance, use the Fortinet Web-based Manager, which is the primary user interface for configuring, monitoring, and maintaining FortiGate appliances.

Before you begin the CloudBridge Connector tunnel configuration on a FortiGate appliance, make sure that:

- You have a user account with administrator credentials on the FortiGate appliance.
- You are familiar with the Fortinet Web-based Manager.
- The FortiGate appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.

> **Note**
>
> The procedures for configuring CloudBridge Connector tunnel on a FortiGate appliance might change over time, depending on the Fortinet release cycle. Citrix recommends that you follow the official Fortinet product documentation for Configuring IPSec VPN tunnels.

To configure a CloudBridge connector tunnel between a Citrix ADC appliance and a FortiGate appliance, perform the following tasks on the FortiGate appliance by using the Fortinet Web-based manager:

- **Enable Policy-based IPSec VPN feature**. Enable this feature for creating policy-based VPN tunnels on the FortiGate appliance. Only policy-based type of VPN tunnel is supported between a FortiGate appliance and a Citrix ADC appliance. A policy-based VPN tunnel configuration on a FortiGate appliance includes phase 1 settings, phase 2 settings, and an IPSec security policy.
- **Define phase 1 parameters.** Phase 1 parameters are used by the FortiGate appliance for IKE Authentication before forming a secure tunnel to the Citrix ADC appliance.

- **Define phase 2 parameters**. Phase 2 parameters are used by the FortiGate appliance for forming a secure tunnel to the Citrix ADC appliance by establishing IKE security associations (SA).
- **Specify private subnets**. Define the FortiGate-side and the Citrix ADC-side private subnets whose IP traffic is to be transported through the tunnel.
- **Define an IPSec security policy for the tunnel**. A security policy allow IP traffic to pass between interfaces on a FortiGate appliance. An IPSec security policy specifies the interface to the private subnet and the interface connecting the Citrix ADC appliance through the tunnel.

To enable Policy-based IPSec VPN feature by using the Fortinet Web-based Manager

1. Navigate to **System** > **Config** > **Features**.
2. On the **Feature Settings** page, select **Show More** and turn on **Policy-based IPSec VPN**.

To define phase 1 parameters by using the Fortinet Web-based Manager

1. Navigate to **VPN** > **IPsec** > **Auto Key (IKE)** and click **Create Phase1**.
2. On the **New Phase 1** page, set the following parameters:
    - Name: Enter a name for this phase 1 configuration.
    - Remote Gateway: Select *Static IP Address.*
    - Mode: Select *Main (ID Protection)*.
    - Authentication Method: Select *Preshared Key*.
    - Pre-Shared Key: Enter a pre-shared key. The same pre-shared key must be configured on the Citrix ADC appliance.
    - Peer Options: Set the following IKE paramters for authenticating a Citrix ADC appliance.
        - IKE Version: Select *1*.
        - Mode Config: Clear this option if it is selected.
        - Local Gateway IP: Select *Main Interface IP*.
        - P1 Proposal: Select the encryption and authentication algorithms for IKE Authentication before forming a secure tunnel to the Citrix ADC appliance.
            * 1 - Encryption: Select *AES128*.
            * Authentication: Select *SHA1*.
            * Keylife: Enter an amount of time (in seconds) for the phase 1 key life.
            * DH Group: Select 2.
        - X-Auth: Select *Disable*.
        - Deed Peer Detection: Select this option.
3. Click **OK**.

To specify private subnets by using the Fortinet Web-based Manager

1. Navigate to **Firewall Objects** > **Address** > **Addresses** and select **Create New**.
2. On the **New Address** page, set the following parameters:
    - Name: Enter a name for FortiGate-side subnet.
    - Type: Select *Subnet*.
    - Subnet / IP Range: Enter the address of the FortiGate-side subnet.

- Interface: Select the local interface to this subnet.

3. Click **OK**.
4. Repeat steps 1-3 to specify the Citrix ADC-side subnet.

To define phase 2 parameters by using the Fortinet Web-based Manager

1. Navigate to **VPN** > **IPsec** > **Auto Key (IKE)** and click **Create Phase 2**.
2. On the **New Phase 2** page, set the following parameters:
   - Name: Enter a name for this phase 2 configuration.
   - Phase 1: Select the Phase 1 configuration from the drop-down list.
3. Click **Advanced** and set the following parameters:
   - P2 Proposal: Select the encryption and authentication algorithms for forming a secure tunnel to the Citrix ADC appliance.
     - 1 - Encryption: Select *AES128*.
     - Authentication: Select *SHA1*.
     - Enable replay detection: Select this option.
     - Enable perfect forward secrecy (PFS): Select this option.
     - DH Group: Select *2*.
   - Keylife: Enter an amount of time (in seconds) for the phase 2 key life.
   - Autokey Keep Alive: Select this option.
   - Auto-negotiate: Select this option.
   - Quick Mode Selector: Specify the FortiGate-side and the Citrix ADC-side private subnets whose traffic is to be traversed through the tunnel.
     - Source Address: Select the FortiGate-side subnet from the drop-down list.
     - Source Port: Enter *0*.
     - Destination Address: Select the Citrix ADC-side subnet from the drop-down list.
     - Destination Port: Enter *0*.
     - Protocol: Enter *0*.
4. Click **OK**.

To define an IPSec security policy by using the Fortinet Web-based Manager

1. Navigate to **Policy** > **Policy** > **Policy** and click **Create New**.
2. On the **Edit Policy** page, set the following parameters:
   - Policy Type: Select *VPN*.
   - Policy Subtype: Select *IPSec*.
   - Local Interface: Select the local interface to the internal (private) network.
   - Local Protected Subnet: Select the FortiGate-side subnet from the drop-down list whose traffic is to be traversed through the tunnel.
   - Outgoing VPN Interface: Select the local interface to the external (public) network.
   - Remote Protected Subnet: Select the Citrix ADC-side subnet from the drop-down list whose traffic is to be traversed through the tunnel.

- Schedule: Keep the default setting (*always*) unless changes are needed to meet specific requirements.
- Service: Keep the default setting (*ANY*) unless changes are needed to meet your specific requirements.
- VPN Tunnel: Select *Use Existing* and select the tunnel from the drop-down list.
- Allow traffic to be initiated from the remote site: Select if traffic from the remote network will be allowed to initiate the tunnel.

3. Click **OK**.

Configuring the Citrix ADC appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a FortiGate appliance, perform the following tasks on the Citrix ADC appliance. You can use either the Citrix ADC command line or the Citrix ADC graphical user interface (GUI):

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and authentication method to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (CloudBridge Connector tunnel end point IP address (of type SNIP) configured on the Citrix ADC appliance), remote IP address (CloudBridge Connector tunnel endpoint IP address configured on the FortiGate appliance), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the Citrix ADC-side subnet whose traffic is to be protected over the tunnel, and set the destination IP address range to specify the FortiGate appliance side subnet whose traffic is to be protected over the tunnel.

To create an IPSEC profile by using the Citrix ADC command line

At the command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line

At the command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName** <string>`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line

At the command prompt, type:

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

To create an IPSEC profile by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** page, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version
   - Perfect Forward Secrecy (Enable this paramter)
4. Configure the IPSec authentication method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the Pre-shared key authentication method and set the Pre-Shared Key Exists parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the **Add IP Tunnel** page, set the following parameters:
   - Name
   - Remote IP
   - Remote Mask
   - Local IP Type (In the Local IP Type drop-down list, select *Subnet IP*).
   - Local IP (All the configured IP addresses of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
   - Protocol
   - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.

2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** page, set the following parameters:
   - Name
   - Action
   - Next Hop Type (Select *IP Tunnel*)
   - IP Tunnel Name
   - Source IP Low
   - Source IP High
   - Destination IP Low
   - Destination IP High
4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC appliance appears in the GUI.

The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

The following commands create settings of Citrix ADC appliance NS_Appliance-1 in "Example of a CloudBridge Connector Configuration."

```
1      > add ipsec profile NS_Fortinet_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
          HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3       Done
4      > add iptunnel NS_Fortinet_Tunnel 203.0.113.200 255.255.255.255
          198.51.100.100 -protocol IPSEC -ipsecProfileName
          NS_Fortinet_IPSec_Profile
5
6       Done
7      > add pbr NS_Fortinet_Pbr -srcIP 10.102.147.0-10.102.147.255 -
          destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_Fortinet_Tunnel
8
9       Done
10     > apply pbrs
11
12      Done
13 <!--NeedCopy-->
```

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying Cloud-

Bridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

CloudBridge Connector tunnel diagnostics and troubleshooting

September 14, 2021

If you have problems with a CloudBridge Connector tunnel configuration, make sure that all prerequisites were observed before the tunnel was set up. If they were, the problem might be with the tunnel end-point IP addresses, a NAT configuration, the way the tunnel was set up, or with the data traffic.

Troubleshooting a CloudBridge Connector tunnel

If your CloudBridge Connector tunnel does not function properly, the issue could be with tunnel establishment or with the data traffic. If you are unsure which type of problem you have, look for an error message in the log file and see if the error message is in the list of tunnel-establishment issues. If you do not find your error message, check the list of possible issues related to data traffic.

Issues Related to tunnel establishment

After the requirements for configuring the IPSec tunnel are met and the CloudBridge Connector tunnel is configured, if the status of the tunnel is not UP, look for debugging information in the iked.log file on one or both Citrix ADC appliances configured as the tunnel end points.

On either appliance, type the following command at the Citrix ADC shell prompt:

```
cat /tmp/iked.debug | tee /var/iked.log
```

The Troubleshooting pdf lists some common errors and their solutions.

Issues related to data traffic

If the data in the CloudBridge Connector tunnel are not exchanged properly between the tunnel end points, do the following.

- For a CloudBridge Connector tunnel that uses GRE and IPSec protocols:
  - Make sure that L2 mode is enabled on both of the CloudBridge Connector tunnel end points. To enable L2 mode, type the following command at the Citrix ADC command line interface:

    ```
    enable mode L2
    ```


        * If one of the CloudBridge Connector tunnel end points is a CloudBridge virtual appliance (VPX) and is provisioned on a VMware ESXi hypervisor, make sure that Promiscuous mode is set to Accept for the vSwitch associated with the CloudBridge VPX appliance.
    - If a VLAN is extended through a CloudBridge Connector tunnel, verify one-to-one mapping on the extended VLAN entity on each of the tunnel end points
    - Make sure that the IP tunnel entity is bound to the correct netbridge entity in each of the tunnel end points.
    - Verify that the ARP entry for the peer CloudBridge Connector tunnel end point exists on the local tunnel end point, by typing the following command at the Citrix ADC command line interface:

      ```
      show arp
      ```
    - If the output shows an incomplete ARP entry, bidirectional traffic is not flowing through the tunnel. If bidirectional traffic is flowing, the ARP entry shows the name of tunnel interface for the devices on the other side of the tunnel.
    - Remove the IP tunnel entities from both tunnel end points and add them again with the same parameters, but with the IPSec profile set to NONE, so that the tunnel uses only the GRE protocol.

      After verifying the following in the IP tunnel (that uses GRE protocol), configure the tunnel with IPSec parameters by specifying a valid IPSec profile to the respective IP tunnel entities on each of the tunnel end points.

      Proper PING or TCP flow through the tunnel.

      Proper flow of data traffic through the tunnel.

      After the configured tunnel (that uses GRE and IPSec protocols) is in UP state, if the data traffic does not flow properly through the tunnel, and if a NAT device was deployed in front of any or both of the tunnel end points, analyze the ingress and egress packets on the NAT devices.
- If a Citrix ADC appliance is used as Router or Gateway.
    - Make sure that L3 mode is enabled on the Citrix ADC appliance. To enable L3 mode, run the following command in the CloudBridge command line.
    - enable mode L3
    - If subnets are bound to a netbridge entity, make sure that correct IP tunnel entity is also bound to the netbridge.
    - Run the following command in the Citrix ADC command line to see where the packets (Input and Output)are getting dropped:

      ```
      stat ipsec counters
      ```
    - Make sure that the correct routes are configured on both the tunnel end points.
    - If no NAT device is deployed in front of the Citrix ADC appliance, make sure that the firewalls are configured to allow any ESP (IP protocol number 50) packets and any UDP pack-

ets for port 4500.

If none of the above measures result in successful exchange of traffic between the tunnel end points, contact Citrix Technical Support.

Checklist before contacting Citrix technical Support

For a speedy resolution, make sure that you have the following items ready before contacting Citrix Technical Support.

- Details of the deployment and network topology.
- Log file collected by typing the following command at the Citrix ADC shell prompt.
  ```
  cat /tmp/iked.debug | tee /var/log/iked.log
  ```
- Tech support bundle captured by typing the following command at the Citrix ADC command line.
  ```
  show techsupport
  ```
- Packet traces captured on both CloudBridge Connector tunnel end points. To start a packet trace, type the following command at the Citrix ADC command line.
  ```
  start nstrace -size 0
  ```
  To stop packet trace, type the following command at the Citrix ADC command line.
  ```
  stop nstrace
  ```
- Output of the following command typed at the Citrix ADC command prompt.
  ```
  show arp
  ```

CloudBridge Connector interoperability – StrongSwan

September 14, 2021

StrongSwan is an opensource IPSec implementation for Linux platforms. You can configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a StrongSwan appliance to connect two datacenters or extend your network to a cloud provider. The Citrix ADC appliance and the StrongSwan appliance form the end points of the CloudBridge Connector tunnel and are called peers.

Example of a CloudBridge Connector tunnel configuration

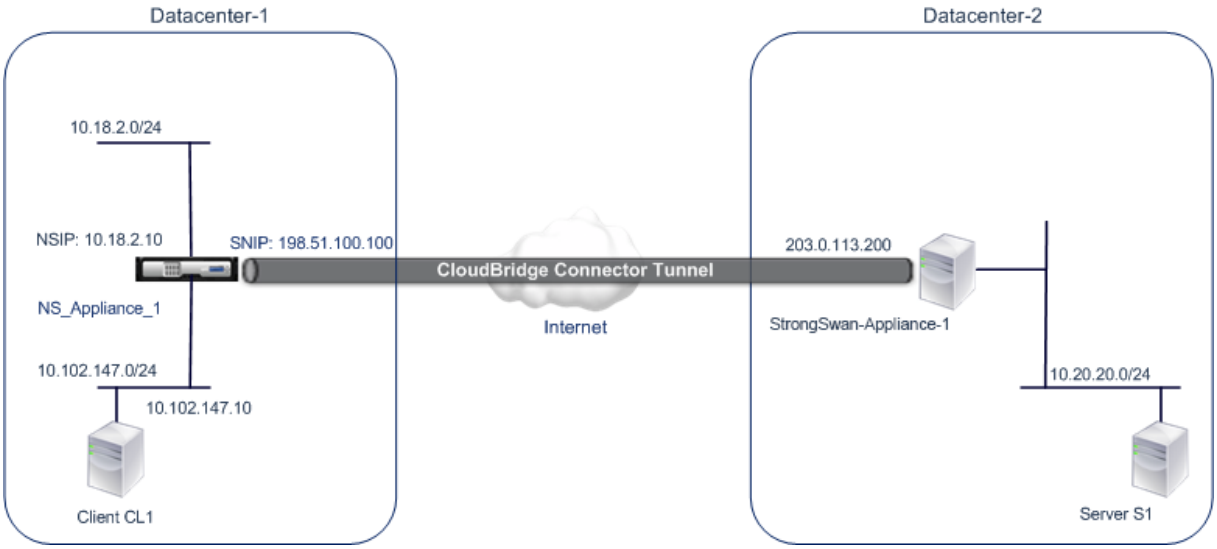
As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- Citrix ADC appliance NS_Appliance-1 in a datacenter designated as Datacenter-1

- StrongSwan appliance StrongSwan-Appliance-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and StrongSwan-Appliance-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and StrongSwan-Appliance-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_StrongSwan_IPSec_Profile, CloudBridge Connector tunnel entity NS_StrongSwan_Tunnel, and policy based routing (PBR) entity NS_StrongSwan_Pbr.



The following table lists the settings used in this example.

Main settings of the CloudBridge Connector tunnel setup

Entity	Details
IP address of the CloudBridge Connector tunnel end point (NS_Appliance-1) in Datacenter-1	198.51.100.100
IP address of the CloudBridge Connector tunnel end point (StrongSwan-Appliance-1) in Datacenter-2	203.0.113.200
Datacenter–1's subnet whose traffic is to be protected over the CloudBridge Connector tunnel	10.102.147.0/24

Entity	Details
Datacenter–2's subnet whose traffic is to be protected over the CloudBridge Connector tunnel	10.20.20.0/24

Settings on Citrix ADC appliance NS_Appliance-1 in Datacenter-1

SNIP1(for reference purposes only)	198.51.100.100	
IPSec profile	NS_StrongSwan_IPSec_Profile	IKE version: v1, Encryption algorithm: AES, Hash algorithm: HMAC_SHA1
psk = examplepresharedkey (Note: This is an example of a pre-share key, for illustration. Citrix does not recommend to use this string in your CloudBridge Connector configuration)		
CloudBridge Connector tunnel	NS_StrongSwan_Tunnel	Remote IP = 203.0.113.200, Local IP= 198.51.100.100, Tunnel protocol = IPSEC, IPSec profile= NS_StrongSwan_IPSec_Profile
Policy based route	NS_StrongSwan_Pbr	Source IP range = Subnet in the Datacenter-1=10.102.147.0-10.102.147.255, Destination IP range =Subnet in Datacenter-2=10.20.20.0-10.20.20.255, IP Tunnel = NS_StrongSwan_Tunnel

Points to consider for a CloudBridge Connector tunnel configuration

Before you begin configuring CloudBridge connector tunnel, make sure that:

- You have a basic knowledge about linux configurations.
- You have a basic Knowledge about IPSec protocol suite.
- The StrongSwan appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.
- The Citrix ADC appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.
- The following IPSec settings are supported for a CloudBridge Connector tunnel between a Citrix ADC appliance and a StrongSwan appliance.
  - IPSec mode: Tunnel mode
  - IKE version: Version 1
  - IKE authentication method: Pre-Shared Key
  - IKE encryption algorithm: AES
  - IKE hash algorithm: HMAC SHA1
  - ESP encryption algorithm: AES
  - ESP hash algorithm: HMAC SHA1
- You must specify the same IPSec settings on the Citrix ADC appliance and the StrongSwan appliance at the two ends of the CloudBridge Connector tunnel.
- Citrix ADC provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore, in the StrongSwan appliance, you must specify the same hash algorithm and same encryption algorithm in IKE and ESP parameters in the IPSec.conf file.
- You must configure the firewall at the Citrix ADC end and StrongSwan end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets

Configure StrongSwan for the CloudBridge Connector tunnel

To configure a CloudBridge connector tunnel between a Citrix ADC appliance and a StrongSwan appliance, perform the following tasks on the StrongSwan appliance:

- **Specify IPsec connection information in ipsec.conf file**. ipsec.conf file defines all control and configuration information for IPsec connections in the strongSwan appliance.
- **Specify pre-shared key in ipsec.secrets file**. ipsec.secrets file defines secrets for IKE/IPsec authentication for IPsec connections in the strongSwan appliance.

The procedures for configuring IPsec VPN (CloudBridge Connector tunnel) on a StrongSwan appliance might change over time, depending on the StrongSwan release cycle. Citrix recommends that you follow the official StrongSwan documentation for Configuring IPSec VPN tunnels.

Following sample excerpt of ipsec.conf file specifies IPsec information for setting up the IPsec VPN tunnel, described in Example of a CloudBridge Connector Configuration topic. For more information, see CloudBridge Connector Configuration pdf.

Following sample excerpt of ipsec.secrets file specifies the IKE authentication pre-shared key for setting up the IPsec VPN tunnel, described in Example of a CloudBridge Connector Configuration topic.

`/etc/ipsec.secrets` PSK 'examplepresharedkey' #pre-shared key for IPsec IKE authentication

Configuring the Citrix ADC appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a StrongSwan appliance, perform the following tasks on the Citrix ADC appliance. You can use either the Citrix ADC command line or the Citrix ADC graphical user interface (GUI):

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and authentication method to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (CloudBridge Connector tunnel endpoint IP address (of type SNIP) configured on the Citrix ADC appliance), remote IP address (CloudBridge Connector tunnel endpoint IP address configured on the StrongSwan appliance), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the Citrix ADC-side subnet whose traffic is to be protected over the tunnel, and set the destination IP address range to specify the StrongSwan side subnet whose traffic is to be protected over the tunnel.

To create an IPSEC profile by using the Citrix ADC command line

At the command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1`
- `show ipsec profile <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line

At the command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`

- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line

At the command prompt, type:

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

To create an IPSEC profile by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** page, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version
4. Configure the IPSec authentication method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the **Pre-shared key authentication method** and set the **Pre-Shared Key Exists** parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the Add IP Tunnel page, set the following parameters:
   - Name
   - Remote IP
   - Remote Mask
   - Local IP Type (In the Local IP Type drop-down list, select *Subnet IP*).
   - Local IP (All the configured IP addresses of the selected IP type are in the Local IP drop-down list. Select the desired IP from the list.)
   - Protocol
   - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** page, set the following parameters:

- Name
- Action
- Next Hop Type (Select *IP Tunnel*)
- IP Tunnel Name
- Source IP Low
- Source IP High
- Destination IP Low
- Destination IP High

4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC appliance appears in the GUI. The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

The following commands create settings of Citrix ADC appliance NS_Appliance-1 in “Example of a CloudBridge Connector Configuration:

```
1      > add ipsec profile NS_StrongSwan_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
          HMAC_SHA1
2
3
4      Done
5
6      > add iptunnel NS_StrongSwan_Tunnel 203.0.113.200 255.255.255.255
          198.51.100.100 -protocol IPSEC -ipsecProfileName
          NS_StrongSwan_IPSec_Profile
7
8
9      Done
10
11     > add pbr NS_StrongSwan_Pbr -srcIP 10.102.147.0-10.102.147.255 -
          destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_StrongSwan_Tunnel
12
13
14     Done
15
16     > apply pbrs
17
18
19     Done
20 <!--NeedCopy-->
```


Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

CloudBridge Connector interoperability – F5 BIG-IP

September 14, 2021

You can configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a F5 BIG-IP appliance to connect two datacenters or extend your network to a cloud provider. The Citrix ADC appliance and the F5 BIG-IP appliance form the end points of the CloudBridge Connector tunnel and are called peers.

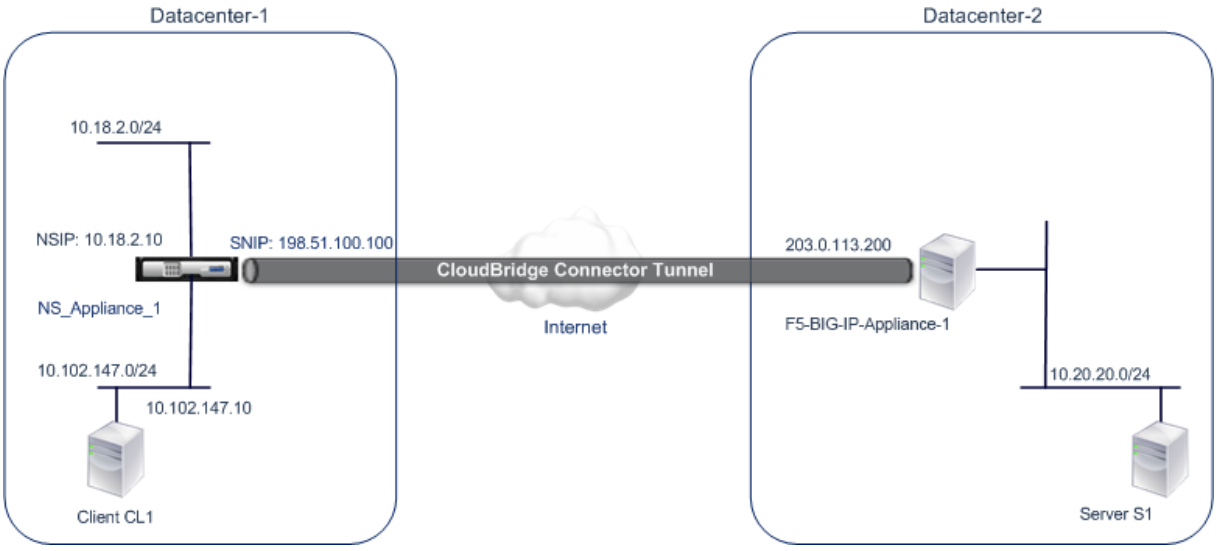
Example of a CloudBridge Connector tunnel configuration

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- Citrix ADC appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- F5 BIG-IP appliance F5-BIG-IP-Appliance-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and F5-BIG-IP-Appliance-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and F5-BIG-IP-Appliance-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_F5-BIG-IP_IPSec_Profile, CloudBridge Connector tunnel entity NS_F5-BIG-IP_Tunnel, and policy based routing (PBR) entity NS_F5-BIG-IP_Pbr.



For more information, refer to F5 big IP pdf.

Points to consider for a CloudBridge Connector tunnel configuration

- The Citrix ADC appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.
- The F5 BIG-IP appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.
- The following IPSec settings are supported for a CloudBridge Connector tunnel between a Citrix ADC appliance and an F5 BIG-IP appliance.
  - IPSec mode: Tunnel mode
  - IKE version: Version 1
  - IKE authentication method: Pre-Shared Key
  - IKE encryption algorithm: AES
  - IKE hash algorithm: HMAC SHA1
  - ESP encryption algorithm: AES
  - ESP hash algorithm: HMAC SHA1
- You must specify the same IPSec settings on the Citrix ADC appliance and the F5 BIG-IP appliance at the two ends of the CloudBridge Connector tunnel.
- Citrix ADC provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore, in the F5 BIG-IP appliance, you must specify the same hash algorithm and same encryption algorithm in IKE (phase 1 configuration) and ESP (phase 2 configuration).
- You must configure the firewall at the Citrix ADC end and F5 BIG-IP end to allow the following.
  - Any UDP packets for port 500

- Any UDP packets for port 4500
- Any ESP (IP protocol number 50) packets

Configuring F5 BIG-IP for the CloudBridge Connector tunnel

To configure a CloudBridge connector tunnel between a Citrix ADC appliance and an F5 BIG-IP appliance, perform the following tasks on the F5 BIG-IP appliance:

- **Create a forwarding virtual server for IPsec**. A forwarding virtual server intercepts IP traffic for the IPsec tunnel.
- **Create an IKE peer.** An IKE peer specifies the local and remote IPsec tunnel endpoints. It also specifies algorithms and credentials to be used for IPsec IKE phase 1.
- **Create a custom IPsec policy.** A policy specifies the IPSec protocol (ESP) and the mode (tunnel) to be used for forming the IPsec tunnel. It also specifies the algorithms and security parameters to be used for IKE IPsec phase 2.
- **Create a bidirectional IPsec traffic selector**. A traffic selector specifies the F5 BIG-IP side and Citrix ADC side subnets whose IP traffic is to be traversed through the IPsec tunnel.

The procedures for configuring IPsec VPN (CloudBridge Connector tunnel) on an F5 BIG-IP appliance might change over time, depending on the F5 release cycle. Citrix recommends that you follow the official F5 BIG-IP documentation for configuring IPSec VPN tunnels, at:

https://f5.com

To create a forwarding virtual server for IPsec by using the F5 BIG-IP GUI

1. On the **Main** tab, click **Local Traffic** > **Virtual Servers**, and then click **Create**.
2. On **New Virtual Server List** screen, set the following parameters:
   - **Name**. Type a unique name for the virtual server.
   - **Type**. Select **Forwarding (IP)**.
   - **Destination Address**. Type a wildcard network address in CIDR format, for example, 0.0.0.0/0 for IPv4 to accept any traffic.
   - **Service Port**. Select **All Ports** from the list.
   - **Protocol list**. Select **All Protocols** from the list.
   - **VLAN and Tunnel Traffic**. Retain the default selection, **All VLANs and Tunnels**.
3. Click **Finished**.

To create a custom IPsec policy by using the F5 BIG-IP GUI

1. On the **Main** tab, click **Network** > **IPsec** > **IPsec Policies**, and then click **Create**.
2. On the **New Policy** screen, set the following parameters:
   - **Name**. Type a unique name for the policy.
   - **IPsec Protocol**. Retain the default selection, ESP.
   - **Mode**. Select Tunnel. The screen refreshes to show additional related settings.

- **Tunnel Local Address**. Type the local IPsec tunnel endpoint IP address (Configured on the F5 BIG-IP appliance).
- **Tunnel Remote Address**. Type the remote IPsec tunnel endpoint IP address (Configured on the Citrix ADC appliance).

3. For the IKE Phase 2 parameters, retain the default values, or select the options that are appropriate for your deployment.
4. Click **Finished**.

To create a bidirectional IPsec traffic selector by using the F5 BIG-IP GUI

1. On the **Main** tab, click **Network** > **IPsec** > **Traffic Selectors**, and then click **Create**.
2. On the **New Traffic Selector** screen, set the following parameters:
   - **Name**. Type a unique name for the traffic selector.
   - **Order**. Retain the default value **(First)**. This setting specifies the order in which the traffic selector appears on the Traffic Selector List screen.
3. From the **Configuration** list, select **Advanced**, and set the following parameters:
   - **Source IP Address**. Click **Host** or **Network**, and in **Address** field, type the address of the F5 BIG-IP side subnet whose traffic is to be protected over the IPsec tunnel.
   - **Source Port**. Select *** All Ports**.
   - **Destination IP Address**. Click **Host**, and in the **Address** field, type the address of the Citrix ADC side subnet whose traffic is to be protected over the IPsec tunnel.
   - **Destination Port**. Select *** All Ports**.
   - **Protocol**. Select *** All Protocols**.
   - **Direction**. Select **Both**.
   - **Action**. Select **Protect**. The **IPsec Policy Name** setting appears.
   - **IPsec Policy Name**. Select the name of the custom IPsec policy that you created.
4. Click **Finished**.

Configuring the Citrix ADC appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a F5 BIG-IP appliance, perform the following tasks on the Citrix ADC appliance. You can use either the Citrix ADC command line or the Citrix ADC graphical user interface (GUI):

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and authentication method to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (CloudBridge Connector tunnel end point IP address (of type SNIP) configured on the Citrix ADC appliance), remote IP address (CloudBridge Connector tunnel endpoint IP address configured on the F5 BIG-IP appliance), protocol (IPSec) used to set

up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.

- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the Citrix ADC-side subnet whose traffic is to be protected over the tunnel, and set the destination IP address range to specify the F5 BIG-IP side subnet whose traffic is to be protected over the tunnel.

To create an IPSEC profile by using the Citrix ADC command line

At the command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecyENABLE`
- `show ipsec profile** <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line

At the command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line

At the command prompt, type:

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

To create an IPSEC profile by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IPSecProfile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** page, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version
4. Configure the IPSec authentication method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the **Pre-shared key authentication method** and set the **Pre-Shared Key Exists** parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the **Add IP Tunnel** page, set the following parameters:
   - Name
   - Remote IP
   - Remote Mask
   - Local IP Type (In the Local IP Type drop-down list, select *Subnet IP*).
   - Local IP (All the configured IP addresses of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
   - Protocol
   - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** page, set the following parameters:
   - Name
   - Action
   - Next Hop Type (Select *IP Tunnel*)
   - IP Tunnel Name
   - Source IP Low
   - Source IP High
   - Destination IP Low
   - Destination IP High
4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC appliance appears in the GUI. The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

The following commands create settings of Citrix ADC appliance NS_Appliance-1 in “Example of a CloudBridge Connector Configuration.:

```
1     > add ipsec profile NS_F5-BIG-IP_IPSec_Profile -psk
        examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
        HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3
4     Done
```



```
5
6       >  add iptunnel NS_F5-BIG-IP_Tunnel 203.0.113.200 255.255.255.255
            198.51.100.100  -protocol IPSEC  -ipsecProfileName NS_F5-BIG-
            IP_IPSec_Profile
7
8
9       Done
10
11      > add pbr NS_F5-BIG-IP_Pbr -srcIP 10.102.147.0-10.102.147.255  -
            destIP 10.20.0.0-10.20.255.255  -ipTunnel NS_F5-BIG-IP_Tunnel
12
13
14      Done
15
16      > apply pbrs
17
18
19      Done
20  <!--NeedCopy-->
```

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

CloudBridge Connector interoperability – Cisco ASA

September 14, 2021

You can configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco ASA appliance to connect two datacenters or extend your network to a cloud provider. The Citrix ADC appliance and the Cisco ASA appliance form the end points of the CloudBridge Connector tunnel and are called peers.

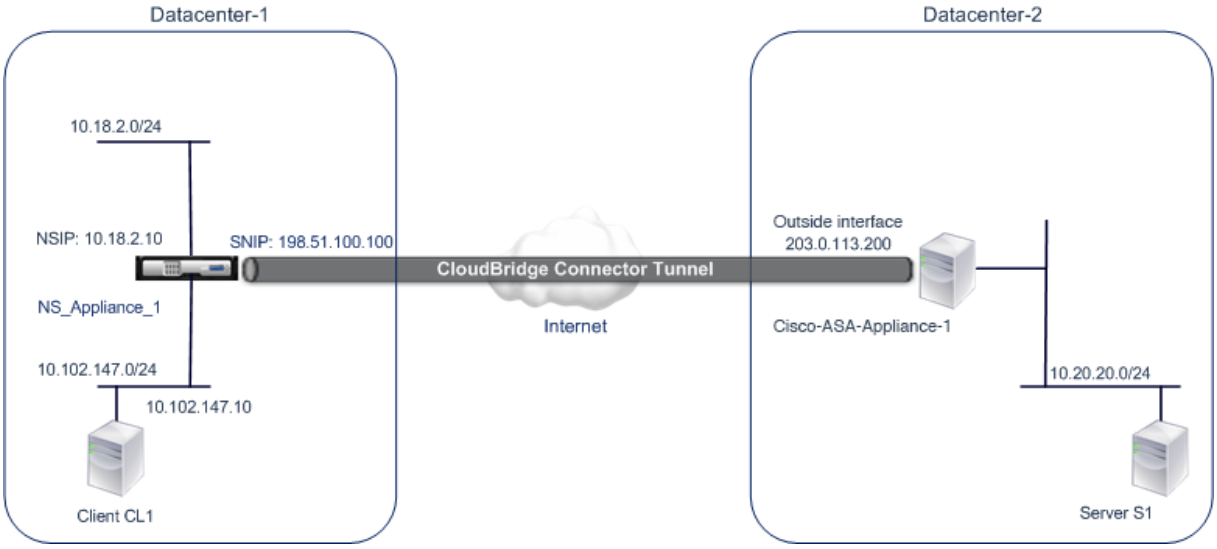
Example of a CloudBridge Connector tunnel configuration

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following appliances:

- Citrix ADC appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- Cisco ASA appliance Cisco-ASA-Appliance-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and Cisco-ASA-Appliance-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and Cisco-ASA-Appliance-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_Cisco-ASA_IPSec_Profile, CloudBridge Connector tunnel entity NS_Cisco-ASA_Tunnel, and policy based routing (PBR) entity NS_Cisco-ASA_Pbr.



Points to consider for a CloudBridge Connector tunnel configuration

Before you begin configuring CloudBridge connector tunnel, make sure that:

- The following IPSec settings are supported for a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco ASA appliance.

IPSec Properties	Settings
IPSec mode	Tunnel mode
IKE version	Version 1
IKE authentication method	Pre-Shared Key
IKE encryption algorithm	AES, 3DES
IKE hash algorithm	HMAC SHA1, HMAC MD5

IPSec Properties	Settings
ESP encryption algorithm	AES, 3DES
ESP hash algorithm	HMAC SHA1, HMAC MD5

- You must specify the same IPSec settings on the Citrix ADC appliance and the Cisco ASA appliance at the two ends of the CloudBridge Connector tunnel.
- Citrix ADC provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore, in the Cisco ASA appliance, you must specify the same hash algorithm and same encryption algorithm in IKE (phase 1 configuration) and ESP (phase 2 configuration).
- You must configure the firewall at the Citrix ADC end and Cisco ASA end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets

Configuring Cisco ASA for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel on a Cisco ASA appliance, use the Cisco ASA command line interface, which is the primary user interface for configuring, monitoring, and maintaining Cisco ASA appliances.

Before you begin the CloudBridge Connector tunnel configuration on a Cisco ASA appliance, make sure that:

- You have a user account with administrator credentials on the Cisco ASA appliance.
- You are familiar with the Cisco ASA command line interface.
- The Cisco ASA appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.

> **Note**
>
> The procedures for configuring CloudBridge Connector tunnel on a Cisco ASA appliance might change over time, depending on the Cisco release cycle. Citrix recommends that you follow the official Cisco ASA product documentation for Configuring IPSec VPN tunnels, at:

- http://www.cisco.com

To configure a CloudBridge connector tunnel between a Citrix ADC appliance and a Cisco ASA appliance, perform the following tasks on the Cisco ASA appliance's command line:

- **Create an IKE Policy**. An IKE policy defines a combination of security parameters to be used

during the IKE negotiation (phase 1). For example, parameters such as hash algorithm, encryption algorithm, and authentication method to be used in the IKE negotiation are set in this task.

- **Enable IKE on the outside interface**. Enable IKE on the outside interface through which the tunnel traffic will flow to the tunnel peer.
- **Create a tunnel group**. A tunnel group specifies the type of tunnel and the pre-shared key. The tunnel type must be set to ipsec-l2l, which stands for IPsec LAN to LAN. A pre-shared key is a text string, which the peers of a CloudBridge Connector tunnel use to mutually authenticate with each other. The pre-shared keys are matched against each other for IKE authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on the Cisco ASA appliance and the Citrix ADC appliance.
- **Define a transform set**. A transform set defines a combination of security parameters (phase 2) to be used in the exchange of data over the CloudBridge Connector tunnel after the IKE negotiation is successful.
- **Create an access List**. Crypto access lists are used to define the subnets whose IP traffic will be protected over the CloudBridge tunnel. The source and destination parameters in the access list specify the Cisco appliance side and Citrix ADC side subnets that are to be protected over the CloudBridge Connector Tunnel. The access list must be set to permit. Any request packet that originates from an appliance in the Cisco appliance side subnet and is destined to an appliance in the Citrix ADC side subnet, and that matches the source and destination parameters of the access list, is sent across the CloudBridge Connector tunnel.
- **Create a crypto map**. Crypto maps define the IPSec parameters for security associations (SAs). They include the following: Crypto access list to identify the subnets whose traffic is to be protected over the CloudBridge tunnel, peer (Citrix ADC) identification by IP address, and transform set to match the peer security settings.
- **Apply the crypto Map to the outside interface**. In this task, you apply the crypto map to the outside interface through which the tunnel traffic will flow to the tunnel peer. Applying the crypto map to an interface instructs the Cisco ASA appliance to evaluate all interface traffic against the crypto map set and to use the specified policy during connection or security association negotiations.

The examples in the following procedures create settings of Cisco ASA appliance Cisco-ASA-Appliance-1 used in Example of CloudBridge Connector Configuration and Data Flow.

To create an IKE policy by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following commands, starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto ikev1 policy priority	Cisco-ASA-appliance-1(config)# crypto ikev1 policy 1	Enter IKE policy configuration mode and identify the policy to create. (Each policy is uniquely identified by the priority number you assign.) This example configures policy 1.
encryption (3des | aes)	Cisco-ASA-appliance-1 (config-ikev1-policy)# encryption 3des	Specify the encryption algorithm. This example configures the 3DES algorithm.
hash (sha | md5)	Cisco-ASA-appliance-1 (config- ikev1-policy)# hash sha	Specify the hash algorithm. This example configures SHA.
authenticationpre-share	Cisco-ASA-appliance-1 (config- ikev1-policy)# authentication pre-share	Specify the pre-share authentication method.
group 2	Cisco-ASA-appliance-1 (config- ikev1-policy)# group 2	Specify 1024-bit Diffie-Hellman group identifier (2).
lifetime seconds	Cisco-ASA-appliance-1 (config- ikev1-policy)# lifetime 28800	Specify the security association's lifetime in seconds. This example configures 28800 seconds, which is the default value of lifetime in a Citrix ADC appliance.

To enable IKE on the outside interface by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following commands, starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto ikev1 enable outside	Cisco-ASA-appliance-1(config)# crypto ikev1 enable outside	Enable IKEv1 on the interface through which the tunnel traffic flows to the tunnel peer. This example enables IKEv1 on the interface named outside.

To create a tunnel group by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following commands, starting in global configuration mode, as show in the attached pdf Tunnel Group using Cisco ASA command line:

To create a crypto access list by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following command in global configuration mode, in the order shown:

Command	Example	Command Description
access-list access-list-number permit IP source source-wildcard destination destination-wildcard	Cisco-ASA-appliance-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	Specify conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example configures access list 111 to protect traffic from subnets 10.20.20.0/24 (at the Cisco-ASA-Appliance-1 side) and 10.102.147.0/24 (at the NS_Appliance-1 side).

To define a transform set by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following commands, starting in global configuration mode. See Transform set using ASA command line table pdf.

To create a crypto map by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto map map-name seq-num match address access-list-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 match address 111	Create a crypto map and specify an access list to it. This example configures crypto map NS-CISCO-CM with sequence number 1 and assigns access list 111 to NS-CISCO-CM.
crypto map map-name seq-num set peer ip-address	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set peer 198.51.100.100	Specify the peer (Citrix ADC appliance) by its IP address. This example specifies 198.51.100.100, which is the tunnel endpoint IP address on the Citrix ADC appliance.
crypto map map-name seq-num set ikev1 transform-set transform-set-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set ikev1 transform-set NS-CISCO-TS	Specify which transform set is allowed for this crypto map entry. This example specifies transform set NS-CISCO-TS.

To apply a crypto map to an interface by using the Cisco ASA command line

At the Cisco ASA appliance's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto map map-nameinterface interface-name	Cisco-ASA-appliance-1(config)# crypto map NS-CISCO-CM interface outside	Apply the crypto map to the interface through which CloudBridge Connector tunnel traffic will flow. This example applies crypto map NS-CISCO-CM to interface outside.

Configuring the Citrix ADC appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a Citrix ADC appliance and a Cisco ASA appliance, perform the following tasks on the Citrix ADC appliance. You can use either the Citrix ADC com-

mand line or the Citrix ADC graphical user interface (GUI):

- Create an IPSec profile.
- Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it.
- Create a PBR rule and associate it with the IP tunnel.

To create an IPSEC profile by using the Citrix ADC command line:

At the command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the Citrix ADC command line:

At the command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the Citrix ADC command line:

At the command prompt, type:

- `**add pbr** <pbrName> **ALLOW** -**srcIP** <subnet-range> -**destIP** <subnet-range>`
- `**ipTunnel** <tunnelName>`
- `**apply pbrs**`
- `**show pbr** <pbrName>`

To create an IPSEC profile by using the GUI:

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** page, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version
   - Perfect Forward Secrecy (Enable this parameter)
4. Configure the IPSec authentication method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the **Pre-shared key authentication** method and set the **Pre-Shared Key Exists** parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI:

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels tab**, click **Add**.
3. In the **Add IP Tunnel** page, set the following parameters:
   - Name
   - Remote IP
   - Remote Mask
   - Local IP Type (In the Local IP Type drop-down list, select Subnet IP).
   - Local IP (All the configured IP addresses of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
   - Protocol
   - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI:

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** page, set the following parameters:
   - Name
   - Action
   - Next Hop Type (Select IP Tunnel)
   - IP Tunnel Name
   - Source IP Low
   - Source IP High
   - Destination IP Low
   - Destination IP High
4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the Citrix ADC appliance appears in the GUI. The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

The following commands create settings of Citrix ADC appliance NS_Appliance-1 in “Example of a CloudBridge Connector Configuration.”:

```
1 >  add ipsec profile NS_Cisco-ASA_IPSec_Profile -psk
     examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
     HMAC_SHA1  - lifetime 315360 -perfectForwardSecrecy ENABLE
2
3 Done
4
5 >  add iptunnel NS_Cisco-ASA_Tunnel 203.0.113.200 255.255.255.255
     198.51.100.100  - protocol IPSEC  - ipsecProfileName NS_Cisco-
```



```
    ASA_IPSec_Profile
6
7
8 Done
9
10 > add pbr NS_Cisco-ASA_Pbr -srcIP 10.102.147.0-10.102.147.255 -destIP
    10.20.0.0-10.20.255.255 -ipTunnel NS_Cisco-ASA_Tunnel
11
12
13 Done
14
15 > apply pbrs
16
17 Done
18
19 <!--NeedCopy-->
```

Monitoring the CloudBridge Connector Tunnel

You can monitor the performance of CloudBridge Connector tunnels on a Citrix ADC appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a Citrix ADC appliance, see Monitoring CloudBridge Connector Tunnels.

High Availability

September 14, 2021

A high availability (HA) deployment of two Citrix ADC appliances can provide uninterrupted operation in any transaction. With one appliance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If, for any reason, the primary node is unable to accept connections, the secondary node takes over.

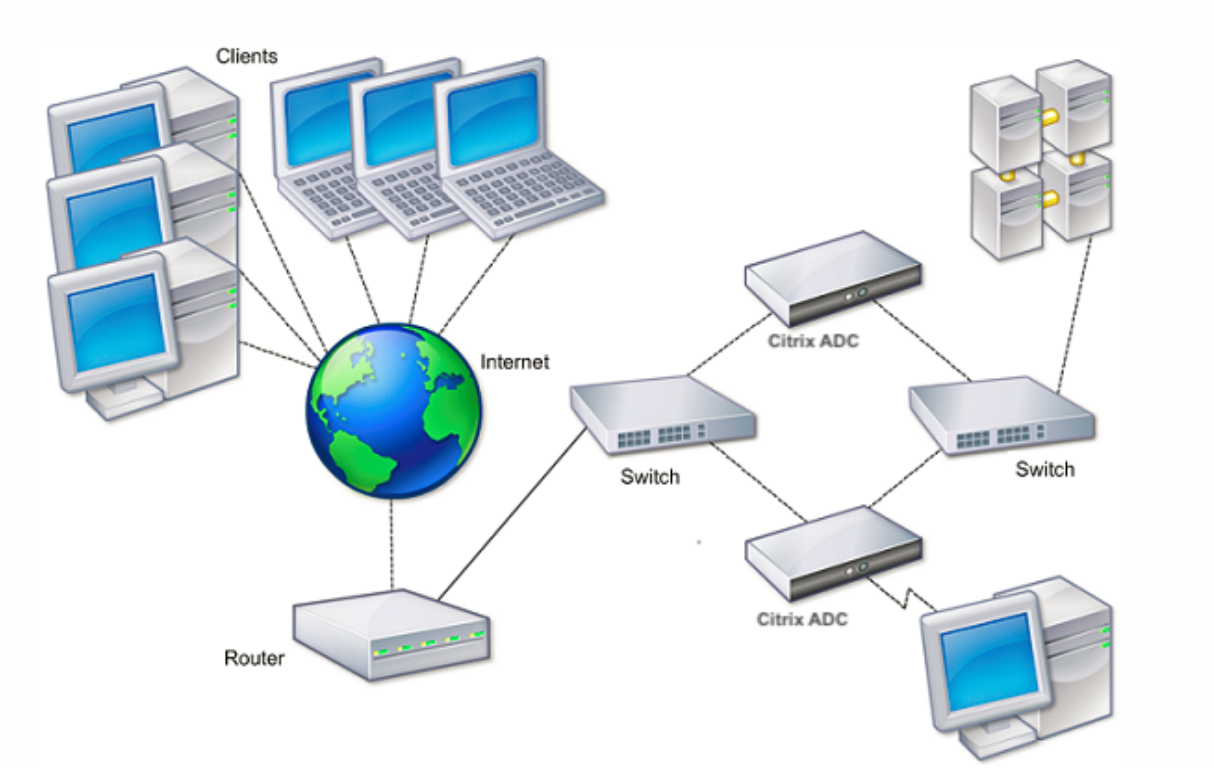
The secondary node monitors the primary by sending periodic messages (often called heartbeat messages or health checks) to determine whether the primary node is accepting connections. If a health check fails, the secondary node retries the connection for a specified period, after which it determines that the primary node is not functioning normally. The secondary node then takes over for the primary (a process called failover).

After a failover, all clients must reestablish their connections to the managed servers, but the session persistence rules are maintained as they were before the failover.

With Web server logging persistence enabled, no log data is lost due to the failover. For logging persistence to be enabled, the log server configuration must carry entries for both systems in the log.conf file.

The following figure shows a network configuration with an HA pair.

Figure 1. Citrix ADC Appliances in a High Availability Configuration



To configure HA, you might want to begin by creating a basic setup, with both nodes in the same subnet. You can then customize the intervals at which the nodes communicate health-check information, the process by which nodes maintain synchronization, and the propagation of commands from the primary to the secondary. You can configure fail-safe mode to prevent a situation in which neither node is primary. If your environment includes devices that do not accept Citrix ADC gratuitous ARP messages, you should configure virtual MAC addresses. When you are ready for a more complex configuration, you can configure HA nodes in different subnets.

To improve the reliability of your HA setup, you can configure route monitors and create redundant links. In some situations, such as when troubleshooting or performing maintenance tasks, you might want to force a node to fail over (assign primary status to the other node), or you might want to force the secondary node to stay secondary or the primary node to stay primary.

Points to consider for a high availability setup

September 14, 2021

> **Note**
>
> The following requirements for configuring systems in an HA setup:

- In an HA configuration, the primary and secondary Citrix ADC appliances should be of the same model. Different Citrix ADC models are not supported in an HA pair.
- In an HA setup, both nodes must run the same version of Citrix ADC.
- Entries in the configuration file (ns.conf) on both the primary and the secondary system must match, with the following exceptions:
  - The primary and the secondary systems must each be configured with their own unique IP addresses (NSIPs.)
  - In an HA pair, the node ID and associated IP address of one node must point to the other node. For example, if you have nodes NS1 and NS2, you must configure NS1 with a unique node ID and the IP address of NS2, and you must configure NS2 with a unique node ID and the IP address of NS1.
- If you create a configuration file on either node by using a method that does not go directly through the GUI or the CLI (for example, importing SSL certificates, or changing to startup scripts), you must copy the configuration file to the other node or create an identical file on that node.
- Initially, all Citrix ADC appliances are configured with the same RPC node password. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. For security, you should change the default RPC node passwords.

  One RPC node exists on each Citrix ADC. This node stores the password, which is checked against the password provided by the contacting system. To communicate with other systems, each Citrix ADC requires knowledge of those systems, including how to authenticate on those systems. RPC nodes maintain this information, which includes the IP addresses of the other systems, and the passwords they require for authentication.

  RPC nodes are implicitly created when adding a node or adding a Global Server Load Balancing (GSLB) site. You cannot create or delete RPC nodes manually.

  > **Note**:
  >
  > If the Citrix ADC appliances in a high availability setup are configured in one-arm mode, you must disable all system interfaces except the one connected to the switch or hub.

  For an IPv6 HA configuration, the following considerations apply:

- You must install the IPv6PT license on both Citrix ADC appliances.
- After installing the IPv6PT license, enable the IPv6 feature by using the GUI or the command line interface.
- Both Citrix ADC appliances require a global NSIP IPv6 address. In addition, network entities (for example, switches and routers) between the two nodes must support IPv6.

Configuring high availability

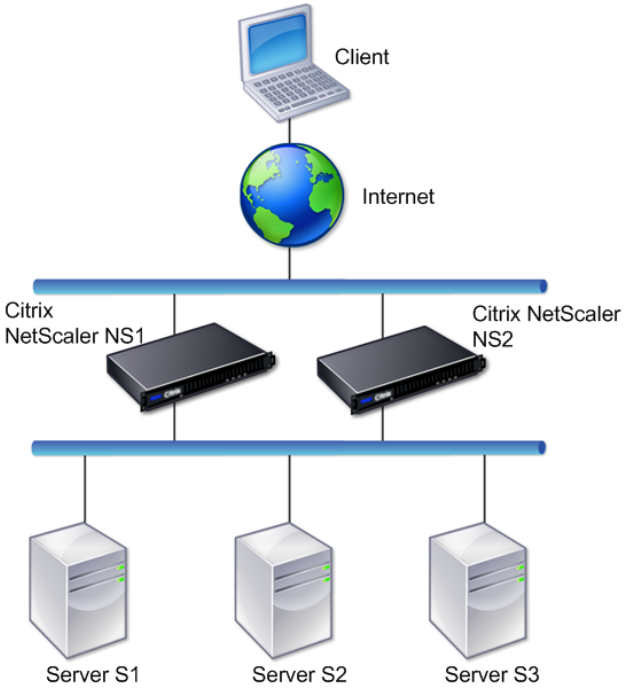
November 22, 2021

To set up a high availability configuration, you create two nodes, each of which defines the other's Citrix ADC IP (NSIP) address as a remote node. Begin by logging on to one of the two Citrix ADC appliances that you want to configure for high availability, and add a node. Specify the other appliance's Citrix ADC IP (NSIP) address as the address of the new node. Then, log on to the other appliance and add a node that has the NSIP address of the first appliance. An algorithm determines which node becomes primary and which becomes secondary.

> **Note**:
>
> The Citrix ADC GUI provides an option that avoids having to log on to the second appliance.

The following figure shows a simple HA setup, in which both nodes are in same subnet.

Figure 1. Two Citrix ADC Appliances Connected in a High Availability Configuration



Adding a remote node

To add a remote Citrix ADC appliance as a node in a high availability setup, you specify a unique node ID and the appliance's NSIP address. When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

> **Note**:
>
> To ensure that each node in the high availability configuration has the same settings, you should synchronize your SSL certificates, startup scripts, and other configuration files with those on the primary node.

To add a node by using the command line interface

At the command prompt, type:

- `add ha node <id> <IPAddress>`
- `show ha node`

Example


```
> add ha node 10 203.0.113.32
<!--NeedCopy-->
```

To disable an HA monitor by using the command line interface

At the command prompt, type:

- `set interface <ifNum> [-haMonitor ( ON | OFF )]`
- `show interface <ifNum>`

Example

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

To add a remote node by using the GUI

Navigate to **System** > **High Availability** and, on the **Nodes** tab, add a new remote node, or edit an existing node.

Disabling or Enabling a Node

You can disable or enable only a secondary node. When you disable a secondary node, it stops sending heartbeat messages to the primary node, and therefore the primary node can no longer check the status of the secondary. When you enable a node, the node takes part in the high availability configuration.

To disable or enable a node by using the command line interface

At the command prompt, type one of the following commands:

- `set ha node -hastatus DISABLED`
- `set ha node -hastatus ENABLED`

To disable or enable a node by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. In the **High Availability Status** list, select **ENABLED (Actively Participate in HA)** or **DISABLED (Do not participate in HA)**.

Removing a Node

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the command line interface

At the command prompt, type:

```
rm ha node <id>
```

Example

```
1 > rm ha node 10
2   Done
3 <!--NeedCopy-->
```

To remove a node by using the GUI

Navigate to **System** > **High Availability** and, on the **Nodes** tab, delete the node.

Configuring the communication intervals

September 14, 2021

The hello interval is the interval at which the heartbeat messages are sent to the peer node. The dead interval is the time interval after which the peer node is marked DOWN if heartbeat packets are not received. The heartbeat messages are UDP packets sent to port 3003 of the other node in an HA pair. Dead interval must be set as a multiple of hello interval.

To set the hello and dead intervals by using the command line interface

At the command prompt, type:

- `set HA node [-helloInterval <msecs>] [-deadInterval <secs>]`
- `show HA node <id>`

To set the hello and dead intervals by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. Set the following parameters:
   - Hello Interval (msecs)
   - Dead Interval (secs)

Configuring synchronization

September 14, 2021

Synchronization is a process of duplicating the configuration of the primary node on the secondary node. The purpose of synchronization is to ensure that there is no loss of configuration information between the primary and the secondary nodes, regardless of the number of failovers that occur. Synchronization uses port 3010.

Synchronization is triggered by either of the following circumstances:

- The secondary node in an HA setup comes up after a restart.
- The primary node becomes secondary after a failover.

Automatic synchronization is enabled by default. You can also force synchronization.

Disabling or enabling synchronization

Automatic HA synchronization is enabled by default on each node in an HA pair. You can enable or disable it on either node.

To disable or enable automatic synchronization by using the command line interface

At the command prompt, type:

- `set HA node -haSync DISABLED`
- `set HA node -haSync ENABLED`

To disable or enable synchronization by using the GUI

1. Navigate to **System** > **High Availability**.
2. Under HA Synchronization, clear or select the Secondary node will fetch the configuration from Primary option.

Forcing the secondary node to synchronize with the primary node

In addition to automatic synchronization, the Citrix ADC supports forced synchronization. You can force the synchronization from either the primary or the secondary node. When you force synchronization from the secondary node, it starts synchronizing its configuration with the primary node.

However, if synchronization is already in progress, forced synchronization fails and the system displays a warning. Forced synchronization also fails in any of the following circumstances:

- You force synchronization on a standalone system.

- The secondary node is disabled.
- HA synchronization is disabled on the secondary node.

To force synchronization by using the command line interface

At the command prompt, type:

```
force HA sync
```

To force synchronization by using the GUI

1. Navigate to **System** > **High Availability**.
2. On the **Nodes** tab, in the Action list, click **Force Synchronization**.

Synchronizing configuration files in a high availability setup

September 14, 2021

In a high availability setup, all configuration files are synchronised automatically from the primary node to the secondary node at an interval of one minute. Synchronizing configuration files can be performed manually by using the command line interface or the GUI at either the primary or the secondary node.

Files located on the secondary that are specific to the secondary (not present on the primary) are not deleted during the synchronization.

To synchronize files in a high availability setup by using the command line interface

At the command prompt, type:

```
sync HA files <mode>
```

Example

```
1 > sync HA files all
2  Done
3 <!--NeedCopy-->
```

```
1 > sync HA files ssl
2  Done
3 <!--NeedCopy-->
```


Parameter descriptions (of the command listed in the CLI procedure)

```
sync ha files <mode>
```

`mode`

Specify one of the following modes of synchronization.

- **all** - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, and Application Firewall XML objects.
- **bookmarks** - Synchronize all Access Gateway bookmarks.
- **ssl** - Synchronize all certificates, keys, and CRLs for the SSL feature.
- **htmlinjection** - Synchronize all scripts configured for the HTML injection feature.
- **imports** - Synchronizeall XML objects (for example, WSDLs, schemas, error pages) configured for the application firewall.
- **misc** - Synchronize all license files and the rc.conf file.
- **all_plus_misc** - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, application firewall XML objects, licenses, and the rc.conf file.

To synchronize files in a high availability setup by using the GUI

Navigate to **System** > **Diagnostics** and, in the **Utilities** group, click **Start HA files synchronization**.

Configuring command propagation

September 14, 2021

In an HA setup, any command issued on the primary node propagates automatically to, and is executed on, the secondary before it is executed on the primary. If command propagation fails, or if command execution fails on the secondary, the primary node executes the command and logs an error. Command propagation uses port 3010.

In an HA pair configuration, command propagation is enabled by default on both the primary and secondary nodes. You can enable or disable command propagation on either node in an HA pair. If you disable command propagation on the primary node, commands are not propagated to the secondary node. If you disable command propagation on the secondary node, commands propagated from the primary are not executed on the secondary node.

> **Note**
>
> After reenabling propagation, remember to force synchronization.

If synchronization occurs while you are disabling propagation, any configuration-related changes that you make before the disabling of propagation takes effect are synchronized with the secondary node. This is also true for cases where propagation is disabled while synchronization is in progress.

To disable or enable command propagation by using the command line interface

At the command prompt, type:

- set HA node -haProp DISABLED
- set HA node -haProp ENABLED

To disable or enable command propagation by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. Clear or select the Primary node will propagate configuration to the Secondary option.

Restricting high availability synchronization traffic to a VLAN

September 14, 2021

In a high availability (HA) deployment, traffic related to maintaining the HA configuration flows between the two HA nodes. This traffic is of the following types:

- Config synchronization
- Config propagation
- Connection mirroring
- Load balancing persistency config synchronization
- Persistent session synchronization
- Session state synchronization

Proper flow of this HA related traffic between the two nodes is critical for the functioning of the HA deployment. Typically, the HA related traffic is small in volume but can become very high during a failover. It becomes very high if stateful connection failover is enabled and the node that was primary before the failover was handling a large number of connections.

By default, the HA related traffic flows through the VLANs to which the NSIP address is bound. To accommodate a potential surge in this traffic, you can separate the HA related traffic from the management traffic and restrict its flow to a separate VLAN. This VLAN is called the HA SYNC VLAN.

Points to consider before Configuring an HA SYNC VLAN

- The configuration of an HA SYNC VLAN is neither propagated nor synchronized. In other words, the HA SYNC VLAN is node specific and is configured independently on each node.
- HA SYNC VLAN configuration is removed when you clear the configuration in only FULL mode.
- HA MON must be set to OFF for interfaces that are part of the HA SYNC VLAN, to avoid a situation in which both nodes function as the primary node.
- Management interfaces (for example, 0/1 and 0/2) must not be part of the HA SYNC VLAN, so that HA related traffic does not flow through management interfaces.
- Citrix recommends disabling high availability heartbeat messages on management interfaces and enabling on HA SYNC VLAN interfaces. After meeting these recommendations, high availability heartbeat messages can also be enabled on data interfaces.

  For more information on disabling high availability heartbeat messages on interfaces, see Managing high availability heartbeat messages on a Citrix ADC appliance.

To configure an HA SYNC VLAN on a Citrix ADC node, specify a configured VLAN with the HA SYNC VLAN parameter of the local node entity.

To configure an HA SYNC VLAN on a local node by using the command line:

At the command prompt, type:

- `set ha node -syncvlan <VLANID>`
- `show node`

Parameter Description:

syncvlan (Sync VLAN) - VLAN on which HA related traffic is sent. This includes traffic for synchronization, propagation, connection mirroring, load balancing persistency, configuration synchronization, persistent session synchronization, and session state synchronization. However, HA heartbeats can use any interface.

To configure an HA SYNC VLAN on a node by using the GUI:

1. Navigate to **System** > **High Availability**.
2. Set the **Sync VLAN** parameter while modifying the local node.

Configuring fail-safe mode

September 14, 2021

In an HA configuration, fail-safe mode ensures that one node is always primary when both nodes fail the health check. This is to ensure that when a node is only partially available, backup methods are

enabled to handle traffic as best as possible. The HA fail-safe mode is configured independently on each node.

The following table shows some of the fail-safe cases. The NOT_UP state means that the node failed the health check yet it is partially available. The UP state means that the node passed the health check.

Node A (Primary) Health State	Node B (Secondary) Health State	Default HA Behavior	Fail-Safe Enabled HA Behavior	Description
NOT_UP(failed last)	NOT_UP (failed first)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
NOT_UP (failed first)	NOT_UP(failed last)	A (Secondary), B (Secondary)	A(Secondary), B(Primary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
UP	UP	A (Primary), B (Secondary)	A (Primary), B (Secondary)	If both nodes pass the health check, no change in behavior with fail-safe enabled.
UP	NOT_UP	A(Primary), B(Secondary)	A (Primary), B (Secondary)	If only the secondary node fails, no change in behavior with fail-safe enabled.

Node A (Primary) Health State	Node B (Secondary) Health State	Default HA Behavior	Fail-Safe Enabled HA Behavior	Description
NOT_UP	UP	A(Secondary), B(Primary)	A(Secondary), B(Primary)	If only the primary fails, no change in behavior with fail-safe enabled.
NOT_UP	UP (STAYSEC-ONDARY)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If the secondary is configured as STAYSEC-ONDARY, the primary remains primary even if it fails.

To enable fail-safe mode by using the command line interface

At the command prompt, type:

```
set HA node [-failSafe ( **ON** | **OFF** )]
```

Example

```
1 set ha node -failsafe ON
2 <!--NeedCopy-->
```

To enable fail-safe mode by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. Under **Fail-Safe Mode**, select the **Maintain one Primary** node even when both nodes are unhealthy option.

Configuring Virtual MAC Addresses

September 14, 2021

A virtual MAC address is a floating entity shared by the primary and the secondary nodes in an HA setup.

In an HA setup, the primary node owns all of the floating IP addresses, such as the MIPs, SNIPs, and VIPs. The primary node responds to Address Resolution Protocol (ARP) requests for these IP addresses with its own MAC address. As a result, the ARP table of an external device (for example, an upstream router) is updated with the floating IP address and the primary node's MAC address.

When a failover occurs, the secondary node takes over as the new primary node. It then uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it acquired from the primary. However, the MAC address that the new primary advertises is the MAC address of its own interface.

Some devices (notably a few routers) do not accept the GARP messages generated by the Citrix ADC appliance. As a result, some external devices retain the old IP to MAC mapping advertised by the old primary node. This can result in a site going down.

You can overcome this problem by configuring a virtual MAC on both nodes of an HA pair. Both nodes then possess identical MAC addresses. Therefore, when failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

To create a virtual MAC, you need to first create a Virtual Router ID (VRID) and bind it to an interface. (In an HA setup, you need to bind the VRID to the interfaces on both nodes.) Once the VRID is bound to an interface, the system generates a virtual MAC with the VRID as the last octet.

This section includes the following details:

- Configuring IPv4 virtual MACs
- Configuring IPv6 virtual MAC6s

Configuring IPv4 virtual MACs

When you create a IPv4 virtual MAC address and bind it to a interface, any IPv4 packet sent from the interface uses the virtual MAC address that is bound to the interface. If there is no IPv4 virtual MAC bound to an interface, the interface's physical MAC address is used.

The generic virtual MAC is of the form `00:00:5e:00:01:<VRID>`. For example, if you create a VRID with a value of 60 and bind it to an interface, the resulting virtual MAC is 00:00:5e:00:01:3c, where 3c is the hex representation of the VRID. You can create 255 VRIDs with values from 1 to 255.

Creating or modifying an IPv4 virtual MAC

You create an IPv4 virtual MAC by assigning it a virtual router ID. You can then you bind the virtual MAC to an interface. You cannot bind multiple VRIDs to the same interface. To verify the virtual MAC

configuration, you should display and examine the virtual MACs and the interfaces bound to the virtual MACs.

To add a virtual MAC by using the command line interface

At the command prompt, type:

- `add vrID`
- `bind vrid <id> -ifnum <interface_name>`
- `show vrID`

Example

```
> add vrID 100
 Done
> bind vrid 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

To unbind interfaces from a virtual MAC by using the command line interface

At the command prompt, type:

- `unbind vrid <id> -ifnum <interface_name>`
- `show vrID`

To configure a virtual MAC by using the GUI

Navigate to **System** > **Network** > **VMAC** and, on the **VMAC** tab, add a new virtual MAC, or edit an existing virtual MAC.

Removing an IPv4 virtual MAC

To remove an IPv4 virtual MAC, you delete its virtual router ID.

To remove an IPv4 virtual MAC by using the command line interface

At the command prompt, type:

`rm vrid <id>`

Example

```
rm vrid 100s
<!--NeedCopy-->
```


To remove an IPv4 virtual MAC by using the GUI

Navigate to **System** > **Network** > **VMAC** and, on the **VMAC** tab, delete the IPv4 virtual MAC.

Configuring IPv6 virtual MAC6s

The Citrix ADC supports virtual MAC6 for IPv6 packets. You can bind any interface to a virtual MAC6, even if an IPv4 virtual MAC is bound to the interface. Any IPv6 packet sent from the interface uses the virtual MAC6 bound to that interface. If there is no virtual MAC6 bound to an interface, an IPv6 packet uses the physical MAC.

Creating or Modifying a virtual MAC6

You create an IPv6 virtual MAC by assigning it an IPv6 virtual router ID. You can then you bind the virtual MAC to an interface. You cannot bind multiple IPv6 VRIDs to an interface. To verify the virtual MAC6 configuration, you should display and examine the virtual MAC6s and the interfaces bound to the virtual MAC6s.

To add a virtual MAC6 by using the command line interface

At the command prompt, type:

- `add vrID6 <id>`
- `bind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

Example

```
1 > add vrID6 100
2 Done
3 > bind vrID6 100 -ifnum 1/1 1/2 1/3
4  Done
5 <!--NeedCopy-->
```

To unbind interfaces from a virtual MAC6 by using the command line interface

At the command prompt, type:

- `unbind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

To configure a virtual MAC6 by using the GUI

Navigate to **System** > **Network** > **VMAC** and, on the **VMAC6** tab, add a new virtual MAC6, or edit an existing virtual MAC6.

Removing a virtual MAC6

To remove an IPv4 virtual MAC, you delete its virtual router ID.

To remove a virtual MAC6 by using the command line interface

At the command prompt, type:

```
rm vrid6 <id>
```

Example

```
1 rm vrid6 100s
2 <!--NeedCopy-->
```

To remove a virtual MAC6 by using the GUI

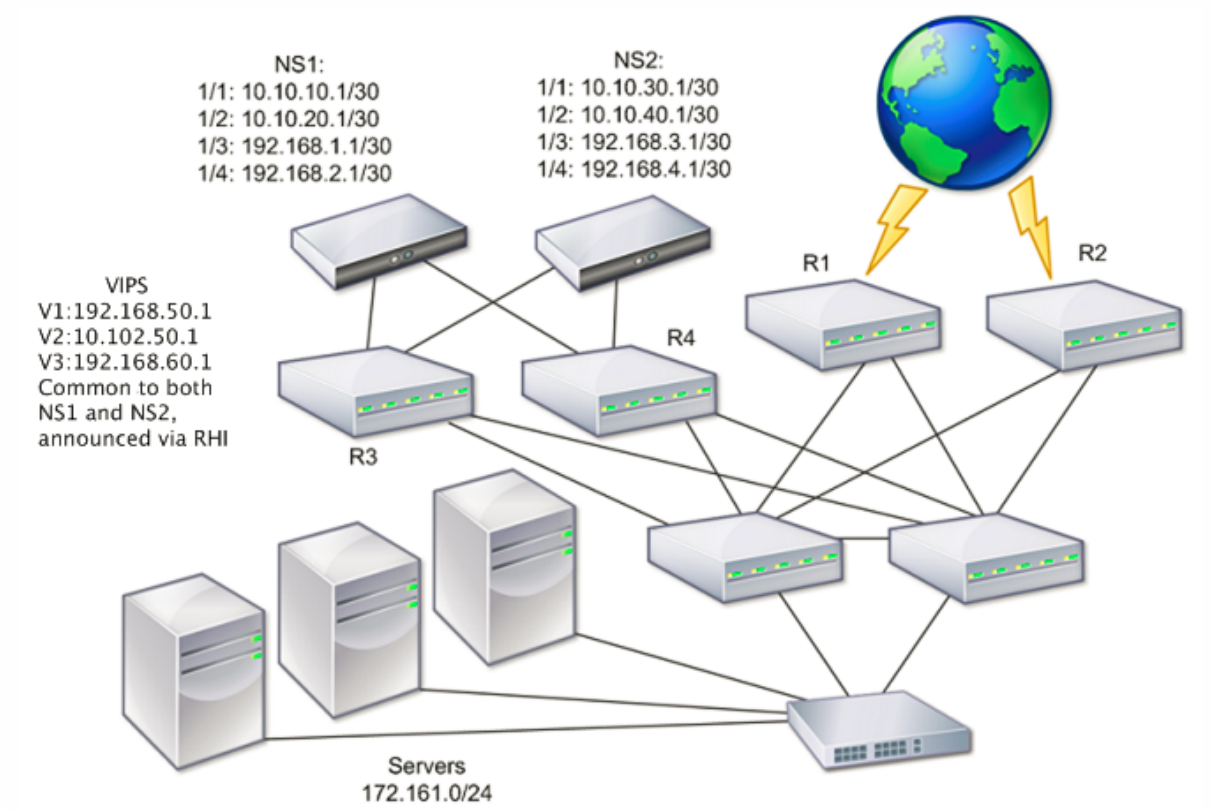
Navigate to **System** > **Network** > **VMAC** and, on the **VMAC6** tab, delete the virtual router ID.

Configuring high availability nodes in different subnets

September 14, 2021

The following figure shows an HA deployment with the two systems located in different subnets:

Figure 1. High Availability over a Routed Network



In the figure, the systems NS1 and NS2 are connected to two separate routers, R3 and R4, on two different subnets. The Citrix ADC appliances exchange heartbeat packets through the routers. This configuration could be expanded to accommodate deployments involving any number of interfaces.

> **Note**:
>
> If you use static routing on your network, you must add static routes between all the systems to ensure that heartbeat packets are sent and received successfully. (If you use dynamic routing on your systems, static routes are unnecessary.)

If the nodes in an HA pair reside on two separate networks, the primary and secondary node must have independent network configurations. This means that nodes on different networks cannot share entities such as SNIP address, VLANs, and routes. This type of configuration, where the nodes in an HA pair have different configurable parameters, is known as Independent Network Configuration (INC) or Symmetric Network Configuration (SNC).

The following table summarizes the configurable entities and options for an INC, and shows how they must be set on each node.

NetScaler entities	Options
IPs (NSIP/SNIPs)	Node-specific. Active only on that node.
VIPs	Floating.

NetScaler entities	Options
VLANs	Node-specific. Active only on that node.
Routes	Node-specific. Active only on that node. Link load balancing routes are floating.
ACLs	Floating (Common). Active on both nodes.
Dynamic routing	Node-specific. Active only on that node. The secondary node should also run the routing protocols and peer with upstream routers.
L2 mode	Floating (Common). Active on both nodes.
L3 mode	Floating (Common). Active on both nodes.
Reverse NAT (RNAT)	RNAT configuration with the NAT IP address set to a virtual server IP address (VIP) because the VIP address is floating (common).

As in configuring HA nodes in the same subnet, to configure HA nodes in different subnets, you log on to each of the two Citrix ADC appliances and add a remote node representing the other appliance.

Adding a Remote Node

When two nodes of an HA pair reside on different subnets, each node must have a different network configuration. Therefore, to configure two independent systems to function as an HA pair, you must specify INC mode during the configuration process.

When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

To add a node by using the command line interface

At the command prompt, type:

- `add ha node <id> <IPAddress> -inc ENABLED`
- `show ha node`

Example

```
1 > add ha node 3 10.102.29.170 -inc ENABLED
2  Done
3 > add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
4   Done
```



```
<!--NeedCopy-->
```

To disable an HA monitor by using the command line interface

At the command prompt, type:

- `set interface <ifNum> [-haMonitor ( **ON** | **OFF** )]`
- `show interface <ifNum>`

Example

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

To add a remote node by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, add a new remote node.
2. Make sure to select the Turn off HA monitor on interfaces/channels that are down and Turn on INC (Independent Network Configuration) mode on self mode options.

Removing a Node

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the command line interface

At the command prompt, type:

```
rm ha node <id>
```

Example

```
> rm ha node 2
  Done
<!--NeedCopy-->
```

To remove a node by using the GUI

Navigate to **System** > **High Availability** and, on the **Nodes** tab, delete the node.

> **Note**:
>
> You can use the Network Visualizer to view the Citrix ADC appliances that are configured as a high availability (HA) pair and perform high availability configuration tasks.

Configuring route monitors

September 14, 2021

You can use route monitors to make the HA state dependent on the internal routing table, whether or not the table contains any dynamically learned or static routes. In an HA configuration, a route monitor on each node watches the internal routing table to make sure that a route entry for reaching a particular network is always present. If the route entry is not present, the state of the route monitor changes to DOWN.

When a Citrix ADC appliance has only static routes for reaching a network, and you want to create a route monitor for the network, you must enable monitored static routes (MSR) for the static routes. MSR removes unreachable static routes from the internal routing table. If MSR is disabled on static routes, an unreachable static route can remain in the internal routing table, defeating the purpose of having the route monitor.

Route Monitors are supported both in non-INC and INC mode.

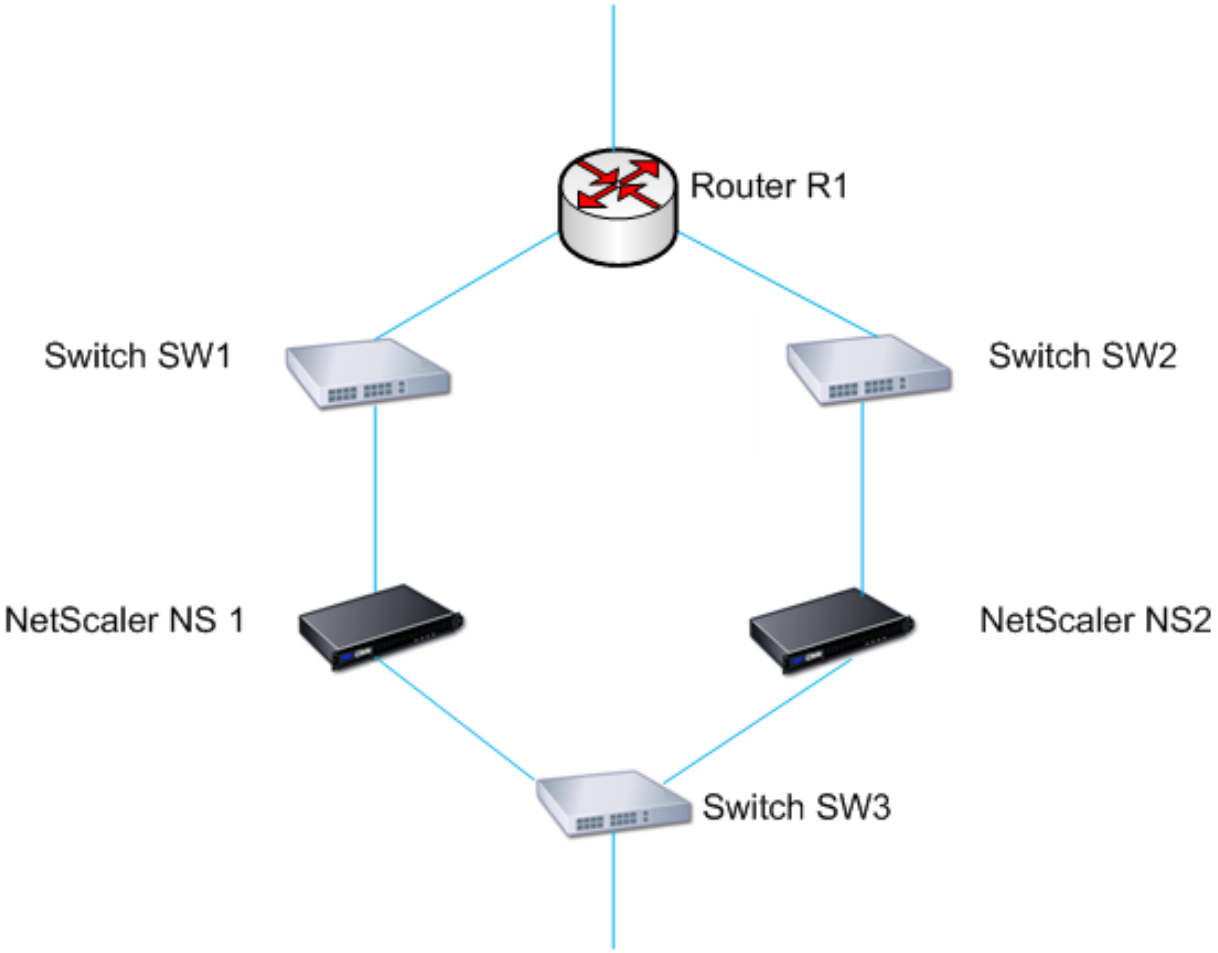
Route Monitors in HA in non-INC mode	Route Monitors in HA in INC mode
Route monitors are propagated by nodes and exchanged during synchronization.	Route monitors are neither propagated by nodes nor exchanged during synchronization.
Route monitors are active only in the current primary node.	Route monitors are active on both the primary and the secondary node.
The Citrix ADC appliance always displays the state of a route monitor as UP irrespective of the whether the route entry is present or not in the internal routing table.	The Citrix ADC appliance displays the state of the route monitor as DOWN if the corresponding route entry is not present in the internal routing table.
A route monitor starts monitoring its route after 180 seconds in the following cases [This is done to allow dynamic routes to get learnt, which may take 180 secs]: reboot, failover, set route6 command for v6 routes, set route msr enable/disable command for v4 routes, adding a new route monitor.	-

Route monitors are useful in a non-INC mode HA configuration where you want the non-reachability of a gateway from a primary node to be one of the conditions for HA failover.

Consider an example of a non-Inc mode HA setup in a two-arm topology that has Citrix ADC appliances NS1 and NS2 in the same subnet, with router R1 and switches SW1, SW2, and SW3.

Because R1 is the only router in this setup, you want the HA setup to failover whenever R1 is not reachable from the current primary node. You can configure a route monitor (say, RM1 and RM2, respectively) on each of the nodes to monitor the reachability of R1 from that node.

Figure 1.



With NS1 as the current primary node, the execution flow is as follows:

1. Route monitor RM1 on NS1 monitors NS1's internal routing table for the presence of a route entry for router R1. NS1 and NS2 exchange heartbeat messages through switch SW1 or SW3 at regular intervals.
2. If switch SW1 goes down, the routing protocol on NS1 detects that R1 is not reachable and therefore removes the route entry for R1 from the internal routing table. NS1 and NS2 exchanges heartbeat messages through switch SW3 at regular intervals.

3. Detecting that the route entry for R1 is not present in the internal routing table, RM1 initiates a failover. If route to R1 is down from both NS1 and NS2, failover happens every 180 seconds till one of the appliances is able to reach R1 and restore the connectivity.

Adding a route monitor to a high availability node

A single procedure creates a route monitor and binds it to an HA node.

To add a route monitor by using the command line interface

At the command prompt, type:

- `bind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])`
- `show HA node`

Example

```
> bind HA node 0 -routeMonitor 10.102.71.0 255.255.255.0
 Done
> bind HA node 0 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
 Done
<!--NeedCopy-->
```

To add a route monitor by using the GUI

Navigate to **System** > **High Availability** and, on the **Route Monitors** tab, click **Configure**.

Removing route monitors

To remove a route monitor by using the command line interface

At the command prompt, type:

- `unbind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])`
- show ha node

Example

```
unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
<!--NeedCopy-->
```


To remove a route monitor by using the GUI

Navigate to **System** > **High Availability** and, on the **Route Monitors** tab, delete the route monitor.

Limiting failovers caused by route monitors in non-INC mode

September 14, 2021

In an HA configuration in non-INC mode, if route monitors fail on both nodes, failover happens every 180 seconds until one of the nodes is able to reach all of the routes monitored by the respective route monitors.

However, for a node, you can limit the number of failovers for a given interval by setting the Maximum Number of Flips and Maximum Flip Time parameters on the nodes. When either limit is reached, no more failovers occur, and the node is assigned as primary (but node state as NOT UP) even if any route monitor fails on that node. This combination of HA state as Primary and Node state as NOT UP is called Stick Primary state.

If the node is then able to reach all of the monitored routes, the next monitor failure triggers resetting of the Maximum Number of Flips and Maximum Flip Time parameters on the node and starting the time specified in the Maximum Flip Time parameter.

These parameters are set independently on each node and therefore are neither propagated nor synchronized.

Parameters for limiting the number of failovers

- **Maximum Number of Flips (maxFlips)**

  Maximum number of failovers allowed, within the Maximum Flip Time interval, for the node in HA in non INC mode, if the failovers are caused by route-monitor failure.

- **Maximum Flip Time (maxFlipTime)**

  Amount of time, in seconds, during which failovers resulting from route-monitor failure are allowed for the node in HA in non INC mode.

To limit the number of failovers by using the command line interface

At the command prompt, type:

- `set HA node [-maxFlips < positive_integer>] [-maxFlipTime <positive_integer >]`
- `show HA node [< id>]`

To limit the number of failovers by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the local node.

2. Set the following parameters:

- Maximum Number of Flips
- Maximum Flip Time

```
> set ha node -maxFlips 30 -maxFlipTime 60
Done
> sh ha node
1) Node ID: 0
IP: 10.102.169.82 (NS)
Node State: UP
Master State: Primary
Fail-Safe Mode: OFF
INC State: DISABLED
Sync State: ENABLED
Propagation: ENABLED
Enabled Interfaces : 1/1
Disabled Interfaces : None
HA MON ON Interfaces : 1/1
Interfaces on which heartbeats are not seen :None
Interfaces causing Partial Failure:None
SSL Card Status: NOT PRESENT
Hello Interval: 200 msecs
Dead Interval: 3 secs
Node in this Master State for: 0:4:24:1 (days:hrs:min:sec)

2) Node ID: 1
IP: 10.102.169.81
Node State: UP
Master State: Secondary
Fail-Safe Mode: OFF
INC State: DISABLED
Sync State: SUCCESS
Propagation: ENABLED
Enabled Interfaces : 1/1
Disabled Interfaces : None
HA MON ON Interfaces : 1/1
Interfaces on which heartbeats are not seen : None
Interfaces causing Partial Failure: None
SSL Card Status: NOT PRESENT

Local node information:
Configured/Completed Flips: 30/0
Configured Flip Time: 60
Critical Interfaces: 1/1
```



```
41
42      Done
43 <!--NeedCopy-->
```

SNMP Alarm for Sticky Primary State

Enable HA-STICKY-PRIMARY SNMP alarm in a node of a high availability set up if you want to be alerted of the node becoming sticky primary. When the node becomes sticky primary, it alerts by generating a trap message (stickyPrimary (1.3.6.1.4.1.5951.1.1.0.138)) and sends it to all the configured SNMP trap destinations. For more information about configuring SNMP alarms and trap destinations, see onfiguring the Citrix ADC to Generate SNMPv1 and SNMPv2 Traps.

Frequently Asked Questions

Consider an example of a high availability setup of two Citrix ADC appliances NS-1 and NS-2 in non-INC mode. Maximum numbers of flips and maximum flip time in both the nodes have been set with the same values.

The following table lists the settings used in this example:

Entity	Detail
IP address of NS-1	10.102.173.211
IP address of NS-2	10.102.173.212
Maximum number of flips	2
Maximum flip time	200

For information about the maximum number of flips and maximum flip time settings, refer to the pdf.

Configuring failover interface set

September 14, 2021

A Failover Interface Set (FIS) is a logical group of interfaces. In an HA configuration, using a FIS is a way to prevent failover by grouping interfaces so that, when one interface fails, other functioning interfaces are still available. A FIS can also be configured for the nodes of a Citrix ADC cluster.

HA MON interfaces that are not bound to an FIS are known as critical interfaces (CI) because if any of them fails, failover is triggered.

> **Note**:
>
> An FIS does not create an active and standby configuration. It also does not prevent bridging loops when connecting to links to the same VLAN.

Creating or Modifying an FIS

To add an FIS and bind interfaces to it by using the command line interface

At the command prompt, type:

- `add fis <name>`
- `bind fis \<name\> \<ifnum\> ...`
- `show fis \<name\>`

Example

```
1 > add fis fis1
2  Done
3 > bind fis fis1 1/3 1/5
4  Done
5 <!--NeedCopy-->
```

An unbound interface becomes a critical interface (CI) if it is enabled and HA MON is on.

To unbind an interface from an FIS by using the command line interface

At the command prompt, type:

- `unbind fis \<name\> \<ifnum\> ...`
- `show fis \<name\>`

Example

```
1 > unbind fis fis1 1/3
2  Done
3 <!--NeedCopy-->
```

To configure an FIS by using the GUI

Navigate to System > High Availability and, on the Failover Interface Set tab, add a new FIS, or edit an existing FIS.

Removing an FIS

When the FIS is removed, its interfaces are marked as critical interfaces.

To remove an FIS by using the command line interface

At the command prompt, type:

```
rm fis <name>
```

Example

```
1 > rm fis fis1
2   Done
3 <!--NeedCopy-->
```

To remove an FIS by using the GUI

Navigate to **System** > **High Availability** and, on the **Failover Interface Set** tab, delete the FIS.

Understanding the causes of failover

September 14, 2021

The following events can cause failover in an high availability configuration:

1. If the secondary node does not receive a heartbeat packet from the primary for a period of time that exceeds the dead interval set on the secondary. (See Note 1.)
2. The primary node experiences a hardware failure of its SSL card.
3. The primary node does not receive any heartbeat packets on its network interfaces for three seconds.
4. On the primary node, a network interface that is not part of a Failover Interface Set (FIS) or a Link Aggregation (LA) channel and has the HA Monitor (HAMON) enabled, fails. (See Note 2.)
5. On the primary node, all interfaces in an FIS fail. (See Note 2.)
6. On the primary node, an LA channel with HAMON enabled fails. (See Note 2.)
7. On the primary node, all interfaces fail (see Note 2). In this case, failover occurs regardless of the HAMON configuration.
8. On the primary node, all interfaces are manually disabled. In this case, failover occurs regardless of the HAMON configuration.
9. You force a failover by issuing the force failover command on either node.
10. A route monitor that is bound to the primary node goes DOWN.

> **Note 1:**
>
> For more information about setting the dead interval, see Configuring the Communication Intervals. Possible causes for a node not receiving heartbeat packets from a peer node include:
>
> - A network configuration problem prevents heartbeats from traversing the network between the HA nodes.
> - The peer node experiences a hardware or software failure that causes it to freeze (hang), reboot, or otherwise stop processing and forwarding heartbeat packets.
>
> **Note 2:**
>
> In this case, fail means that the interface was enabled but goes to the DOWN state, as can be seen from the show interface command or from the GUI. Possible causes for an enabled interface to be in the DOWN state are LINK DOWN and TXSTALL.

Forcing a node to fail over

September 14, 2021

You might want to force a failover if, for example, you need to replace or upgrade the primary node. You can force failover from either the primary or the secondary node. A forced failover is not propagated or synchronized. To view the synchronization status after a forced failover, you can view the status of the node.

A forced failover fails in any of the following circumstances:

- You force failover on a standalone system.
- The secondary node is disabled.
- The secondary node is configured to remain secondary.

The Citrix ADC appliance displays a warning message if it detects a potential issue when you run the force failover command. The message includes the information that triggered the warning, and requests confirmation before proceeding.

You can force a failover on a primary node, secondary node, and when nodes are in listen mode.

- **Forcing Failover on the Primary Node**.

  If you force failover on the primary node, the primary becomes the secondary and the secondary becomes the primary. Forced failover is possible only when the primary node can determine that the secondary node is UP.

  If the secondary node is DOWN, the force failover command returns the following error message: “Operation not possible due to invalid peer state. Rectify and retry.”

If the secondary system is in the claiming state or inactive, it returns the following error message:

```
Operation not possible now. Please wait for the system to stabilize
before retrying.
```

- **Forcing Failover on the Secondary Node**.

  If you run the force failover command from the secondary node, the secondary node becomes primary and the primary node becomes secondary. A force failover can occur only if the secondary node's health is good and it is not configured to stay secondary.

  If the secondary node cannot become the primary node, or if the secondary node was configured to stay secondary (using the STAYSECONDARY option), the node displays the following error message:

  ```
  Operation not possible as my state is invalid. View the node for more
  information.
  ```

- **Forcing Failover When Nodes Are in Listen Mode**.

  When the two nodes of an HA pair are running different versions of the system software, the node running the higher version switches to the listen mode. In this mode, neither command propagation nor synchronization works.

  Before upgrading the system software on both nodes, test the new version on one of the nodes. To do this, you must force a failover on the system that has already been upgraded. The upgraded system then takes over as the primary node, but neither command propagation or synchronization occurs. Also, all connections must be re-established.

> **Important!**
>
> If you force a failover when an HA synchronization operation is in progress, some active data sessions on the HA setup might be lost. So, wait for the HA synchronization operation to be completed before performing the force failover operation.

To force failover on a node by using the command line interface:

At the command prompt, type:

```
force HA failover
```

To force failover on a node by using the GUI:

Navigate to **System** > **High Availability** and, on the **Nodes** tab, select the node, in the Action list, select **Force Failover**.

Forcing the secondary node to stay secondary

September 14, 2021

In an HA setup, the secondary node can be forced to stay secondary regardless of the state of the primary node.

For example, suppose the primary node needs to be upgraded and the process will take a few seconds. During the upgrade, the primary node may go down for a few seconds, but you do not want the secondary node to take over; you want it to remain the secondary node even if it detects a failure in the primary node.

When you force the secondary node to stay secondary, it will remain secondary even if the primary node goes down. Also, when you force the status of a node in an HA pair to stay secondary, it does not participate in HA state machine transitions. The status of the node is displayed as STAYSECONDARY.

Forcing the node to stay secondary works on both standalone and secondary nodes. On a standalone node, you must use this option before you can add a node to create an HA pair. When you add the new node, the existing node stops processing traffic and becomes the secondary node. The new node becomes the primary node.

> **Note**:
>
> When you force a system to remain secondary, the forcing process is not propagated or synchronized. It affects only the node on which you run the command.

To force the secondary node to stay secondary by using the command line interface

At the command prompt, type:

```
set ha node -hastatus STAYSECONDARY
```

To force the secondary node to stay secondary by using the GUI

Navigate to **System** > **High Availability**, on the **Nodes** tab, open the local node, and select **STAY SECONDARY**.

Forcing the primary node to stay primary

September 14, 2021

In an HA setup, you can force a healthy primary node to remain primary even after a failover. You can enable this option either on a primary node in an HA pair. This option allows the primary node to be in primary state as long as it is healthy.

On a standalone node, you must use this option before you can add a node to create an HA pair. When you add the new node, the existing node continues to function as the primary node, and the new node becomes the secondary node.

To force the primary node to stay primary by using the command line interface

At the command prompt, type:

```
set ha node -hastatus STAYPRIMARY
```

To force the primary node to stay primary by using the GUI

Navigate to **System** > **High Availability**, on the **Nodes** tab, open the local node, and select **STAY PRIMARY**.

Understanding the high availability health check computation

September 14, 2021

The following table summarizes the factors examined in a health check computation:

- State of the failover interface sets
- State of the critical interfaces
- State of the route monitors

The following table summarizes the health check computation.

Failover interface sets	Critical interfaces	Route monitor	Condition
N	Y	N	If the system has any critical interfaces, all of those critical interfaces must be UP.
Y	Y	N	If the system has any failover interface sets, all of those failover interface sets must be UP.

Failover interface sets	Critical interfaces	Route monitor	Condition
Y	Y	Y	If the system has any route monitors configured, all monitored routes must be present in the failover interface set.

High Availability FAQs

September 14, 2021

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

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

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

2. What are the conditions that trigger synchronization?

   Synchronization is triggered by any of the following conditions:

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

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

The following commands are neither propagated nor synced to the secondary node:

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

> **Note:**
>
> The following configurations are neither synced nor propagated only in HA in INC mode. Each node has its own:
>
> - SNIPs
> - VLANs
> - Routes (except LLB routes)
> - Route monitors
> - RNAT rules (except any RNAT rule with VIP as the NAT IP)
> - Dynamic routing configurations
> - Net profiles

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

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

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

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

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

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

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

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

   Forced synchronization fails in any of the following circumstances:

   - You force synchronization when synchronization is already in progress.

- You force synchronization on a standalone Citrix ADC appliance.
- The secondary node is disabled.
- HA synchronization is disabled on the current secondary node.
- HA propagation is disabled on the current primary node and you force synchronization from the primary.

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

   Synchronizing configuration files fail in either of the following circumstances:

   - On a standalone system.
   - With the secondary node disabled.

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

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

10. What are the conditions for failure of the force failover command?

    A forced failover fails in any of the following circumstances:

    - You force failover on a standalone system.
    - The secondary node is disabled.
    - The secondary node is configured to remain secondary.
    - The primary node is configured to remain primary.
    - The state of the peer node is unknown.

Troubleshooting high availability issues

September 14, 2021

The most common high availability issues involve the high availability feature not working at all, or working only intermittently. Following are common high availability issues, and probable causes and resolutions.

- **Issue**
  The inability of the Citrix ADC appliances to pair the Citrix ADC appliances in a high availability setup.
  - **Cause**
    Network connectivity
    Resolution
    Verify that both the appliances are connected to the switch and the interfaces are enabled.

- **Cause**
  Mismatch in the Password for the default Administrator account
  Resolution
  Verify that the password on both the appliances is the same.
- **Cause**
  IP conflict
  Resolution
  Verify that both the appliances have unique Citrix ADC IP (NSIP) address. The appliances should not have the same NSIP address.
- **Cause**
  Node ID mismatch
  Resolution
  Verify that the Node ID Configuration on both the appliances is unique. The appliances should not have the same Node ID configuration. Additionally, you must assign value for a Node ID between 1 and 64.
- **Cause**
  Mismatch in the password of the RPC node
  Resolution
  Verify that both the nodes have the same RPC node password.
- **Cause**
  An administrator has disabled the remote node
  Resolution
  Enable the remote node.
- **Cause**
  The Firewall application has blocked the heartbeat packets
  Resolution

Verify that the UDP port 3003 is allowed.

- **Issue**
  Both the appliances claim to be the primary appliance.
  - **Cause**
    Missing heartbeat packets between the appliances
    Resolution
    Verify that the UDP port 3003 is not blocked for communication between the appliances.
- **Issue**
  The Citrix ADC appliance is not able to synchronize the configuration.
  - **Cause**
    A Firewall application is blocking the required port.
    Resolution
    Verify that the UDP port 3010 (or UDP port 3008 with secure synchronization) is not blocked

for communication between the appliances.

- **Cause**
  An administrator has disabled synchronization.
  Resolution
  Enable synchronization on the appliance that has the issue.
- **Cause**
  Different Citrix ADC releases or builds are installed on appliances.
  Resolution
  Upgrade the appliances to the same Citrix ADC release or build.

- **Issue**
  Command propagation fails between the appliances.
  - **Cause**
    A Firewall application is blocking the port.
    Resolution
    Verify that the UDP port 3011 (or UDP port 3009 with secure propagation) is not blocked for communication between the appliances.
  - **Cause**
    An administrator has disabled command propagation.
    Resolution
    Enable command propagation on the appliance that has the issue.
  - **Cause**
    Different Citrix ADC releases or builds are installed on appliances.
    Resolution
    Upgrade the appliances to the same Citrix ADC release or build.
- **Issue**
  The Citrix ADC appliances in the high availability pair are unable to run the force failover process.
  - **Cause**
    The Secondary node is disabled.
    Resolution
    Enable the secondary node.
  - **Cause**
    The Secondary node is configured to stay secondary.
    Resolution
    Set the secondary high availability status of the secondary node to Enable from Stay Secondary.
- **Issue**
  The secondary appliance does not receive any traffic after the failover process.
  - **Cause**
    The upstream router does not understand GARP messages of Citrix ADC appliance.

Resolution
Configure virtual MAC address on the secondary appliance.

Managing high availability heartbeat messages on a Citrix ADC appliance

September 14, 2021

The two nodes in a high availability configuration send and receive heartbeat messages to and from each other on all interfaces that are enabled. The heartbeat messages flow regardless of the HA MON setting on these interfaces. If NSVLAN or both (NSVLAN and SYNC) are configured on an appliance, the heartbeat messages flow only through the enabled interfaces that are part of the NSVLAN and SYNCVLAN.

If a node does not receive the heartbeat messages on an enabled interface, it sends critical alerts to the specified Command Center and SNMP managers. These critical alerts give false alarms and draw unnecessary attention from the administrators for interfaces that are not configured as part of the connections to the peer node.

To resolve this issue, the HAHeartBeat option for interfaces and channels is used for enabling or disabling HA heartbeat-message flow on them.

To manage the high availability heartbeat messages on an interface by using the command line interface

At the command prompt, type:

- `set interface <ID> [-HAHeartBeat ( ON | OFF )]`
- `show interface <ID>`

To manage the high availability heartbeat messages on a channel by using the command line interface

At the command prompt, type:

- `set channel <ID> [-HAHeartBeat ( ON | OFF )]`
- `show channel <ID>`

To manage the high availability heartbeat messages for an interface by using the GUI

1. Navigate to **System** > **Network** > **Interfaces**.
2. Enable or disable the **HA Heart Beat** parameter.

To manage the high availability heartbeat messages on a channel by using the GUI

1. Navigate to **System** > **Network** > **Channels**.
2. Enable or disable the **HA Heart Beat** parameter.

Remove and Replace a Citrix ADC in a High Availability Setup

September 14, 2021

This topic helps you to address RMA replacements. Also, this topic has instructions on how to backup configurations, upgrade or downgrade shipped software version, and setup of RPC password on ADC.

Points to Consider

The following configurations are not synchronized or propagated in a high availability configuration in INC (Independent Network Configuration) or non-INC mode:

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

The following configurations are not synced nor propagated in an HA configuration in INC mode (Independent Network Configuration):

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

Instructions

Complete the following steps to replace a Citrix ADC in high availability setup:

- Remove an Active Citrix ADC secondary node
- Configure Replacement secondary node
- Verify and Update the Software Build on Replacement ADC
- Set Password on New secondary to Match primary
- Add Licenses to Replacement ADC
- Creating HA Pair between primary and New secondary node

Remove an Active Secondary Node

1. Log on to both ADCs and run the following command to confirm which node is primary and which node is secondary:

```
show ha node
<!--NeedCopy-->
```

2. Log on to the primary ADC, backup the configurations on the primary node, and copy the files off of the ADC prior to the changes. These files are located under "/var /ns_sys_backup/" directory.

   The steps are as follows:

   a) Save the ADC running configurations to memory:

```
save ns config
<!--NeedCopy-->
```

   b) Create the full backup file package:

```
create system backup -level full
<!--NeedCopy-->
```

   c) Create the basic backup file package:

```
create system backup -level basic
<!--NeedCopy-->
```

3. After all backup files have been generated, be sure to copy them off of the device before proceeding.

   From a windows terminal, open a Command Prompt and copy the backup files off of the ADC and onto your local hard drive. This can be done using the following command:

```
pscp <username>@<NSIP>:<Target file source>  <Target file
   destination>
<!--NeedCopy-->
```

   Example:

```
pscp nsroot@10.125.245.78:/var/ns_sys_backup/backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz c:\nsbackup\backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz
<!--NeedCopy-->
```

   When prompted, enter the password for the specified administrator account, then hit Enter. Repeat these steps until all backup bundles are copied to the local PC before proceeding.

4. SSH into the secondary ADC, and set the unit to the “STAYSECONDARY” status. This will force the unit to not attempt to assume the primary role in the event of a detected failure during the swap. Confirm that you are connected to the secondary ADC before executing this step

```
set ha node  -haStatus <state>
set ha node  -haStatus STAYSECONDARY
<!--NeedCopy-->
```

5. Once the secondary ADC’s **Node State** successfully displays STAYSECONDARY, switch to the primary ADC and delete the secondary node and run the following command:

```
save ns config
<!--NeedCopy-->
```

While logged into the primary ADC, run the following commands

a) Run the following command to identify which numerical value represents the secondary HA node:

```
show ha node
<!--NeedCopy-->
```

b) Run the following command to remove the secondary ADC from the primary HA pair;

```
rm ha node <node ID>
<!--NeedCopy-->
```

c) Run the following command to save the configuration:

```
save ns config
<!--NeedCopy-->
```

d) With the secondary ADC now removed, shutdown, disconnect, and remove the secondary ADC from the network.

Note. Be sure to label all connections before disconnecting.

Configure Replacement Secondary Node

1. With the replacement ADC in place, power up the new device. DO NOT CONNECT the network connections at this point.
2. With boot-up complete, use the console port to connect to the ADC and configure the NSIP that you will use to connect to the unit.

3. When prompted, select **4**.

   Note. In this example, we are using a different NSIP for the replacement ADC. If you wish to use the original secondary unit's IP, You may change it on the replacement before binding the new ADC to the primary HA unit.

4. The ADC should now be booted. Now connect the network interface that will be used for Management traffic, and confirm that the IP address is reachable from your network.

Verify and Update the Software Build on Replacement ADC

Before syncing the new unit to the primary ADC, we need to ensure that both ADCs are running the same build.

1. To verify the version on ADC run the following command:

```
show version
<!--NeedCopy-->
```

2. While on the new secondary ADC, create a subfolder in **/var** to be used for the upgrade.
3. Go to Citrix Downloads and download the appropriate package that matches the build version running on the primary ADC.
4. Download and extract the .tgz file:

```
tar -xvzf "file.tgz"
<!--NeedCopy-->
```

5. Copy the extracted files to the secondary ADC. On your windows terminal, open a "Command Prompt" and navigate to the directory containing the extracted .tgz build package and run the following pscp command:

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

   Example:

```
C:\inetpub>pscp c:\inetpub\build-12.1-47.14_nc.tgz nsroot@10
   .20.245.80:/var/NS_upg_12.1_47.14/build-12.1-47.14_nc.tgz
<!--NeedCopy-->
```

6. After the file has been transferred, return to the secondary ADC and upgrade. For detailed instructions, see Upgrading a Citrix ADX Standalone Appliance.

7. Once the new secondary has rebooted, SSH back into the unit and confirm that the upgrade is successful and the build matches that of the primary.

Set Password on Replacement Secondary Node to Match Primary

Note: If at this point you want to change the management IP (NSIP) address of the new secondary ADC, you may do so before moving forward.

Change the password on the new secondary ADC to match the password that is currently on the primary ADC.

1. Make that the default administrator (nsroot) account password is the same as the primary ADC. This is accomplished using the following command while logged in through SSH into the new secondary unit:

```
set system user <user> <password>
<!--NeedCopy-->
```

   This command set/resets the password for the specified user.

2. SSH into the primary and new secondary ADC and confirm that passwords match.

Add Licenses to Replacement Secondary Node

With the new ADC updated and ready for pairing, download and install the appropriate licensing for the replacement node.

1. Navigate to https://www.citrix.com to request and download licenses for the new replacement unit.
2. Once you have all appropriate licenses downloaded, SSH into the new secondary ADC and type the following command to see the current state of licensing:

```
show license
<!--NeedCopy-->
```

3. From the Windows terminal command prompt you must now upload the license files to the new secondary ADC using the following command:

   Note. If you have multiple licenses, repeat this step until all licenses are uploaded.

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

   Example:


```
C:\inetpub>pscp c:\inetpub\NS-VPX-3K-LIC-020030ad0024.lic
   nsroot@10.125.245.80:/nsconfig/license/NS-VPX-3K-LIC-020030
   ad0024.lic
<!--NeedCopy-->
```

4. SSH into the new secondary ADC and perform a warm reboot using the following command:

```
reboot  -w
<!--NeedCopy-->
```

   After the unit is restarted, SSH into the unit and run show license command once again. At this point, the licenses should be applied.

Set up High Availability between primary and New Secondary Node

At this point, we are now ready to join the Citrix ADC units into a high availability pair. For more information, see Configuring high availability.

Request retry

November 22, 2021

When a Citrix ADC appliance receives an HTTP request but has a connection failure with a back-end server, the appliance uses a retry directive. The request retry addresses connection failure scenarios and enables the appliance to choose the next available service and forward the request. By doing a request retry, the client can save round trip time (RTT).

Request retry feature is applicable for the following connection failure scenarios:

- If a back-end server resets a TCP connection when an HTTP request is received. For more details, see Request retry.
- If a back-end server resets a TCP connection during connection establishment. For more details, see Request retry.
- If a response from a back-end server times out (based on the configured time-out value) when an appliance sends an HTTP request. For more details, see Request retry.

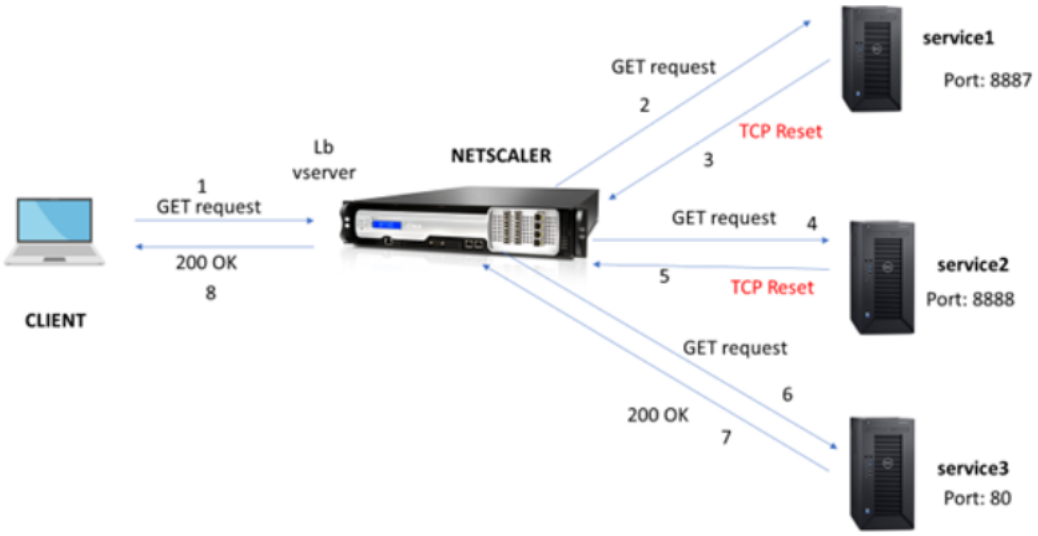
Request retry if back-end server resets TCP connection

September 14, 2021

When a back-end server resets a TCP connection, the request retry feature forwards the request to the next available server, instead of sending the reset to the client. By doing reload balancing, the client saves RTT when the appliance initiates the same request to next available service.

How request retry works when back-end server resets a TCP connection

The following diagram shows how components interact with each other.



1. The process starts by enabling appqoe feature on your appliance.
2. When the client sends an HTTP or HTTPS request, the load balancing virtual server sends the request to the back-end server.
3. If the requested service is unavailable, the back-end server resets the TCP connection.
4. If the appqoe configuration has “retry” enabled with the desired number of retry attempts specified, the load balancing virtual server uses the configured load balancing algorithm to forward the request to the next available application server.
5. After the load balancing virtual server receives the response, the appliance forwards the response to the client.
6. If the available back-end servers is equal or lesser than the retry count and if all the servers send reset, the appliance would respond a 500 internal server error. Consider a scenario with five available servers and the retry count set as six. If all the five servers resets the connection, then the appliance returns a 500 internal server error to the client.
7. Similarly, if the number of back-end servers is more than the retry count and if the back-end servers resets the connection, the appliance forwards the reset to the client. Consider a scenario with three back-end servers and the retry count set as two. If the three servers resets the connection, then the appliance sends a reset response to the client.

Configure request retry for GET method

For configuring retry feature for GET method, you must complete the following steps.

1. Enable AppQoE
2. Add AppQoE action
3. Add AppQoE policy
4. Bind AppQoE policy to load balancing virtual server

Enable AppQoE

At the command prompt, type:

```
enable ns feature appqoe
```

Add AppQoE action

You must configure an AppQoE action to specify if you want the appliance to retry after a TCP reset and the number of retry attempts.

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries <
positive_integer>]
```

Example:

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

Where,
retryOnReset. Enable retry if the back-end server resets a TCP connection.
numretries. Retry count.

Add AppQoE policy

To implement AppQoE you must configure AppQoE policy to prioritize incoming HTTP or SSL request in a specific queue.

At the command prompt, type:

```
add appqoe policy <name> -rule <expression> -action <string>
```

Example:

```
add appqoe policy reset_policy -rule http.req.method.eq(get)-action reset_action
```


Bind appqoe policy to load balancing virtual server

When a back-end server resets a TCP packet request and if you want the load balancing virtual server to forward the request to the next available service, you must bind the load balancing virtual server to the AppQoE policy.

At the command prompt, type:

```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

Example:

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```

Configure request retry for POST requests

You must always exercise caution when you reload balance requests that write data into the back-end server. For such requests, ensure the content length is short. If the content length is long, then it might result in resource consumption. Follow the steps given below to configure reload balancing for POST requests.

1. Enable AppQoE
2. Add AppQoE action
3. Add AppQoE policy
4. Bind appQoE policy to load balancing virtual server

Enable AppQoE

At the command prompt, type:

```
enable ns feature appqoe
```

Add Appqoe action

You must add an AppQoE action to retry after a TCP reset and number of retry attempts.

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries <
positive_integer>]
```

Example:

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```


Add Appqoe policy

To implement AppQoE you must configure AppQoE policy to define how to queue the connections in a specific queue.

At the command prompt, type:

```
add appqoe policy <name> -rule <expression> -action <string>
```

Example:

```
add appqoe policy reset_policy -rule HTTP.REQ.CONTENT_LENGTH.le(2000)-
action reset_action
```

> **Note:**
>
> You can use this configuration if you prefer to restrict the request retry feature for content length less than 2000.

Bind load balancing virtual server to AppQoE policy

When a back-end server resets a TCP packet request and if you want the load balancing virtual server to forward the request to the next available service through a specific queue, you must bind the load balancing virtual server to the AppQoE policy.

At the command prompt, type:

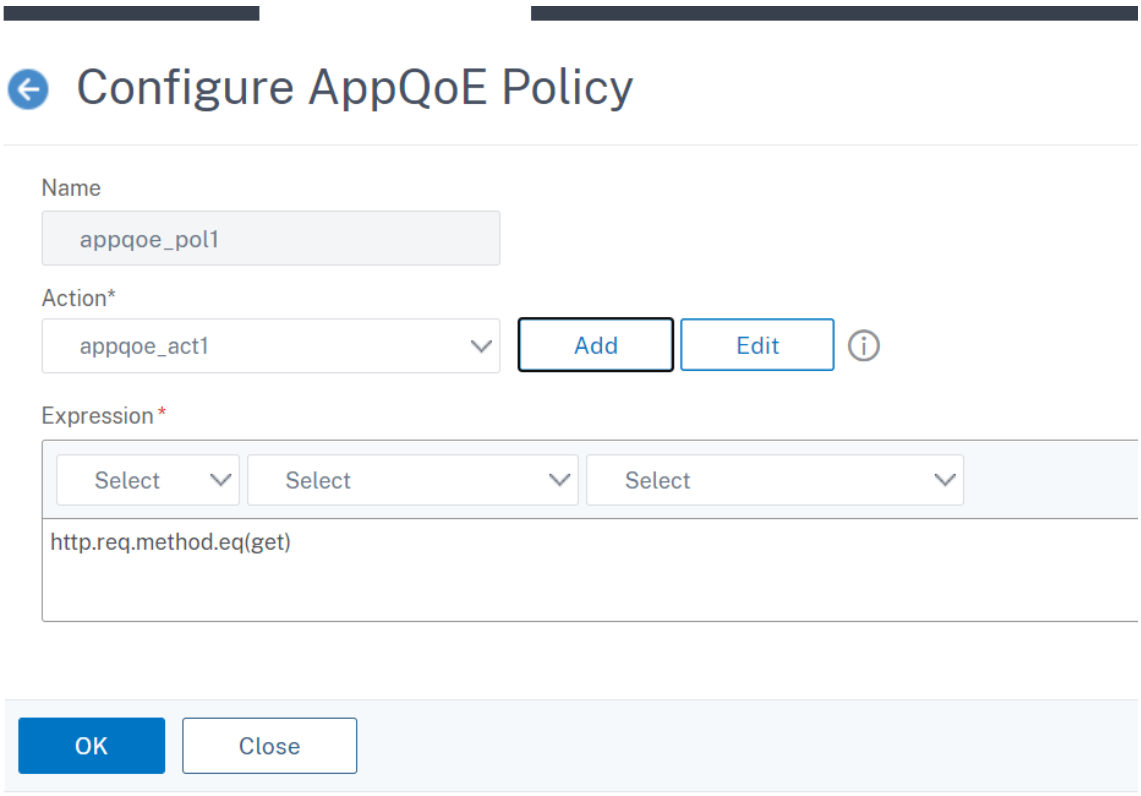
```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

Example:

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```

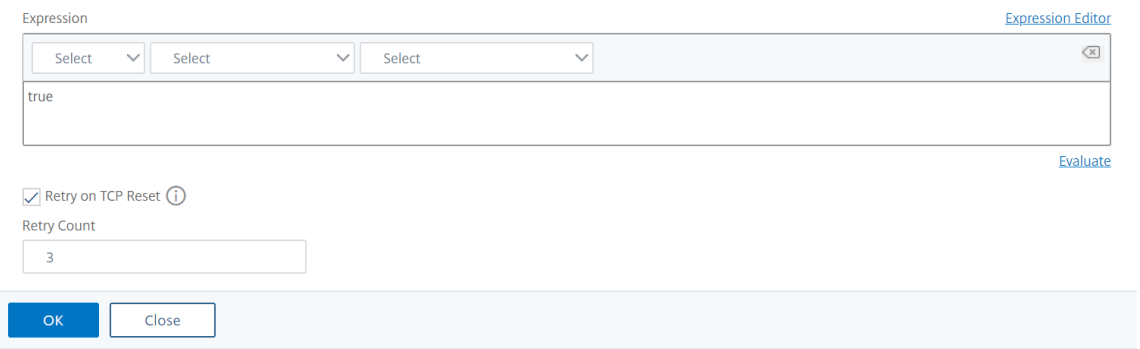
Configure AppQoE policy for request retry by using the Citrix ADC GUI

1. Navigate to **AppExpert** > **AppQoE** > **Policies**.
2. In the **AppQoE Policies** page, click **Add**.
3. In the **Create an AppQoE Policy** page, set the following parameters:
   a. Name. AppQoE policy name
   b. Action. Add or edit an action. To create an action, see section.
   c. Expression. Select or enter `HTTP.REQ.CONTENT_LENGTH.le (2000)` policy expression.
4. Click **Create** and **Close**.



Configure AppQoE action for request retry balancing by using the Citrix ADC GUI

1. Navigate to **AppExpert** > **AppQoE** > **Action**.
2. In the **AppQoE Actions** page, click **Add**.
3. In the **Create AppQoE Action** page, set the following parameters for retry on TCP reset:
   a. Retry on TCP Reset. Select the check box to enable retry action for TCP reset.
   b. Retry Count. Enter the retry count.
4. Click **Create** and **Close**.



Configure request retry for GET method when back-end server resets on TCP SYN establishment

The CLI and GUI configuration is similar to steps followed for GET method. For more information, see Configure request try for GET method section. when back-end server resets a connection section.

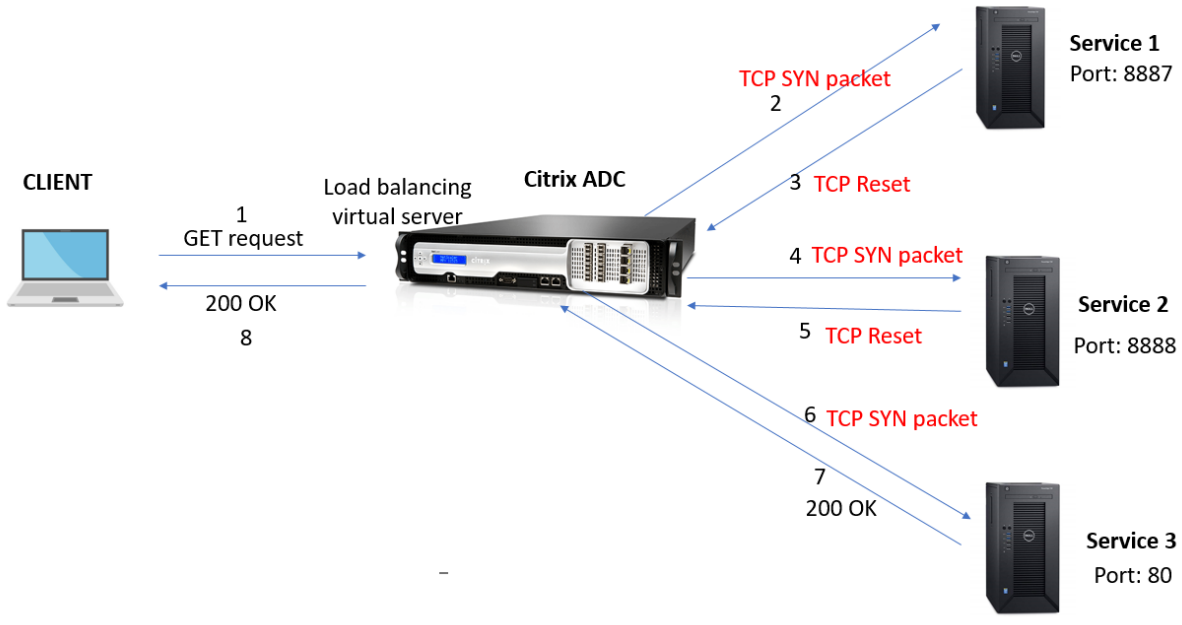
Request retry if back-end server resets TCP connection during connection establishment

September 14, 2021

When a back-end server resets a TCP connection during connection establishment, the request retry feature forwards the request to the next available server, instead of sending the reset to the client. By doing reload balancing, the client saves RTT when the appliance initiates the same request to next available service.

How request retry works when back-end server resets a TCP connection on SYN establishment

The following diagram show the components interact with each other:



1. The process starts by enabling appqoe feature on your appliance.
2. When the client sends an HTTP or HTTPS request, the load balancing virtual server initiates connection to backend server.

3. If the requested service is unavailable on TCP SYN establishment, the back-end server resets the TCP connection.
4. If the appqoe configuration has “retry” enabled with the desired number of retry attempts specified, the load balancing virtual server uses the configured load balancing algorithm to forward the request to the next available application server.
5. After the load balancing virtual server receives the response, the appliance forwards the response to the client.
6. If the available back-end servers is equal or lesser than the retry count and if all the servers send reset, the appliance would respond a 500 internal server error. Consider a scenario with five available servers and the retry count set as six. If all the five servers resets the connection, then the appliance returns a 500 internal server error to the client.
7. Similarly, if the number of back-end servers is more than the retry count and if the back-end servers resets the connection on TCP SYN establishment, the appliance forwards the reset to the client. Consider a scenario with three back-end servers and the retry count set as two. If the three servers resets the connection, then the appliance sends a reset packet to the client.

Configure request retry (GET and POST method) when back-end server resets on TCP SYN establishment

The CLI and GUI configuration is similar to steps followed for GET and POST method. For more information, see Configure request retry for GET method topic, Configure request retry for POST method when back-end server resets a connection section.

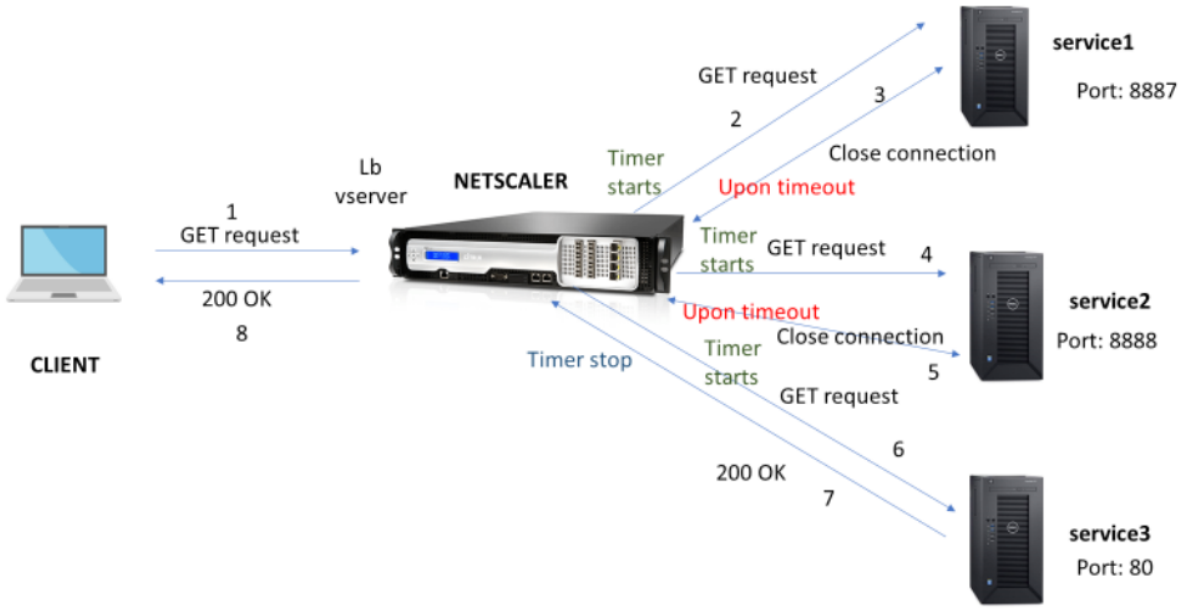
Request retry if back-end server response times out

September 14, 2021

Request retry is available for one more scenario where, if a back-end server takes more time to respond to requests, the appliance performs re-load balancing upon timeout and forwards the request to the next available server.

How request retry works when back-end server response times out

The following diagram show the components interact with each other:



1. The process starts by enabling appqoe feature on your appliance.
2. The appqoe configuration has “retryOnTimeout” parameter in milliseconds.
3. When the appliance sends a request and if the server takes more time to respond, the appliance performs re-load balancing based on the configured timeout value. The appliance resets the connection, chooses another service and forwards the request instead of waiting for the server response.
4. After the load balancing virtual server receives the response, the appliance forwards the response to the client. The usage of a time out parameter prevents the appliance to keep waiting for server response leading to an increased RTT.
5. If the available back-end servers is equal or lesser than the retry count and if all the servers times out for the request , the appliance would respond a 500 internal server error. Consider a scenario with five available servers and the retry count set as six. If all the five servers times out for the request, then the appliance returns a 500 internal server error to the client.
6. Similarly if the number of backend servers is more than the retry count and if the back-end server times out upon a request, the appliance keeps waiting upon the last service until the server sends out a response or client idle connection times out. Consider a scenario with three back-end servers and the retry count set as two. If all the three servers times out upon the request, the appliance keeps waiting upon the third service until the server sends out a response or client idle connection times out.

Configure request retry (GET and POST method) when back-end server response times out

For configuring request retry for GET method on timeout, you must complete the following steps.

1. Enable appqoe
2. Configure appqoe action
3. Add appqoe policy
4. Bind appqoe policy to load balancing virtual server

> **Note:**
>
> The request retry upon timeout scenario is also applicable for POST method.

Enable appqoe

At the command prompt, type:

```
enable ns feature appqoe
```

Add appqoe action for timeout

You must configure the appqoe action to retry on timeout and define the number of retry attempts.

At the command prompt, type:

```
add appqoe action <name> -retryOnTimeout <msecs> -numRetries <positive_integer
>
```

Example:

```
add appqoe action appact1 -retryOnTimeout 35 -numRetries 5
```

Add appqoe policy

To implement appqoe you must configure appqoe policy to define how to queue the connections.

At the command prompt, type:

```
add appqoe policy <name> -rule <rule> -action <name>
```

Example:

```
add appqoe policy timeout_policy -rule http.req.method.eq(get)-action
appact1
```


Bind appqoe policy to load balancing virtual server

When a back-end server takes a long time to respond and if you want the load balancing virtual server to forward the request to the next available service, you must bind the appqoe policy to balancing virtual server.

At the command prompt, type:

```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )]
```

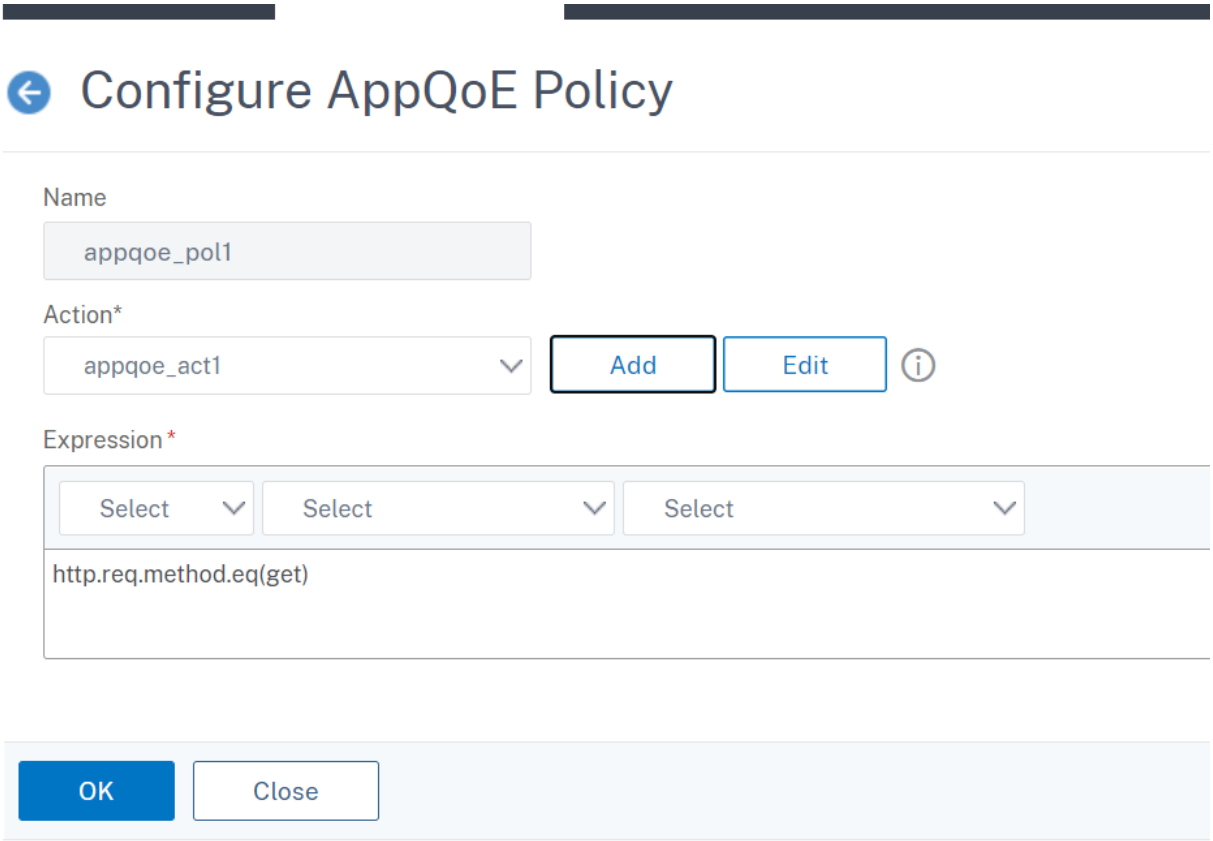
Example:

```
bind lb vserver v1 -policyName timeout_policy -type REQUEST -priority 1
```

Configure AppQoE policy for re-loadbalancing on timeout by using the Citrix ADC GUI

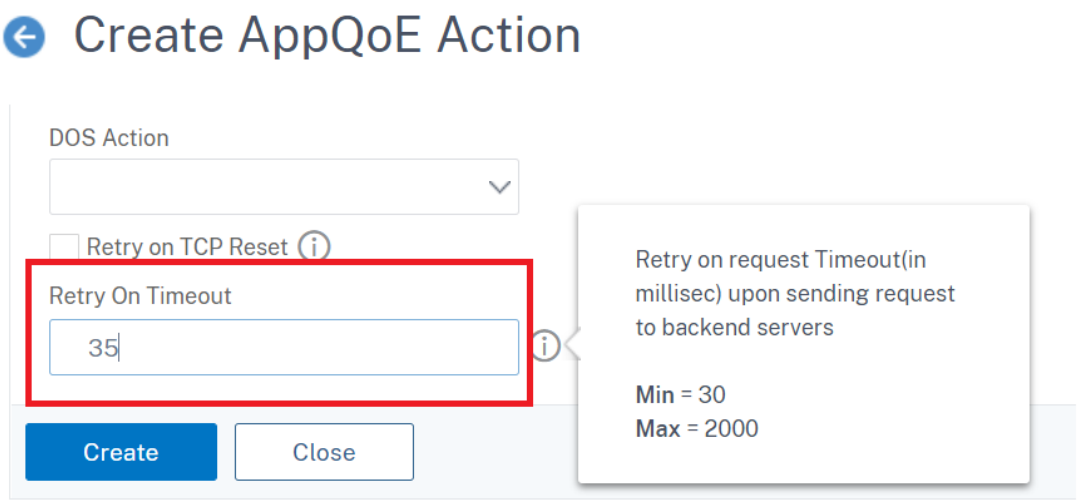
1. Navigate to **AppExpert > AppQoE > Policies**.
2. In the **AppQoE Policies** page, click **Add**.
3. In the **Create an AppQoE Policy** page, set the following parameters:

   a. Name. AppQoE policy name
   b. Action. Add or edit an action. To create a new action, see Create AppQoE Action section.
   c. Expression. Select or enter “http.req.method.eq(get)” policy expression.
4. Click **Create** and **Close**.



Configure AppQoE action for request retry by using the Citrix ADC GUI

1. Navigate to **AppExpert > AppQoE > Action**.
2. In the **AppQoE Actions** page, click **Add**.
3. In the **Create AppQoE Action** page, set the following parameter for retry on back-end server response time out:
   a. Retry on Timeout. Retry on request timeout (in millisec) upon sending request to backend servers.
4. Click **Create** and **Close**.



TCP optimization

September 14, 2021

TCP uses the following optimization techniques and congestion control strategies (or algorithms) to avoid network congestion in data transmission.

Congestion Control Strategies

The TCP has long been used to establish and manage Internet connections, handle transmission errors, and smoothly connect web applications with client devices. But network traffic has become more difficult to control, because packet loss does not depend only on the congestion in the network, and congestion does not necessarily cause packet loss. Therefore, to measure congestion, a TCP algorithm should focus on both packet loss and bandwidth.

Proportional Rate Recovery (PRR) algorithm

TCP Fast Recovery mechanisms reduce web latency caused by packet losses. The new Proportional Rate Recovery (PRR) algorithm is a fast recovery algorithm that evaluates TCP data during a loss recovery. It is patterned after Rate-Halving, by using the fraction that is appropriate for the target window chosen by the congestion control algorithm. It minimizes window adjustment, and the actual window size at the end of recovery is close to the Slow-Start threshold (ssthresh).

TCP Fast Open (TFO)

TCP Fast Open (TFO) is a TCP mechanism that enables speedy and safe data exchange between a client and a server during TCP's initial handshake. This feature is available as a TCP option in the TCP profile bound to a virtual server of a Citrix ADC appliance. TFO uses a TCP Fast Open Cookie (a security cookie) that the Citrix ADC appliance generates to validate and authenticate the client initiating a TFO connection to the virtual server. By using this TFO mechanism, you can reduce an application's network latency by the time required for one full round trip, which significantly reduces the delay experienced in short TCP transfers.

How TFO works

When a client tries to establish a TFO connection, it includes a TCP Fast Open Cookie with the initial SYN segment to authenticate itself. If authentication is successful, the virtual server on the Citrix ADC appliance can include data in the SYN-ACK segment even though it has not received the final ACK segment of the three-way handshake. This saves up to one full round-trip compared to a normal TCP connection, which requires a three-way handshake before any data can be exchanged.

A client and a back-end server perform the following steps to establish a TFO connection and exchange data securely during the initial TCP handshake.

1. If the client does not have a TCP Fast Open Cookie to authenticate itself, it sends a Fast Open Cookie request in the SYN packet to the virtual server on the Citrix ADC appliance.
2. If the TFO option is enabled in the TCP profile bound to the virtual server, the appliance generates a cookie (by encrypting the client's IP address under a secret key) and responds to the client with an SYN-ACK that includes the generated Fast Open Cookie in a TCP option field.
3. The client caches the cookie for future TFO connections to the same virtual server on the appliance.
4. When the client tries to establish a TFO connection to the same virtual server, it sends SYN that includes the cached Fast Open Cookie (as a TCP option) along with HTTP data.
5. The Citrix ADC appliance validates the cookie, and if the authentication is successful, the server accepts the data in the SYN packet and acknowledges the event with an SYN-ACK, TFO Cookie, and HTTP Response.

> **Note:**
>
> If the client authentication fails, the server drops the data and acknowledges the event only with a SYN indicating a session timeout.

1. On the server side, if the TFO option is enabled in a TCP profile bound to a service, the Citrix ADC appliance determines whether the TCP Fast Open Cookie is present in the service to which it is trying to connect.

2. If the TCP Fast Open Cookie is not present, the appliance sends a cookie request in the SYN packet.
3. When the back-end server sends the Cookie, the appliance stores the cookie in the server information cache.
4. If the appliance already has a cookie for the given destination IP pair, it replaces the old cookie with the new one.
5. If the cookie is available in the server information cache when the virtual server tries to reconnect to the same back-end server by using the same SNIP address, the appliance combines the data in SYN packet with the cookie and sends it to the back-end server.
6. The back-end server acknowledges the event with both data and a SYN.

Note: If the server acknowledges the event with only a SYN segment, the Citrix ADC appliance immediately resends the data packet after removing the SYN segment and the TCP options from the original packet.

Configuring TCP fast open

To use the TCP Fast Open (TFO) feature, enable the TCP Fast Open option in the relevant TCP profile and set the TFO Cookie Timeout parameter to a value that suits the security requirement for that profile.

Enable or disable TFO by using the CLI

At the command prompt, type one of the following commands to enable or disable TFO in a new or existing profile.

Note: The default value is DISABLED.

```
1     add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
2     set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
3     unset tcpprofile <TCP Profile Name> - tcpFastOpen
4     Examples
5     add tcpprofile Profile1 - tcpFastOpen
6     Set tcpprofile Profile1 - tcpFastOpen Enabled
7     unset tcpprofile Profile1 - tcpFastOpen
8 <!--NeedCopy-->
```

To set TCP Fast Open cookie timeout value by using the command line interface

At the command prompt, type:

```
1     set tcpparam - tcpfastOpenCookieTimeout <Timeout Value>
2     Example
```



```
3     set tcpprofile  - tcpfastOpenCookieTimeout 30secs
4 <!--NeedCopy-->
```

To configure the TCP Fast Open by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **TCP Fast Open** check box.
3. Click **OK** and then **Done**.

To Configure the TCP Fast Cookie timeout value by using the GUI

Navigate to **Configuration** > **System** > **Settings** > **Change TCP Parameters** and then **Configure TCP Parameters** page to set the TCP Fast Open Cookie timeout value.

TCP HyStart

A new TCP profile parameter, HyStart, enables the HyStart algorithm, which is a slow-start algorithm that dynamically determines a safe point at which to terminate (ssthresh). It enables a transition to congestion avoidance without heavy packet losses. This new parameter is disabled by default.

If congestion is detected, HyStart enters a congestion avoidance phase. Enabling it gives you better throughput in high-speed networks with high packet loss. This algorithm helps maintain close to maximum bandwidth while processing transactions. It can therefore improve throughput.

Configuring TCP HyStart

To use the HyStart feature, enable the Cubic HyStart option in the relevant TCP profile.

To configure HyStart by using the command line interface (CLI)

At the command prompt, type one of the following commands to enable or disable HyStart in a new or existing TCP profile.

```
1 add tcpprofile <profileName> -hystart ENABLED
2 set tcpprofile <profileName> -hystart ENABLED
3 unset tcprofile <profileName> -hystart
4 <!--NeedCopy-->
```

Examples:


```
    add tcpprofile profile1 -hystart ENABLED
    set tcpprofile profile1 -hystart ENABLED
    unset tcprofile profile1 -hystart
<!--NeedCopy-->
```

To configure HyStart support by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **Cubic Hystart** check box.
3. Click **OK** and then **Done**.

TCP burst rate control

It is observed that TCP control mechanisms can lead to a bursty traffic flow on high speed mobile networks with a negative impact on the overall network efficiency. Due to mobile network conditions such as congestion or Layer-2 retransmission of data, TCP acknowledgments arrive clumped at the sender triggering a burst of transmission. These groups of consecutive packets sent with a short inter-packet gap it is called TCP packet burst. To overcome traffic burst, the Citrix ADC appliance uses a TCP Burst Rate Control technique. This technique evenly spaces data into the network for an entire round-trip-time so that the data is not sent into a burst. By using this burst rate control technique, you can achieve better throughput and lower packet drop rates.

How TCP burst rate control works

In a Citrix ADC appliance, this technique evenly spreads the transmission of a packet across the entire duration of the round-trip-time (RTT). This is achieved by using a TCP stack and network packet scheduler that identifies the various network conditions to output packets for ongoing TCP sessions to reduce the bursts.

At the sender, instead of transmitting packets immediately upon receipt of an acknowledgment, the sender can delay transmitting packets to spread them out at the rate defined by scheduler (Dynamic configuration) or by the TCP profile (Fixed configuration).

Configuring TCP burst rate control

To use the TCP Burst Rate Control option in the relevant TCP profile and set the burst rate control parameters.

To set TCP burst rate control by using the command line

At the command prompt, set one of the following TCP Burst Rate Control commands are configured in a new or existing profile.

Note: The default value is DISABLED.

```
add tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
    | Fixed

set tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
    | Fixed

unset tcpprofile <TCP Profile Name> -burstRateControl Disabled |
    Dynamic | Fixed
<!--NeedCopy-->
```

Where,

Disabled – If the Burst rate control is disabled, then a Citrix ADC appliance does not perform burst management other than the maxBurst setting.

Fixed – If the TCP burst rate control is Fixed, the appliance uses the TCP Connection Payload Send Rate value mentioned in the TCP Profile.

Dynamic – If the Burst Rate Control is “Dynamic” the connection is being regulated based on various network conditions to reduce TCP bursts. This mode works only when the TCP connection is in END-POINT mode. When Dynamic Burst Rate control is enabled the maxBurst parameter of the TCP profile is not in effect.

```
add tcpProfile  profile1 -burstRateControl Disabled

set tcpProfile profile1 -burstRateControl Dynamic

unset tcpProfile profile1 -burstRateControl Fixed
<!--NeedCopy-->
```

To set TCP Burst Rate Control parameters by using the command line interface

At the command prompt, type:

```
    set ns tcpprofile nstcp_default_profile  - burstRateControl <type of
        burst rate control>  - tcprate <TCP rate> -rateqmax <maximum
       bytes in queue>

    T1300-10-2> show ns tcpprofile nstcp_default_profile
            Name: nstcp_default_profile
            Window Scaling status:  ENABLED
            Window Scaling factor: 8
            SACK status: ENABLED
```



```
8               MSS: 1460
9               MaxBurst setting: 30 MSS
10              Initial cwnd setting: 16 MSS
11              TCP Delayed-ACK Timer: 100 millisec
12              Nagle's Algorithm: DISABLED
13              Maximum out-of-order packets to queue: 15000
14              Immediate ACK on PUSH packet: ENABLED
15              Maximum packets per MSS: 0
16              Maximum packets per retransmission: 1
17              TCP minimum RTO in millisec: 1000
18              TCP Slow start increment: 1
19              TCP Buffer Size: 8000000 bytes
20              TCP Send Buffer Size: 8000000 bytes
21              TCP Syncookie: ENABLED
22              Update Last activity on KA Probes: ENABLED
23              TCP flavor: BIC
24              TCP Dynamic Receive Buffering: DISABLED
25              Keep-alive probes: ENABLED
26              Connection idle time before starting keep-alive probes: 900
                    seconds
27              Keep-alive probe interval: 75 seconds
28              Maximum keep-alive probes to be missed before dropping
                   connection: 3
29              Establishing Client Connection: AUTOMATIC
30              TCP Segmentation Offload: AUTOMATIC
31              TCP Timestamp Option: DISABLED
32              RST window attenuation (spoof protection): ENABLED
33              Accept RST with last acknowledged sequence number: ENABLED
34              SYN spoof protection: ENABLED
35              TCP Explicit Congestion Notification: DISABLED
36              Multipath TCP: DISABLED
37              Multipath TCP drop data on pre-established subflow:
                   DISABLED
38              Multipath TCP fastopen: DISABLED
39              Multipath TCP session timeout: 0 seconds
40              DSACK: ENABLED
41              ACK Aggregation: DISABLED
42              FRTO: ENABLED
43              TCP Max CWND : 4000000 bytes
44              FACK: ENABLED
45              TCP Optimization mode: ENDPOINT
46              TCP Fastopen: DISABLED
47              HYSTART: DISABLED
48              TCP dupack threshold: 3
49              Burst Rate Control: Dynamic
```



```
50                TCP Rate: 0
51                TCP Rate Maximum Queue: 0
52  <!--NeedCopy-->
```

To configure the TCP Burst Rate Control by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select **TCP Burst Control** option from the drop-down list:
   a) BurstRateCntrl
   b) CreditBytePrms
   c) RateBytePerms
   d) RateSchedulerQ
3. Click **OK** and then **Done**.

Protection against wrapped sequence (PAWS) algorithm

If you enable the TCP timestamp option in the default TCP profile, the Citrix ADC appliance uses the Protection Against Wrapped Sequence (PAWS) algorithm to identify and reject old packets whose sequence numbers are within the current TCP connection's receive window because the sequence has "wrapped" (reached its maximum value and restarted from 0).

If network congestion delays a non-SYN data packet and you open a new connection before the packet arrives, sequence-number wrapping might cause the new connection to accept the packet as valid, leading to data corruption. But if the TCP timestamp option is enabled, the packet is discarded.

By default, the TCP timestamp option is disabled. If you enable it, the appliance compares the TCP timestamp (SEG.TSval) in a packet's header with the recent timestamp (Ts.recent) value. If SEG.TSval is equal to or greater than Ts.recent, the packet is processed. Otherwise, the appliance drops the packet and sends a corrective acknowledgment.

How PAWS works

The PAWS algorithm processes all the incoming TCP packets of a synchronized connection as follows:

1. If `SEG.TSval` < `Ts.recent`: The incoming packet is not acceptable. PAWS sends an acknowledgment (as specified in RFC-793) and drops the packet. Note: Sending an ACK segment is necessary to retain TCP's mechanisms for detecting and recovering from half-open connections.
2. If packet is outside the window: PAWS rejects the packet, as in normal TCP processing.
3. If `SEG.TSval` > `Ts.recent`: `PAWS` accepts the packet and processes it.
4. If `SEG.TSval` <= `Last.ACK.sent`(arriving segment satisfies): PAWS must copies `SEG.TSval` value to `Ts.recent` (is it copied to Ts. Recent field in the db?.
5. If the packet is in sequence: PAWS accepts the packet.

6. If packet is not in sequence: The packet is treated as a normal in-window, out-of-sequence TCP segment. For example, it might be queued for later delivery.
7. If the `Ts.recent` value is idle for more than 24 days: The validity of `Ts.recent` is checked if the PAWS timestamp check fails. If the Ts.recent value is found to be invalid, the segment is accepted and the `PAWS rule` updates the `Ts.recent` with the TSval value from the new segment.

To enable or disable TCP timestamp by using the command line interface

At the command prompt, type:

```
`set nstcpprofile nstcp_default_profile -TimeStamp (ENABLED | DISABLED)
    `
```

To enable or disable TCP timestamp by using the GUI

Navigate to **System** > **Profile** > **TCP Profile**, select the default TCP profile, click **Edit**, and select or clear the **TCP timestamp** check box.

Optimization Techniques

TCP uses the following optimization techniques and methods for optimized flow controls.

Policy based TCP Profile Selection

Network traffic today is more diverse and bandwidth-intensive than ever before. With the increased traffic, the effect that Quality of Service (QoS) has on TCP performance is significant. To enhance QoS, you can now configure AppQoE policies with different TCP profiles for different classes of network traffic. The AppQoE policy classifies a virtual server's traffic to associate a TCP profile optimized for a particular type of traffic, such as 3G, 4G, LAN, or WAN.

To use this feature, create a policy action for each TCP profile, associate an action with AppQoE policies, and bind the policies to the load balancing virtual servers.

For information about using subscriber attributes to perform TCP optimization, see Policy-based TCP Profile.

Configuring policy based TCP profile selection

Configuring policy based TCP profile selection consists of the following tasks:

- Enabling AppQoE. Before configuring the TCP profile feature, you must enable the AppQoE feature.

- Adding AppQoE Action. After enabling the AppQoE feature, configure an AppQoE action with a TCP profile.
- Configuring AppQoE based TCP Profile Selection. To implement TCP profile selection for different classes of traffic, you must configure AppQoE policies with which your Citrix ADC can distinguish the connections and bind the correct AppQoE action to each policy.
- Binding AppQoE Policy to Virtual Server. Once you have configured the AppQoE policies, you must bind them to one or more load balancing, content switching, or cache redirection virtual servers.

Configuring using the command line interface

To enable AppQoE by using the command line interface

At the command prompt, type the following commands to enable the feature and verify that it is enabled:

- `enable ns feature appqoe`
- `show ns feature`

To bind a TCP profile while creating an AppQoE action using the command line interface

At the command prompt, type the following AppQoE action command with the `tcpprofiletobind` option.

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS | NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse |HICResponse )] [-tcpprofiletobind <string>]
show appqoe action
```

To configure an AppQoE policy by using the command line interface

At the command prompt, type:

```
add appqoe policy <name> -rule <expression> -action <string>
```

To bind an AppQoE policy to load balancing, cache redirection or content switching virtual servers by using the command line interface

At the command prompt, type:


```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <priority
>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <priority
>
```

Example

```
1     add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
         ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500
          -slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
         sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
         ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack
         ENABLED -tcpmode ENDPOINT
2     add appqoe action appact1 -priority HIGH -tcpprofile tcp1
3     add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
         action appact1
4     bind lb vserver lb2 -policyName apppol1 -priority 1 -
         gotoPriorityExpression END -type REQUEST
5     bind cs vserver cs1 -policyName apppol1 -priority 1 -
         gotoPriorityExpression END -type REQUEST
6  <!--NeedCopy-->
```

Configuring policy based TCP profiling using the GUI

To enable AppQoE by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Advanced Features**.
3. In the **Configure Advanced Features** dialog box, select the **AppQoE** check box.
4. Click **OK**.

To configure AppQoE policy by using the GUI

1. Navigate to **App-Expert** > **AppQoE** > **Actions**.
2. In the details pane, do one of the following:
3. To create a new action, click **Add**.
4. To modify an existing action, select the action, and then click **Edit**.
5. In the **Create AppQoE Action** or the **Configure AppQoE Action** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):

a) Name—name
b) Action type—respondWith
c) Priority—priority
d) Policy Queue Depth—polqDepth
e) Queue Depth—priqDepth
f) DOS Action—dosAction

6. Click **Create**.

To bind AppQoE policy by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select a server and then click **Edit**.
2. In the **Policies** section and click (+) to bind an AppQoE policy.
3. In the **Policies** slider, do the following:
   a) Select a policy type as AppQoE from the drop-down list.
   b) Select a traffic type from the drop-down list.
4. In the **Policy Binding** section, do the following:
   a) Click **New** to create a new AppQoE policy.
   b) Click **Existing Policy** to select an AppQoE policy from the drop-down list.
5. Set the binding priority and click **Bind** to the policy to the virtual server.
6. Click **Done**.

SACK block generation

TCP performance slows down when multiple packets are lost in one window of data. In such a scenario, a Selective Acknowledgment (SACK) mechanism combined with a selective repeat retransmission policy overcomes this limitation. For every incoming out-of-order packet, you must generate a SACK block.

If the out-of-order packet fits in the reassembly queue block, insert packet info in the block, and set the complete block info as SACK-0. If an out-of-order packet does not fit into the reassembly block, send the packet as SACK-0 and repeat the earlier SACK blocks. If an out-of-order packet is a duplicate and packet info is set as SACK-0 then D-SACK the block.

Note: A packet is considered as D-SACK if it is an acknowledged packet, or an out of order packet which is already received.

Client reneging

A Citrix ADC appliance can handle client reneging during SACK based recovery.

Memory checks for marking end_point on PCB are not considering total available memory

In a Citrix ADC appliance, if the memory usage threshold is set to 75 percent instead of using the total available memory, it causes new TCP connections to bypass TCP optimization.

Unnecessary retransmissions due to missing SACK blocks

In a non-endpoint mode, when you send DUPACKS, if SACK blocks are missing for few out of order packets, triggers more retransmissions from the server.

SNMP for connections bypassed optimization because of overload

The following SNMP ids have been added to a Citrix ADC appliance to track number of connections bypassed TCP optimizations due to overload.

1. 1.3.6.1.4.1.5951.4.1.1.46.131 (tcpOptimizationEnabled). To track the total number of connections enabled with TCP optimization.
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed). To track the total number of connections bypassed TCP Optimization.

Dynamic receive buffer

To maximize TCP performance, a Citrix ADC appliance can now dynamically adjust the TCP receive buffer size.

Tail Loss Probe algorithm

A retransmission timeout (RTO) is a loss of segments at the tail end of a transaction. An RTO occurs if there are application latency issues, especially in short web transactions. To recover loss of segments at the end of a transaction, TCP uses the Tail Loss Probe (TLP) algorithm.
TLP is a sender only algorithm. If a TCP connection is not receiving any acknowledgment for a certain period, TLP transmits the last unacknowledged packet (loss probe). In the event of a tail loss in original transmission, acknowledge from loss probe triggers a SACK or FACK recovery.

Configuring the Tail Loss Probe

To use the Tail Loss Probe (TLP) algorithm, you must enable the TLP option in the TCP profile and set the parameter to a value that suits the security requirement for that profile.

Enable TLP by using the command line

At the command prompt, type one of the following commands to enable or disable TLP in a new or existing profile.

> **Note**:
>
> The default value is DISABLED.

```
add tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED

set tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED

unset tcpprofile <TCP Profile Name> - taillossprobe
```

Examples:

```
add tcpprofile nstcp_default_profile - taillossprobe

set tcpprofile nstcp_default_profile -taillossprobe Enabled

unset tcpprofile nstcp_default_profile -taillossprobe
```

Configure the Tail Loss Probe algorithm by using the Citrix ADC GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **Tail Loss Probe** check box.
3. Click **OK** and then **Done**.

Troubleshooting solutions for Citrix ADC

September 14, 2021

This topic gives you some basic troubleshooting solutions needed to resolve issues that occur in your appliance. It gives you an understanding of NetScaler appliance, how it integrates with the network, and what issues you can expect in basic system features.

How to record a packet trace on Citrix ADC

September 14, 2021

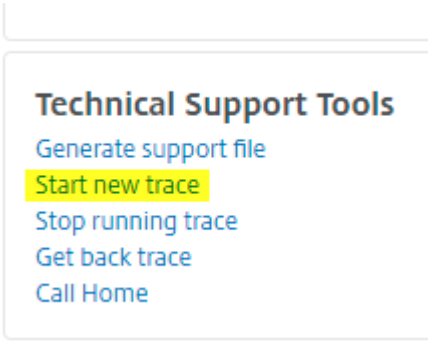
This troubleshooting article explains how an administrator can record a network packet trace using the Citrix ADC GUI.

Points to remember

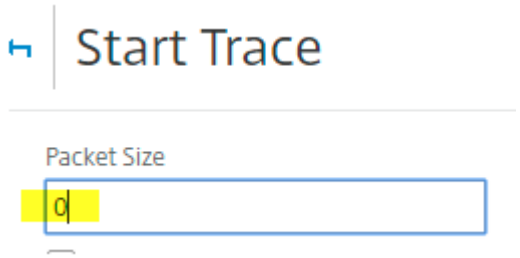
- Citrix recommends you to use the recent Wireshark version from the “automated build section” available in the following webpage: http://www.wireshark.org/download/automated.
- In Citrix ADC version 10.5 or later, to decrypt the capture and ensure ECC (Elliptic Curve Cryptography), Session Reuse and DH parameters are disabled from the virtual server. You must do before you capture a trace.

Record packet trace on NetScaler version 11.1

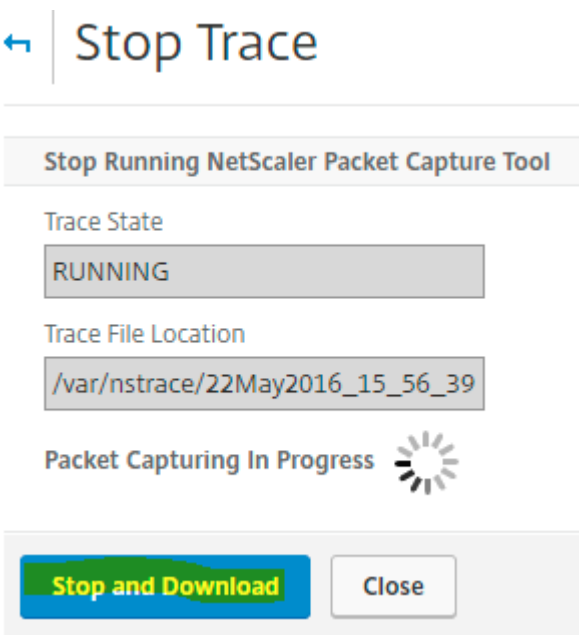
1. Navigate to **System** > **Diagnostics** page.
2. click the **Start new trace** link in the **Diagnostic** page, as shown in the following screenshot.



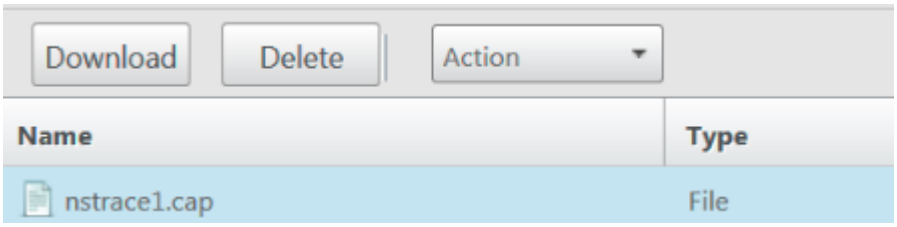
3. Update the packet size to 0 in the **Packet size** field.



4. Click **Start** to start recording the network packet trace.
5. Click **Stop and Download** to stop recording the network packet trace after the test is complete.



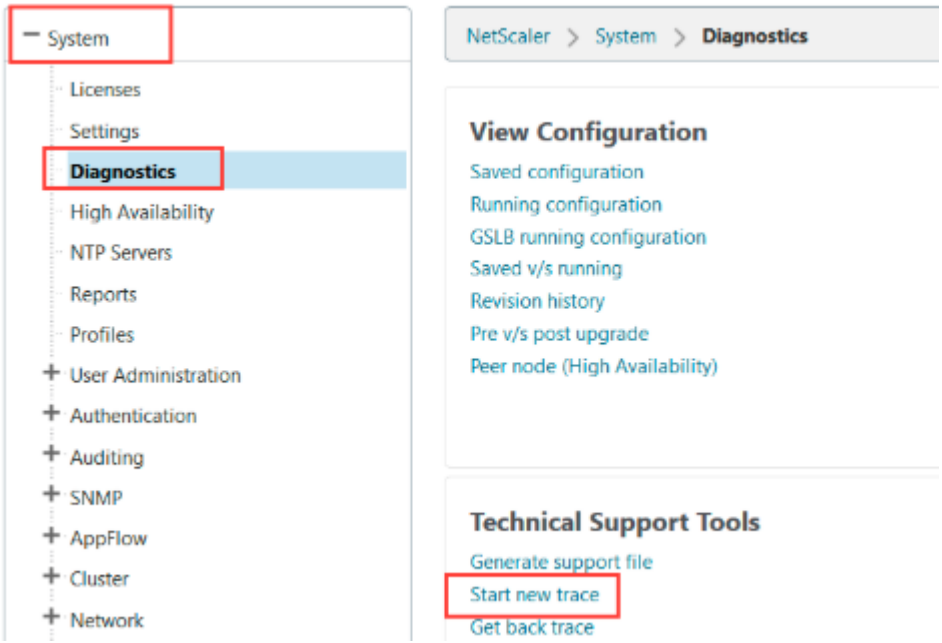
6. Select the required file and click **Select** and click **Download**.



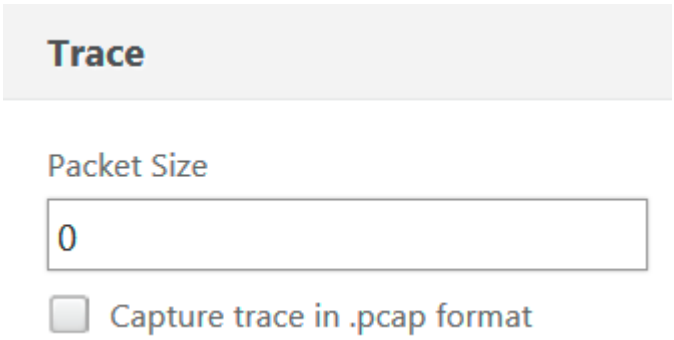
7. Open the network packet trace file with the Wireshark utility to display the content of the file.

Record packet trace on NetScaler 10.5 appliance

1. Navigate to **System** > **Diagnostics** page.

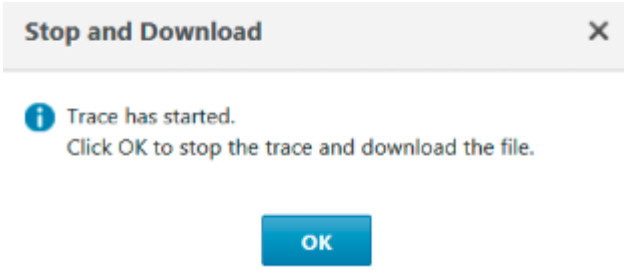


2. Click the **Start new trace** link under **Technical Support Tool**s as shown in the following screen-shot.
3. Update the packet size to 0 in the **Packet Size** field.



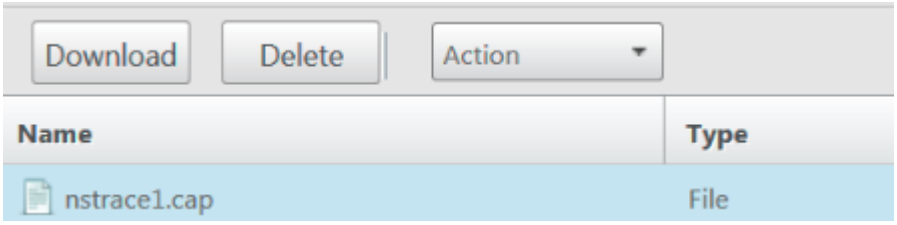
> **Note**: If appliance headers are not required then select Capture trace in .pcap format.

4. Click **Start** to start recording the network packet trace.
5. Click **OK** to stop recording the network packet trace after the test is complete.



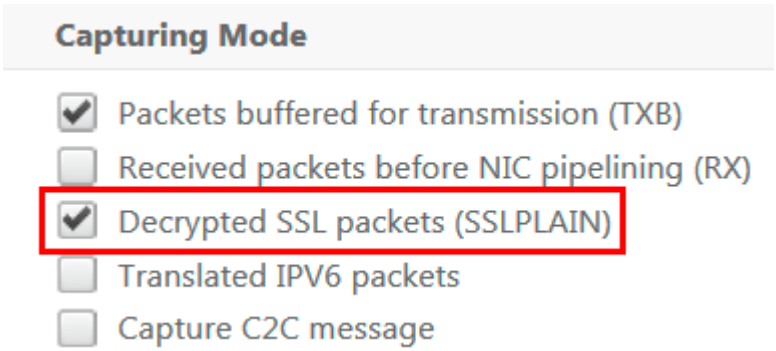
An nstrace.cap file is generated, which contains the network packet trace.

6. Highlight the required file and click **Download**.



7. Specify a destination and save the packet trace.
8. Open the network packet trace file with the Wireshark utility to display the content of the file.

> **Note**: Select Decrypted SSL packets (SSLPLAIN) to decrypt the packet trace without the private key.



Capture SSL master keys

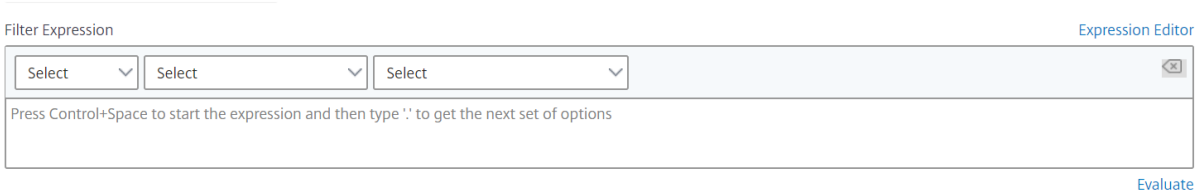
In the 11.0, 11.1 version and above there is an option to capture the session keys which is valid for only for that particular session/nstrace and this option can be used if you do not want to share the private key or use SSLPLAIN mode. For more information, see https://support.citrix.com/article/CTX135889.

Export Session Keys without sharing Private key

In most of the scenarios the private key is not available or shared. In such scenarios we can suggest exporting the **SSL session** keys instead of the private key. Read,
[How to Export and Use SSL Session Keys to Decrypt SSL Traces Without Sharing the SSL Private Key, see https://support.citrix.com/article/CTX135889.

Filters

Also, it is always recommended to add IP based filters while taking traces. The process ensures that you capture only interested traffic which eases your troubleshooting. Adding filters also decreases the load on the appliance while taking traces.



Simple IP-based filters are enough to get the right captures. For more information about `nstrace` filters and examples, see Citrix Documentation page.

Use case to capture a packet trace with virtual server IP filter (both front-end and back end)

Using a filter of the virtual server IP address and enabling the option "–link" in CLI or selecting the option "Trace filtered connection peer traffic" in GUI (available 10.1 and above), you can capture both the front-end and back-end traffic for the IP address.

```
start nstrace -size 0 -filter "CONNECTION.IP.EQ(1.1.1.1)" -link ENABLED

show nstrace
        State:  RUNNING          Scope:  LOCAL            TraceLocation
            :  "/var/nstrace/24Mar2017_16_00_19/..." Nf:  24
                           Time:  3600             Size:  0
                          Mode:  TXB NEW_RX
        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  24
            Mar2017_16_00_19 Filter:  "CONNECTION.IP.EQ(1.1.1.1)" Link:
             ENABLED          Merge:  ONSTOP           Doruntimecleanup
            :  ENABLED
        TraceBuffers:  5000      SkipRPC:  DISABLED       Capsslkeys:
            DISABLED    InMemoryTrace:  DISABLED
<!--NeedCopy-->
```

Capturing cyclic traces

It is always challenging to troubleshoot an intermittent issue. Cyclic tracing is best suited for issues which are intermittent. The traces can be run over a span of few hours or days before the issue occurs. Also, you can use a specific filter and evaluate the size of the trace files that are generated before you run it for a longer time.

Run the following command from the CLI:

```
start nstrace -nf 60 -time 30 -size 0
This particular trace will create 60 files each of them for 30 sec.
   This means the files will start getting overwritten after 60 trace
   files or 30 mins
Show nstrace à To check the status of the nstrace
Stop nstrace à To stop the nstrace.

<!--NeedCopy-->
```

Best Practices

On a unit handling GB of traffic per second, capturing traffic is a very resource intensive process. The impact to resources is mainly in terms of the CPU and the disk space. Disk space impact can be reduced by using filtering expressions. However, the impact on the CPU remains and sometimes causes a slight increase as the appliance now needs to process packets according to the filter before capturing them.

The best practice with regard to tracing is:

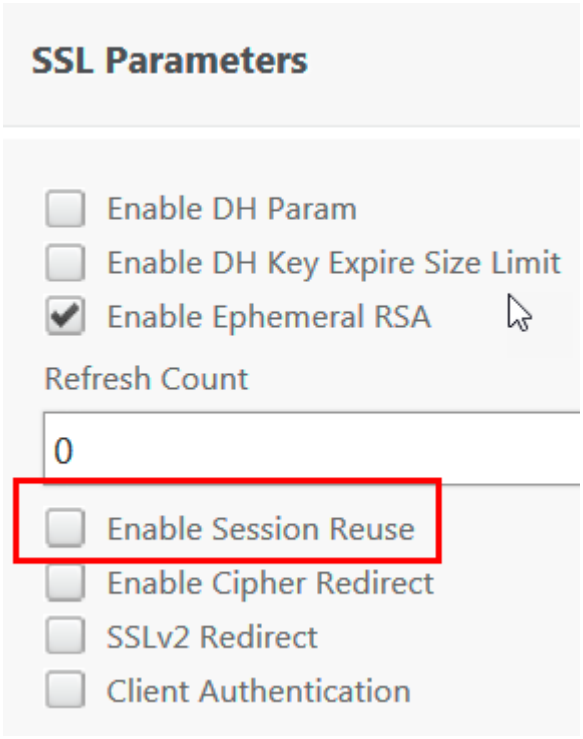
1. The duration for which the trace is run must be as limited as possible when you still ensure the packets of interest are captured.
2. Schedule the tracing activity to happen at a time when the number of users (and hence the traffic) is greatly reduced, such as during off hours.

More resources

Disable session reuse on virtual server from the GUI

Session reuse is disabled when you capture a trace to complete an SSL handshake in the trace. When it is enabled, you can capture a partial handshake in the trace. Ensure you enable the option after the trace collection.
Do not disable an SSL session reuse when the persistence method is sslsession, as it breaks the persistence for existing connections. For more information refer to https://support.citrix.com/article/CTX121925.

1. Open the virtual server and navigate to SSL Parameters.
2. Disable Enable Session Reuse if enabled.



Disable session reuse on virtual server from the CLI

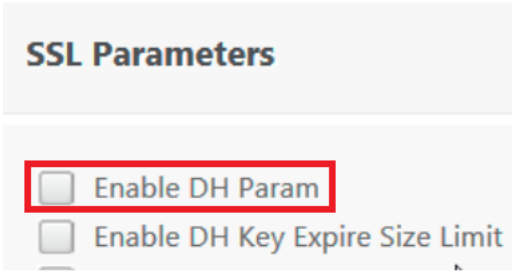
1. SSH to the appliance console.
2. Run the following command to disable DH Param from the virtual server:

```
set ssl vserver "vServer_Name"-sessReuse DISABLED
```

Disable DH parameter on virtual server from the GUI

Refer to https://support.citrix.com/article/CTX213335 To understand about DH Parameter.

1. Open the virtual server and navigate to SSL Parameters.
2. Disable DH Param if enabled.



Disable DH parameter on virtual server from the CLI

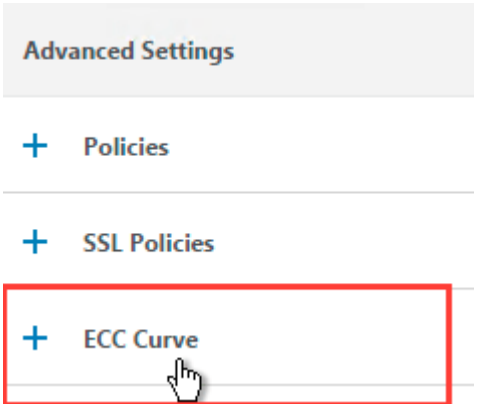
1. SSH to the appliance console.
2. Run the following command to disable DH Param from the virtual server:

```
set ssl vserver "vServer_Name"-dh DISABLED
```

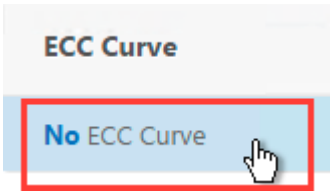
Disable ECC curve on virtual server from the GUI

ECC curve is disabled to decrypt the captured SSL trace with private key. You must not disable the keys if the related SSL ciphers are used. For more information about the ECC curve, see https://support.citrix.com/article/CTX205289

1. Open the virtual server and navigate to ECC Curve.



2. If there is no ECC Curve bound to the virtual server then no other action is required.



3. If any ECC Curve is bound to the virtual server then click the ECC Curve and Unbind it from the virtual server.

Disable ECC curve on virtual server from the CLI

1. SSH to the appliance console.
2. Run the following command for each ECC Curve bound to the virtual server:

```
unbind ssl vserver "vServer_Name"-eccCurveName "ECC_Curve_Name"
```


How to free space on the VAR directory for logging issues with a Citrix ADC appliance

September 14, 2021

The following article explains how an administrator can free the space from the `/var` directory of a Citrix ADC appliance. You can follow the steps when the Citrix GUI is not accessible.

When the amount of disk space is low in the /var directory of the appliance, you might not be able to sign in to the Citrix GUI. In this scenario, you can remove the old log files to create free space in the /var directory.

Points to remember

- Ensure that you back up the files before removing the files from the appliance.

To free space in the `/var` directory of a Citrix ADC appliance, complete the following procedure:

1. Log on to the CLI of Citrix ADC by using SSH. For more information to complete this task, see the Citrix ADC Documentation.
2. After you log on to the Citrix ADC CLI, switch to the shell prompt using the following command. `shell`
3. Run the following command to see the availability of space on the Citrix ADC appliance. `df -h`
4. If the memory capacity of the `/var` directory is filled up to 90 percent, then you must delete few files from this directory.

   - Run the following commands to view the contents of the /var directory:

     ```
     cd /var
     ls -l
     ```

   The directories that are usually of interest are as follows:

   ```
   /var/nstrace - This directory contains trace files.This is the
      most common reason for HDD being filled on the Citrix ADC
      appliance. This is due to an nstrace being left running for
      indefinite amount of time. All traces that are not of interest
      can and should be deleted. To stop an nstrace, go back to the
      CLI and issue stop nstrace command.

   /var/log - This directory contains system specific log files.

   /var/nslog - This directory contains Citrix ADC log files.

   ```



```
/var/tmp/support - This directory contains technical support files, also known as, support bundles. All files not of interest should be deleted.

/var/core - Core dumps are stored in this directory. There will be directories within this directory and they will be labeled with numbers starting with 1. These files can be quite large in size. Clear all files unless the core dumps are recent and investigation is required.

/var/crash - Crash files, such as process crashes are stored in this directory. Clear all files unless the crashes are recent and investigation is required.

/var/nsinstall -  Firmware is placed in this directory when upgrading. Clear all files, except the firmware that is currently being used.
```

- Verify if any of the directories are using more space:

```
du -hs *
44k     cache
2.0k    clusterd
2.0k    configdb
6.0k    core
989M    crash
4.0k    cron
2.0k    dev
6.0k    download
2.0k    gui
2.0k    install
2.0k    krb
2.0k    learnt_data
122M    log
366M    NetScaler
14k     ns_gui
86k     ns_sys_backup
631M    nsinstall
883M    nslog
32k     nsproflog
2.0k    nssynclog
16k     nstemplates
36k     nstmp
4.5G    nstrace
8.1M    opt
```



```
6.0k    pubkey
52k     run
28M     safenet
72M     tmp
2.0k    vmtools
14k     vpn
```

- Delete the files which are not required:

```
rm -r nstrace/*
```

For more help on deleting files see FreeBSD Man Pages.

- Delete the files which are not required.

  `rm -r nstrace/*`

  For more help on deleting files see FreeBSD Man Pages.

- If the log or `nslog` directory is using more space, then run the following commands to open the log directory and view its contents:

```
cd /var/log
ls -l
cd /var/nslog
ls -l
```

1. Ensure that all files are compressed. This is indicated by the .tar.gz file name extension.
2. If you are using Citrix ADM or Command Center, then verify the /var/ns_system_backup directory. Ensure that Citrix ADM or Command Center clears the backup files it creates.

More resources

For information on any of the commands mentioned in the preceding procedure, see - http://ss64.com/bash/

How to download core or crashed files from Citrix ADC appliance

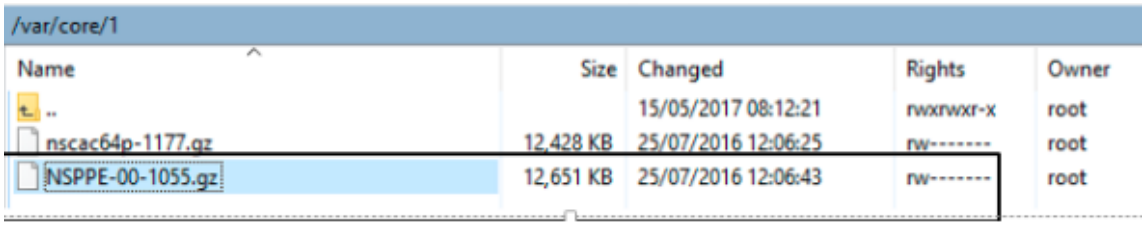
September 14, 2021

This troubleshooting article explains how an administrator can download core or crash files from the Citrix ADC appliance.

Download core or crash files from Citrix ADC appliance using SFTP client

To download the core or crash files from a NetScaler appliance, complete the following procedure:

1. Open WinSCP and log on to the NetScaler Management IP address.
2. Navigate to the `/var/core/1` to download the files.



> **Note:**
>
> To download the latest crash or core file, you can also use the WinSCP tool through command interface. The files can be located either in the core or crash directory.

How to collect performance statistics and event logs

September 14, 2021

You can collect performance statistics of virtual servers and associated services from an archived `newnslog` file present in the `/var/nslog` directory. The `newnslog` files are interpreted by running `/netscaler/nsconmsg`.

Collect performance statistics and event logs using the CLI

You can run the `nsconmsg` command from the Citrix ADC shell prompt to report events.

At the command prompt, type:

`/netscaler/nsconmsg -K /var/nslog/newnslog -d event`

```
1 Displaying event information
2 NetScaler V20 Performance Data
3 NetScaler NS10.5: Build 57.7.nc, Date: May 14 2015, 07:35:21
4 rtime: Relative time between two records in milliseconds
5 seqno rtime event-message                          event-time
6 11648 16310 PPE-0 MonServiceBinding_10.104.20.110:443_(tcp-default)
7 <!--NeedCopy-->
```


View the time span covered by a given “newnslog” file

At the command prompt, type:

```
/netscaler/nsconmsg -K /var/nslog/newnslog -d setime
```

The current data is appended to the `/var/nslog/newnslog` file. NetScaler archives the `newnslog` file automatically every two days by default. To read the archived data, you must extract the archive as shown in the following example:

`cd /var/nslog` - command to go to a particular directory from NetScaler Shell Prompt.

`tar xvfz newnslog.100.tar.gz` - command to extract the tar file.

`/netscaler/nsconmsg -K newnslog.100 -d setime` - Command to check the time span covered by the particular file, in this example `newnslog.100`.

`ls -l` Command checks all the logs file and time stamp associated with those files.

```
root@NETSCALER## cd /var/nslog
root@NETSCALER## ls -l
```

```
1  wheel    461544 Aug  7  2014 newnslog.1.tar.gz
2 -rw-r--r--   1 root       wheel    191067 Aug  7  2014 newnslog.10.tar.
     gz
3 -rw-r--r--   1 root       wheel  11144873 Apr 26 22:04 newnslog.100.tar
     .gz
4 -rw-r--r--   1 root       wheel  11095053 Apr 28 22:04 newnslog.101.tar
     .gz
5 -rw-r--r--   1 root       wheel  11114284 Apr 30 22:04 newnslog.102.tar
     .gz
6 -rw-r--r--   1 root       wheel  11146418 May  2 22:04 newnslog.103.tar
     .gz
7 -rw-r--r--   1 root       wheel  11104227 May  4 22:04 newnslog.104.tar
     .gz
8 -rw-r--r--   1 root       wheel  11297419 May  6 22:04 newnslog.105.tar
     .gz
9 -rw-r--r--   1 root       wheel  11081212 May  8 22:04 newnslog.106.tar
     .gz
10 -rw-r--r--   1 root       wheel  11048542 May 10 22:04 newnslog.107.tar
     .gz
11 -rw-r--r--   1 root       wheel  11101869 May 12 22:04 newnslog.108.tar
     .gz
12 -rw-r--r--   1 root       wheel  11378787 May 14 22:04 newnslog.109.tar
     .gz
13 -rw-r--r--   1 root       wheel  44989298 Apr 11  2014 newnslog.11.gz
14 <!--NeedCopy-->
```


Display the time span within a file

Use the nsconmsg command to only display a span of time within the given file, as shown in the following example:

```
/netscaler/nsconmsg -K /var/nslog/newnslog -s time=22Mar2007:20:00 -T 7 -s ConLb=2 -d oldconmsg
```

Where,

s - time=22Mar2007:20:00:00 is start at March 22, 2007 at exactly 20:00.

T 7 - Displays seven seconds of data

s - Displays detail level of load balancing statistics.

d - Displays statistical information.

> **Note:**
>
> From ADC release 12.1 you need add at the “time” seconds as well, that is: 22Mar2007:20:00:00

The statistical information provided by the -d oldconmsg parameter is recorded every seven seconds. The following is a sample output.

```
1  VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec) Mbps(1.02)
       Pers(OFF) Err(0)
2  Pkt(186/sec, 610 bytes) actSvc(4) DefPol(NONE) override(0)
3  Conn: Clt(253, 1/sec, OE[252]) Svr(3)
4  S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
       Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms)
5  Other: Pkt(36/sec, 712 bytes) Wt(10000) RHits(31555)
6  Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) RP(11) SQ(0)
7  S(10.128.49.39:80:UP) Hits(9731048, 4/sec, P[2929279, 0/sec]) ATr(9)
       Mbps(0.27) BWlmt(0 kbits) RspTime(161.69 ms)
8  Other: Pkt(41/sec, 756 bytes) Wt(10000) RHits(31555)
9  Conn: CSvr(32, 0/sec) MCSvr(19) OE(13) RP(4) SQ(0)
10 S(10.128.49.38:80:UP) Hits(9341366, 5/sec, P[2700778, 0/sec]) ATr(4)
       Mbps(0.27) BWlmt(0 kbits) RspTime(120.50 ms)
11 Other: Pkt(42/sec, 720 bytes) Wt(10000) RHits(31556)
12 Conn: CSvr(37, 0/sec) MCSvr(19) OE(13) RP(9) SQ(0)
13 S(10.128.49.37:80:UP) Hits(9685018, 4/sec, P[2844418, 0/sec]) ATr(3)
       Mbps(0.23) BWlmt(0 kbits) RspTime(125.38 ms)
14 Other: Pkt(38/sec, 670 bytes) Wt(10000) RHits(31556)
15 Conn: CSvr(32, 0/sec) MCSvr(20) OE(10) RP(7) SQ(0)
16 <!--NeedCopy-->
```


> **Note:**
>
> The client connection counts of the individual services do not add up to the client connection count of the virtual server. The reason is because of session reuse between the Citrix ADC appliance and the back-end service.

Virtual Server Output

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec)Mbps(1.02)Pers(
OFF)Err(0)Pkt(186/sec, 610 bytes)actSvc(4)DefPol(NONE)override(0)Conn: Clt
(253, 1/sec, OE[252])Svr(3)
```

The following list describes the virtual server statistics:

1. `IP (IP address:port:state:Load balancing method)`. The IP address and port of the Virtual IP address as configured. The virtual server state or virtual IP address is UP, DOWN, or OUT OF SERVICE; Load balancing method configured for the Virtual IP address.
2. `Hits (##)`. Number of requests that reached the virtual server.
3. `Mbps (##)`. Total traffic Volume on the virtual server (Rx + Tx) converted into Mbits/s
4. `Pers`: Type of persistence configured.
5. `Err (##)`. Number of times an error page was generated by the virtual server.
6. `Pkt (##/sec, ## bytes)`: Volume of network traffic (as packets) passing through the virtual server and average packet size flowing through the virtual server.
7. `actSvc(##)`. Number of active services that are bound to the virtual server.
8. `DefPol (RR)`. Indicates whether the default load balancing method is active. Default load balancing method is used for some number of initial requests to smooth the behavior of the other methods.
9. `Clt (##, ##/sec)`. Number of current client connections to the virtual server rate.
10. `OE [##]`. Number of server connections from the virtual server in open established state.
11. `Svr (##)`. Number of current server connections from the virtual server.

In the preceding output, `Svr(3)` indicates the command collects he statistical sample. There are three active connections for the virtual server to the back-end server, even though there are four services in total. When a client establishes a connection with the virtual server, it is not necessary that the client sends or receives any traffic when the command collects the information. Therefore, it is common to see the `Svr` counter lower than the `OE[]` number. The `Svr` counter represents the number of active connections that are actively sending or receiving data. The Mapped IP address (MIP) or Subnet IP address (SNIP) is connected to the associated back-end server. And, the Citrix ADC tracks the virtual server connected to the back-end server and calculates the counter.

Virtual service output


```
S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
   Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms)
Other: Pkt(36/sec, 712 bytes) Wt(10000) RHits(31555)
Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) RP(11) SQ(0)
<!--NeedCopy-->
```

The following list describes the service statistics:

1. `S (IP address:port:state)`. IP address, port, and state of the service such as, DOWN, UP, or OUT OF SERVICE.
2. `Hits (##, P[##])`. Number of requests directed to the service, Number of requests directed to the service due to configured server persistence.
3. `ATr (##)`. Number of active connections to the service.

> **Note:**
>
> Active connections are ones which have the outstanding request to the service or currently have traffic activity.

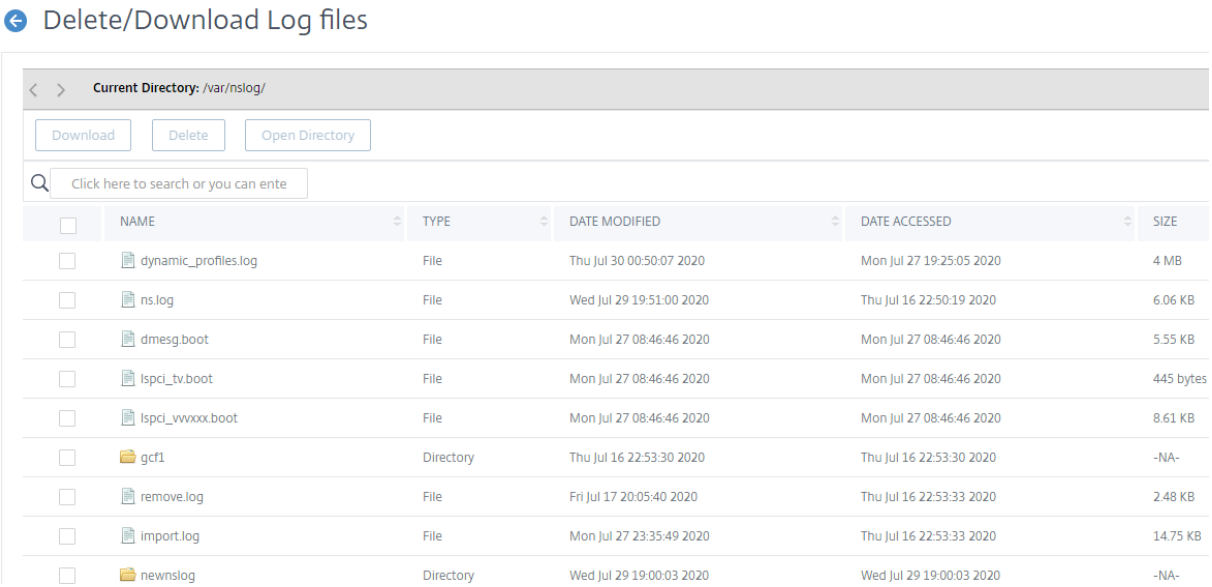
1. `Mbps (##.####)`. Total traffic Volume on the Service (Rx + Tx) converted into Mbits/s
2. `BWlmt (## kbits)`: Defined bandwidth limit.
3. `RspTime (## ms)`. Average response time of the service in milliseconds.
4. `Pkt(##/sec, ##bytes)`. Traffic volume in terms of packets per second going to the service; Average size of the packets.
5. `Wt (##)`. Weight index, used in load balancing algorithm.

> **Note:**
>
> If you divide this value by 10,000, then you get the actual configured weight of the service.

1. `RHits (##)`. Running requests counter used in Round Robin load balancing algorithm.
2. `CSvr (##, ##/sec)`. Number of connections to the service rate.
3. `MCSvr (##)`. Maximum number of connections to the service.
4. `OE (##)`. Number of connections to the service in the established state.
5. `RP (##)`. Number of connections to the service, residing in the reuse pool.
6. `SQ (##)`. Number of connections to the service, waiting in the surge queue.

Collect performance statistics and event logs using the Citrix ADC GUI

1. Navigate to **System > Diagnostics > Maintenance > Delete/Download log files**.
2. Select a file and click **Download** to download the file.



How to configure log file rotation

September 14, 2021

The Citrix ADC appliance generates logs in multiple directories and in various formats. Some of these logs are not rotated by default and can grow in size consuming too much disk space. By using the included utilities for log rotation (`newsyslog`), you can manage these logs consistently, by keeping only relevant information for easier management and administration.

The `newsyslog` utility included in the Citrix ADC firmware archives log files and rotates the system logs so the current log is empty during rotation. The system crontab runs this utility every hour and it reads the configuration file which specifies the files to rotate and the conditions. The archived files might be compressed if necessary.

The existing configuration is located in `/etc/newsyslog.conf`. However, because this file resides in the memory filesystem, the administrator must save the modifications to `/nsconfig/newsyslog.conf` so the configuration survives restarting the NetScaler.

The entries contained in this file have the following format:

```
logfilename [owner:group] mode count size when flags [/pid_file] [sig_num]
```

> **Note:**
>
> Fields within squared brackets are optional and can be omitted.

Each line on the file represents a log file and the conditions under which rotation must occur.

In the example, the `size` field indicates that the size of `ns.log` as 100 Kilobytes. The `count` field

indicates that the number of archived `ns.log` files as 25. A size of 100 K and count of 25 are the default size and count values.

> **Note:**
>
> When the field is configured with an asterisk (*), meaning that the ns.log file is not rotated based on time. Every hour, a crontab job runs the `newsyslog` utility which checks if the size of ns.log is greater than or equal to the size configured in this file. In this example, if it is greater than or equal to 100 K, it rotates that file.

```
root@ns# cat /etc/newsyslog.conf
# Netscaler newsyslog.conf

# This file is present in the memory filesystem by default, and any
    changes
# to this file will be lost following a reboot. If changes to this file
# require persistence between reboots, copy this file to the /nsconfig
# directory and make the required changes to that file.
#
# logfilename [owner:group] mode count size when flags [/pid_file] [
    sig_num]
/var/log/cron 600 3 100 * Z
/var/log/amd.log 644 7 100 * Z
/var/log/auth.log 600 7 100 * Z
/var/log/ns.log 600 25 100 * Z
<!--NeedCopy-->
```

The `size` field can be changed to modify the minimum size of the `ns.log` file or the field can be changed to rotate the `ns.log` file based on a certain time.

The daily, weekly, and/or monthly specification is given as: `[Dhh]`, and `[Dhh [Mdd]]`, respectively. The time-of-day fields, which are optional, default to midnight. The ranges and meanings for these specifications are:

```
Hh hours, range 0 ... 23
w day of week, range 0 ... 6, 0 = Sunday
dd day of month, range 1 ... 31, or the letter L or l to specify the
    last day of the month.
<!--NeedCopy-->
```

Examples:

Here are some examples with explanations for the logs that are rotated by default:

`/var/log/auth.log 600 7 100 * Z`

The authentication log is rotated when the file reaches 100 K, the last 7 copies of the auth.log are

archived and compressed with gzip (Z flag), and the resulting archives are assigned the following permissions –rw——-.

```
/var/log/all.log 600 7 * @T00 Z
```

The catch-all log is rotated 7 times at midnight every night (@T00) and compressed with gzip. The resulting archives are assigned the following permissions –rw-r—-.

```
/var/log/weekly.log 640 5 * $W6D0 Z
```

The weekly log is rotated 5 times at midnight every Monday. The resulting archives are assigned with permissions.

Common Rotation Patterns:

- `D0`. rotate every night at midnight
- `D23`. rotate every day at 23:00
- `W0D23`. rotate every week on Sunday at 23:00
- `W5`. rotate every week on Friday at midnight
- `MLD6`. rotate at the last day of every month at 6:00
- `M5`. rotate on every fifth day of the month at midnight

If an interval and a time specification are both given, then both conditions must be met. That is, the file must be as old as or older than the specified interval and the current time must match the time specification.

You can control the minimum file size but there is no limit on the file size before the `newsyslog` utility gets its turn in the next hour slot.

Debug newsyslog:

To debug the behavior of the `newsyslog` utility, add the verbose flag.

```
root@dj_ns# newsyslog -v
/var/log/cron <3Z>: size (Kb): 31 [100] --> skipping
/var/log/amd.log <7Z>: does not exist, skipped.
/var/log/auth.log <7Z>: size (Kb): 2 [100] --> skipping
/var/log/kerberos.log <7Z>: does not exist, skipped.
/var/log/lpd-errs <7Z>: size (Kb): 0 [100] --> skipping
/var/log/maillog <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/sendmail.st <10>: age (hr): 0 [168] --> skipping
/var/log/messages <5Z>: size (Kb): 7 [100] --> skipping
/var/log/all.log <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/slip.log <3Z>: size (Kb): 0 [100] --> skipping
/var/log/ppp.log <3Z>: does not exist, skipped.
/var/log/security <10Z>: size (Kb): 0 [100] --> skipping
/var/log/wtmp <3>: --> will trim at Wed Apr 1 04:00:00 2009
/var/log/daily.log <7Z>: does not exist, skipped.
```



```
/var/log/weekly.log <5Z>: does not exist, skipped.
/var/log/monthly.log <12Z>: does not exist, skipped.
/var/log/console.log <5Z>: does not exist, skipped.
/var/log/ns.log <5Z>: size (Kb): 18 [100] --> skipping
/var/log/nsvpn.log <5Z>: size (Kb): 0 [100] --> skipping
/var/log/httperror.log <5Z>: size (Kb): 1 [100] --> skipping
/var/log/httpaccess.log <5Z>: size (Kb): 1 [100] --> skipping
root@dj_ns#
<!--NeedCopy-->
```

How to free space on a /flash directory in a Citrix ADC appliance

September 14, 2021

This troubleshooting article explains how an administrator can free space from the /flash directory of a Citrix ADC appliance.

Procedure to free space in the /flash directory of a Citrix ADC appliance

1. Log on to the CLI of Citrix ADC by using SSH.
2. After you log on to the Citrix ADC CLI, switch to the shell prompt using the following command.`shell`.
3. Run the `df -h` command to see the availability of space on the Citrix ADC appliance.
4. If the capacity of the /flash directory is more than 90 percent or low, you must delete few files from this directory.
5. Run the following commands to view the contents of the /flash directory:

   ```
   cd /flash
   ls -l
   ```

6. You might find multiple files of various versions of the NetScaler software release. Ensure that the files present in this location are the ones applicable to the current version of the NetScaler software on your appliance. Run the following command to remove any other files from the appliance.

   ```
   rm <filename>
   ```


> **Note**
>
> Remove only the older versions of the kernel. The /flash directory must contain the files that the current version or build of the NetScaler software release is using and the kernel.gz file. Citrix recommends not to remove these files from the /flash directory.

Reference Material

October 15, 2021

Use this reference information to get an in-depth understanding of the following Citrix ADC components:

Citrix ADC SNMP OIDs - Details of the SNMP OIDs that can be used to obtain information from a Citrix ADC appliance.

Citrix ADC Syslog Messages - Details of the Syslog messages given by the Citrix ADC appliance.

Citrix ADC CLI Commands - Details of the commands that can be used to configure the Citrix ADC appliance through the CLI. You can also view the details of each command in the CLI, by entering the `man <ns-command-name>` command.

Citrix NITRO API Reference - Details of all operations that can be performed on the Citrix ADC appliance by using the REST API.

Citrix ADC Advanced Policy Expressions - Details of the expressions that can be used to define advanced policies.

Locations
Corporate Headquarters | 851 Cypress Creek Road Fort Lauderdale, FL 33309, United States
Silicon Valley | 4988 Great America Parkway Santa Clara, CA 95054, United States